

システム工学

工学博士 小野俊彦著

福岡工業大学

まえがき

現代はシステムの時代といわれるようにシステムが各所に導入され、その規模もますます増大している。システム工学はこのようなシステムを取り扱うに必要な技術の集大成として位置づけられており、わが国に導入されてから、すでに50数年が経過している。電力、鉄鋼、化学などの大形プラントはもちろんシステムであるが、単一の装置も視点によっては部品から成り立つシステムと見ることができる。したがってシステム工学を必要とする分野は多岐にわたっている。

システム工学は他の多くの工学のような特定の分野の理論・技術というより、各種の理論、工学のうちシステムに関連する部分を取り出して、集大成したものといえる。したがって、どの分野をシステム工学に入れるかについての考え方には個人差があり、システム工学の図書の内容も著者によりかなり幅の広いものになっている。本書はシステムの計画から運用までの各段階で必要とされる技術のうちから特に、対象のモデルの作成、計画技法、システムの最適化のための手法、システムの工程管理、そしてシステム信頼性について述べている。

本書は大学の情報系工学科のシステム工学の教科書として、半期の講義用として執筆したものであり、最新の話題も盛り込むよう配慮した。以下、簡単に本書の構成について述べる。1章では概論としてシステム全般に関する事項、すなわちシステムの定義、構造、分類、システム工学の定義、特徴を述べたのちシステム実現手順とそれに必要な理論、工学について言及し、本書を読むに当たってのガイダンスを与えている。2章ではモデリング、すなわち、モデルの作成とそれを使用してのシミュレーションについて述べている。システムを解析したり、シミュレーションを行なったりする際に的確なモデルを選ぶことが大事であり、そのための指針を与えることを目的にしている。モデルとしては、まず動的システムのモデルについて述べたのち、グラフモデルについて記している。グラフモデルには確率モデルであるマルコフモデルにも触れている。さらに最近の話題であるファジーモデルも取り上げている。その後、シミュレーションについて触れている。3章はシステムの計画、評価の際に必要な技法として、発想・予測技法であるKJ法、デルファイ法などについて述べたのち、評価技法である一対比較、デシジョンツリーなど、さらにはライフ

サイクルコストについても説明している。

4章から6章はシステム工学の中心的な課題である最適化手法を取り上げている。まず4章では基本的ともいえる線形計画法，5章では動的計画法について述べたのち，6章では非線形計画法の代表である勾配法や最近の話題のニューラルネットワークによる解法，遺伝的アルゴリズム，多目的最適化にも触れている。7章では大形プラントなど建設時の工程管理に不可欠な手法であるPERT法について述べている。8章ではシステムの信頼性，安全性について記している。最近の宇宙開発を始め大規模なシステムでは信頼性，安全性を高めることが重要である。まず信頼性の基礎的理論について説明ののち，システムの信頼性・安全性の解析手法の代表である故障モード解析(FMEA)や事故樹解析(FTA)について述べている。

ここで触れた話題は何れも一冊の本になるくらいの内容があり広範にわたるためすべてを尽くすことはできないので，ここではシステムの取扱いに必要な基本的な事項に限って述べている。したがって，さらに内容の深い追求を希望する場合はそれぞれの専門書に委ねたい。

教科書という性格から執筆に当たっては多く人々の研究成果や図書を参考にさせて頂いた。ここに厚く御礼申し上げる。なお参照した個所などは参考文献にて触れているが，もし洩れている個所があればご容赦を頂きたい。

平成10年4月1日

小野 俊彦

目次

1	システム工学概論	1
1.1	システムとは	1
1.1.1	定義	1
1.1.2	システムの例	2
1.1.3	システムの構造	2
1.1.4	システムの分類	3
1.2	システム工学とは	5
1.2.1	定義	5
1.2.2	システム工学の特徴	5
1.2.3	システム実現手順	6
1.2.4	システム実現上の問題点とシステム工学	7
1.3	本書の構成	9
2	モデリングとシミュレーション	11
2.1	概要	11
2.2	動的システムの数式モデル	12
2.2.1	微分方程式モデル	12
2.2.2	伝達関数モデル	12
2.2.3	周波数応答モデル	12
2.2.4	状態変数モデル	13
2.2.5	入出力モデル	14
2.2.6	離散系の状態変数モデル	15
2.2.7	パルス伝達関数	16
2.2.8	確率的システムのモデル	16
2.2.9	ブロック線図	17
2.2.10	シグナルフロ - 線図	17

2.2.11	各数式モデル間の関係	17
2.2.12	モデルと解析	18
2.3	グラフモデル	20
2.3.1	有向グラフによるネットワークモデル	21
2.3.2	マルコフモデル	23
2.3.3	フローモデル	25
2.4	ファジーモデル	27
2.5	シミュレーション	29
3	システムの計画技法	33
3.1	発想・予測技法	33
3.1.1	KJ法	33
3.1.2	シナリオライティング	34
3.1.3	デルファイ法	36
3.2	評価技法	36
3.2.1	一対比較	36
3.2.2	マトリックス法	38
3.2.3	デシジョンツリ	40
3.2.4	テクノロジー - アセスメント	41
3.3	ライフサイクルコスト	42
4	線形計画法	45
4.1	はじめに - 最適化手法について	45
4.2	線形計画法とは	46
4.3	図的解法	46
4.4	シンプレックス法	47
4.5	2段階法	57
4.6	双対問題	63
5	動的計画法	65
5.1	はじめに	65
5.2	動的計画法の一般解	65
5.3	最短経路決定問題 [応用例 - 1]	67
5.4	資源配分問題 [応用例 - 2]	69
5.5	倉庫運営問題 [応用例 - 3]	72
5.6	資金配分問題 [応用例 - 4]	75

6	その他の最適化手法	79
6.1	勾配法	79
6.1.1	最適勾配法	80
6.1.2	共役勾配法	80
6.1.3	ニュートン法	82
6.2	生物学的方法	82
6.2.1	ニューラルネットワーク	82
6.2.2	遺伝的アルゴリズム	86
6.3	多目的最適化	88
6.3.1	問題の定義と解の性質	88
6.3.2	多目的最適化の解の求め方	89
6.4	制約条件つき最適化	90
6.4.1	ラグランジェ乗数法	91
6.4.2	一般の場合	91
7	システム管理	93
7.1	概要	93
7.2	P E R T	93
7.3	C P M	99
8	システム信頼性	101
8.1	信頼性の概念と評価指標	101
8.1.1	信頼度関数と故障率	101
8.1.2	信頼性の指標	102
8.2	故障モデル	104
8.2.1	故障のモデルとワイブル分布	104
8.2.2	その他の分布	107
8.2.3	故障パターンと対策	107
8.3	システムの信頼度	108
8.3.1	直列系，並列系の信頼度	108
8.3.2	マルコフモデルによる冗長系の解析	111
8.4	システム信頼性解析手法	116
8.4.1	故障モード影響解析 (F M E A / C A)	117
8.4.2	事故樹解析 (F T A)	119
8.4.3	最小カットセットと最小パスセット	122
8.4.4	システム信頼度の計算	123

8.4.5	解析プログラム	125
8.4.6	解析時に考慮すべき事項	127
9	演習問題略解	133

システム工学概論

[概要] システムとは何か，システム工学とはどのような工学か，システム工学はどのようなものから成り立ち，どの分野で用いられているか，など全体的事項について述べて2章以降の導入部を構成する．

システムとは

1.1.1 定義

現代は「システム」(system)の時代と言われているように，システムの語は各所に氾濫し，日常語化しており，改めて「システムとは何か」と問われても返答に困るほどである．systemの日本語訳は「系，系統，体系，制度」などであるが，むしろ，そのままを日本語化した「システム」の方がぴったりする．システム工学ではシステムを対象とするので，まず，システム概念を一度，考え直して置くことはシステム工学を学ぶ上で無益なことではないだろう．

さてシステムを特徴付けている事項を思い付くままに並べてみると

1. 複数の要素より成立っている．1個より成立っているもの，すなわち単体は普通，システムとは言わない．
2. 相互に関連し，組織化され，一つの閉社会を構成している．たとえ複数のものから成立っていても，相互に関係がなかったり，各個がばらばらに機能しているもの，あるいは構成要素が不明確であったり，他との区別のつかないものもシステムとは言わないだろう．
3. 共通の目的をもち，全体として統一している．前項に関連するが，全体が一つにまとまるとともに共通の目的・使命をもって活動してはじめてシステムと言えよう．
4. 主として人工的なもの．人体システムや自然システムなどのシステムもあるが，システム工学で対象とするのは人為的に変えることのできる人工物より構成されたシステムを対象とする．

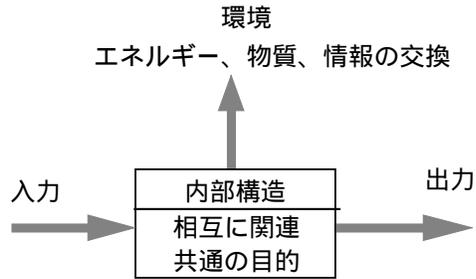


図 1.1: システムの構造

以上を総合すると「システム」とは JIS (Z8121) で規定しているように

多数の構成要素が有機的な秩序を保ち、同一の目的に向かって行動するもの

というのが妥当な定義であろう。したがって、システムを図示すると図 1.1 のとおりになる。システムはこのように機能に名付けられたものであり、物、すなわちハードウェアではないし、また、その適用も時と場合によって変わることが、システムを分かり難くしている。人間を例にとって考えると、一人の人間は社会の構成要素であり、システムとはいえないかも知れないが、立場を変えると人体は脳を中心に手・足などの運動器官、目・耳などの感覚器官、内臓などの多くの器官が有機的に結合して、一つの目的をもって機能する立派なシステムである。このように視点の相違によってシステムになったり、ならなかったりする。要は現在、取扱う問題のレベルによって決まることである。

1.1.2 システムの例

一般にシステムの名前が付けられているものを思いつくままに集めたのが、表 1.1 である。社会・行政機構から始まって、大規模設備、製造設備、装置、運用・管理方式、制御方法など多岐にわたっている。このように形のあるものから形のないもの、大きなものから小さいもの、あるいは社会的なものから工業的なもの、と千差万別である。共通するのは先に定義した機能をいずれも持っていることである。

1.1.3 システムの構造

システムを子細に観察するとシステム自体がさらに小さい、いくつかの小規模のシステムより構成されるという具合にシステムが構造を持つ場合が多い。また、大規模のシステムをつくる時などでは、個々の要素より直接、全体システムを構成するのではなく、機能をいくつかに分割して、それぞれを遂行する小規模のシステムをつくり、これの集合体と

表 1.1: システムの例

分野	システム名	
社会, 行政	広域行政システム	地震予知システム
医療	医療システム 人体計測システム	患者集中監視システム
交通	新交通システム 列車座席予約システム	列車運行管理システム
航空	航空管制システム	
エネルギー	エネルギー管理システム	発電所最適運転システム
製造	FMS	燃料供給システム
管理	事務管理システム	生産管理システム
装置, 方式	情報システム 二重化システム ネットワークシステム 制御システム	計算機制御システム ハイアラーキシステム バックアップシステム
判断	意志決定支援システム	

して大規模システムとすることがよく行われる。このような小規模のシステムを部分システム (subsystem) という。このような部分システムへの分割の仕方には各種あるが、その代表的なものが 図 1.2 に示す階層構造 (hierarchical structure) と分散構造 (decentralized structure) である。階層構造では上下の関係で機能の分担が行われる。情報の交流は指令と応答というように上下の関係を通じて行われ、上下間で強固の関係を保っている反面、同一の階層間の情報交換も上位のシステムを介して行われるため、必ずしも迅速ではない。一方、分散構造では水平的な機能分散が行われ、同一階層間での協調のもとでシステムの使命が遂行される。もちろん、この場合もシステムとしての統一は必要なので、分散されたサブシステムのいずれかが全体の統制機能も持つことになるが、簡単に他のサブシステムのいずれかに移すことができ、事故などの異常事態に直ちに応じられる融通性を持っており、最近の計算機制御システムなどに好んで採用される。

1.1.4 システムの分類

システムはその特性よりいくつかに分類して取扱われる。その代表的なものを示す。

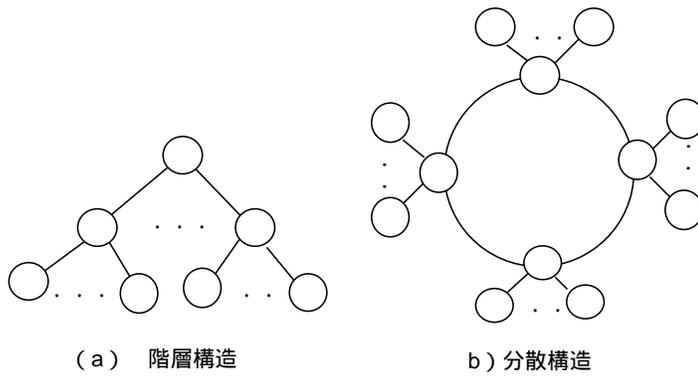


図 1.2: 代表的なシステム構造

1. 時間の継続の形態により連続 (時間) システム (continuous system) と離散 (時間) システム (discrete system) に分けられる。計算機制御のように、対象のシステムが連続システムでも計算機は離散時間で動作するので離散システムになるものがある。

このほか空間的・時間的なものを含めて切れ目の有無により離散事象システム (discrete event system) , 連続事象システム (continuous event system) の分類が用いられる場合もある。

2. 時間的な特性に関しての分類として、状態値が時間の経過に関係しない静的システム (static system) と状態値の時間的な変化を問題とする動的システム (dynamic system) とがある。
3. 特性が時間に関係なく一定な時不変システム (time-invariant system) と時間とともに変化する時変システム (time-variant system) とがある。
4. 特性の線形性から線形システム (linear system) と非線形システム (nonlinear system) とに分けられる。
5. 現象の発生状況から確定的システム (deterministic system) と確率的システム (stochastic system) とがある。システムに確率的な外乱 (雑音) がある場合には後者の扱いを必要とする。

実際は以上に述べた分類のいくつかを組合わせた性質を持っている。

システム工学とは

1.2.1 定義

システム工学とは，JIS Z8121（オペレ - ションリサ - チ用語）による定義に従い

システムの目的を最もよく達成するために，対象となるシステムの構成要素，
組織構造，情報の流れ，制御機構などを分析し，設計する技術

と見てよいだろう．まず問題はシステムの目的であり，これには性能，品質，信頼性，生産性，時間など各種に関係するものがあり，いずれを選ぶかによって取組みも変わってくる．そのおもなものを挙げると

- * 最小の資源（原材料，エネルギー，人件費）
- * 最大の生産効率，生産性
- * 安定した性能
- * 高い信頼性

などがあり，いずれを重視するか，あるいは，どのように調整するかは問題ごとによって変わってくるし，それにより適用するシステム工学の内容が異なる．

1.2.2 システム工学の特徴

システム工学の特長を挙げると

1. 対象が多様であり，あいまいなものが多いことである．前節で述べたように，対象とするシステムには各種のものがあり，そこで問題とされる事項も各種各様である．さらに問題点も明確に与えられていなく，あいまいな場合も多い．そのような場合システムを分析して問題点を発見する能力も要求される．
2. 各種の解決方法がある．システム工学は一つのまとまった工学というより，各種の工学の中からシステムに関連するものを抽出して，まとめたものである．一般に解決方法は一つではなく，いろいろな解決策が考えられる場合が多く，対象とする問題に最も適合した方法を選択して，適用する能力を必要とする．
3. 各種の要求事項を調整し，まとめる能力が求められる．システムに要求される各種の事項，時には互いに対立する要求事項を調整し，全体として満足すべき状態にもたらず機能が必要である．また評価関数も単一ではなく，多属性で，多くの要求を同時に満足する解を求めねばならない場合も多い．
4. 人間を含むシステムがある．社会システムを始め，人間をそのなかに含むシステムが多い．その際は人間の持つ特性，特に，あいまい性，不正確さ，きまぐれさ，などを考慮した取組みが不可欠である．人間工学の助けも必要としよう．

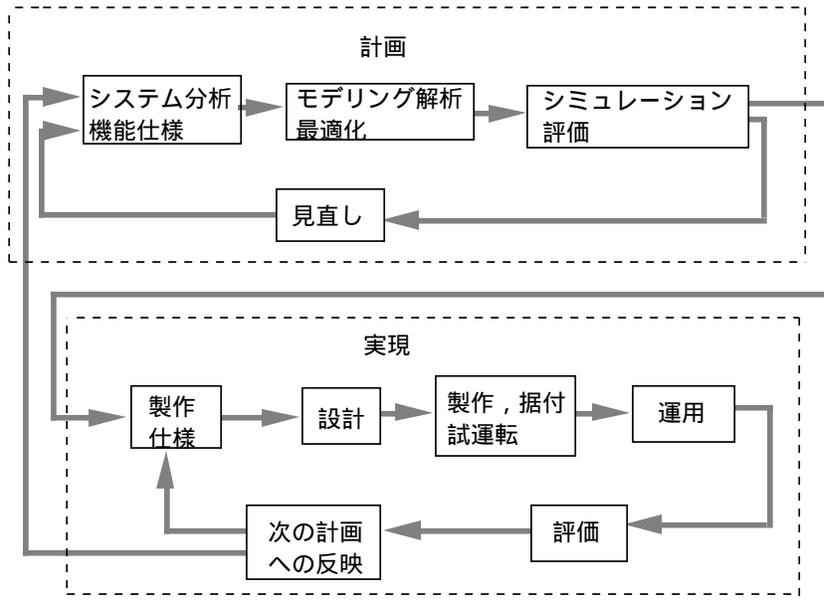


図 1.3: システムの取り組み方

5. 悪構造の問題が多い。人間を含む問題に関連し、その構造が明確でない場合が多い。このようなモデル化の困難さに対する配慮も必要とする。

1.2.3 システム実現手順

システム工学はシステムを計画し実現し、運用するまでの各段階において重要な働きをする。これを眺めるため、システムの実現手段をここでは考えてみよう。図1.3に全体の系統を示す。

まず計画段階から始める。

1. 実現しようとするシステムを分析し、その目標を定め、それを具体的に記述したシステムの機能仕様の形で明確にする。
2. つぎに、このような目標を達成できるかどうかを見極める必要があり、そのためにはシステムのモデル化を行い、数学的な手法などを適用できるよう抽象化する。これに基づいての解析、代替案の検討、システムの最適化を行う。
3. これらの結果はシミュレーションなどにより、所定の性能を満足できるか、あるいはコスト対性能比など、各方面からの評価を行う。これに成功すれば計画段階は終了し、実現段階に移る。満足できない場合にはモデルの見直し、あるいは最初に戻り分析

をやり直して機能仕様書の変更を行う。このようなサイクルの繰返しにより目標に近付ける努力を重ねる。

つぎに実現段階であるが

4. 計画に基づいて具体的に製作仕様書を作成し、
5. これに従って設計し、
6. さらに製造、試験、現地据付け、試運転を経て、
7. 運用段階に到達し、無事、システムを実現できることになる。この段階では寿命が尽きるまで正常に運転することが求められる。
8. 運用の結果は性能、信頼性、保全性、運転費など各方面から評価を行い、
9. それに従って、つぎの計画に反映させる。

システム工学はこれらの各段階において重要な働きをする。つぎに、これを眺めてみよう。

1.2.4 システム実現上の問題点とシステム工学

システムの実現過程においてシステム工学に期待されることを大別すると、次の3点になる。図1.4にこれら問題点とその解決に用いられるシステム工学の手法との関係を示している。

(1) システムの機能の定性的・定量的な把握：計画したシステムが所定の目的を達成できるか否かを判断するためには、その機能をできるだけ正確に把握する必要がある。しかし、先に述べたようにシステムは多種多様であるため、機能の把握の方法もそれぞれにより異なってくる。たとえば、動的システムの性能や安定性の把握のためには状態空間での解析や安定性判定手法が必要になる。社会システムのような多数の要素が互いに関連し合うものにはシステム・ダイナミックスの手法が用いられている。あるいは多数の要素より成りランダム性のあるものでは統計的手法の主成分分析あるいはクラスタ分析などが有効である。理論的なモデルが作れないものには経験や実データを基にしたモデルが必要であり、GMDH(Group Method of Data Handling) などがある。このほか、システムの信頼性や安全性を確認するために、システム信頼性・安全性の解析手法も必要である。さらに、性能を実地に検証する前に、コンピュータによるシミュレーションも有効な手法として各所で用いられている。

(2) 複数の要求を調整しつつ、最適な解を得る手段：システムに期待される事項には互いに対立するものも多い。たとえば、最小の費用で最大の効果を出したい、最低のコストで最高の信頼性を得たい、最低の人員で最短期間に完成させたい、などがある。あるいは、

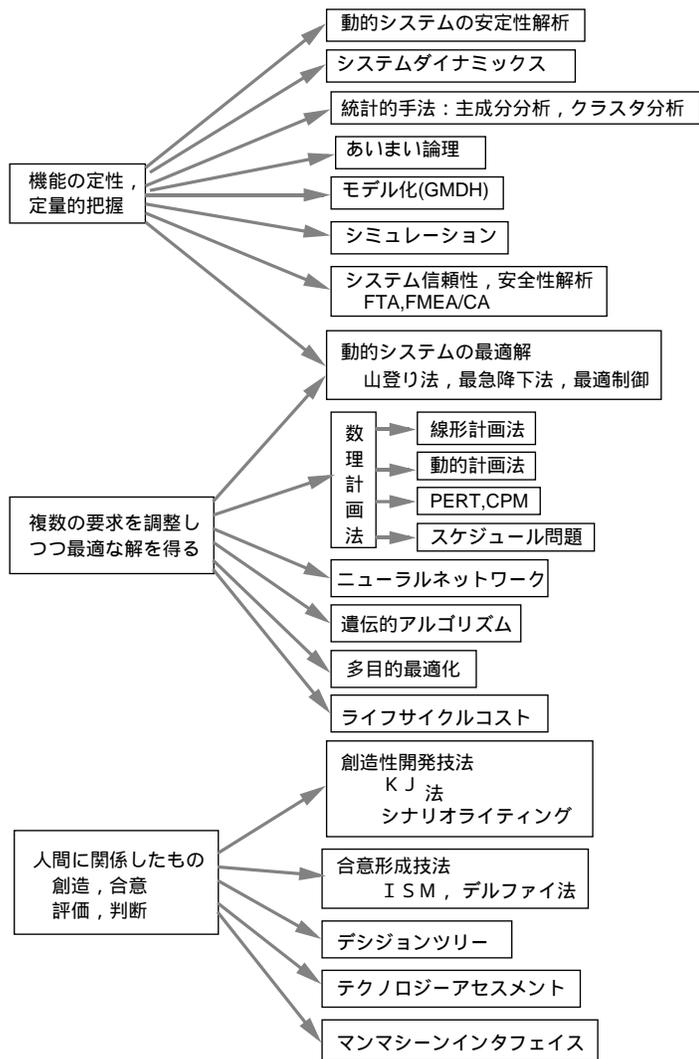


図 1.4: システム実現上の問題点とシステム工学

各種の定数，パラメータを調整しつつ最適解を求める場合もある．このような目的には従来より数理計画法の名のもとで線形計画法，動的計画法，PERT，CPM，などがある．最近では生物の機能を模擬したニューラルネットワークや遺伝的アルゴリズムなども用いられている．さらに多目的最適化の手法もある．このほか建設費や保全費などを総合的に考えるライフサイクル・コストの考え方も重要である．

(3) 判断，評価，創造性など人間に関連した事項の取扱い：これに関してはKJ法，デルファイ法を始めとする創造性開発技法，合意形成技法，あるいはテクノロジー・アセスメン

トなどが用いられている．さらに突発的な事故の際の対応のために，緊急時の人間行動やマンマシン・インタ・フェイスもシステムが巨大になればなるほど重要となる．

本書の構成

以上に述べた事項を中心に本書ではつぎの順序で説明している．

まず2章ではモデルの問題を取り上げ，その中でも数式モデルとして代表的な動的システムのモデルについて説明の後，グラフモデル，ファジーモデルについて触れる．3章ではシステム計画技法について述べる．まず発想，予測技法として代表的なKJ法，シナリオライティング法について説明したのち，ライフサイクルコストについて考える．4章から6章は最適化問題を扱い，線形計画法，動的計画法，勾配法，最近のニューラルネットワークや遺伝的アルゴリズムによる方法，および多目的最適化について説明している．7章ではシステムの工程管理のための手法としてPERT，CPMについて述べる．8章ではシステム信頼性，安全性に関して考える．信頼性を扱う際の基本的事項について説明の後，代表的な手法であるFMEA/CA，FTAについて触れる．

演習問題

- 1.1 システム工学の特徴を挙げよ．
- 1.2 システムを構築する際，どのような点に配慮すべきか．
- 1.3 システム実現上の問題点を挙げ，システム工学の役割を示せ．
- 1.4 システムの階層構造と分散構造の特徴を比較せよ．

モデリングとシミュレーション

〔概要〕システム工学の第一歩としてモデリングが重要である。モデルを基に数学的な解析を行ったり、シミュレーションによる性能の検証などを行なう。代表的なモデルである数学モデル、グラフモデルおよびファジィ・モデルについて述べたのち、シミュレーションについて説明する。

概要

システム工学においてはシステムの持つ特性、構成要素の相互関係を問題とするので、その検討のために、システムを数式で表したり、あるいは図、グラフにより視角的に表現することが有効である。このようにシステムの持つ性質を抽出して、数式や図などに表現したものがモデルであり、それを作成することをモデリング (modeling または modelling) という。このようなモデリングを通してシステムの特性を解明し、問題の解決方法を探ることができる。さらに解決策を実地に実施する前の計算機シミュレーションによる事前の性能検証の際にも、このモデリングは不可欠である。

モデルの内容はそれによる解析、シミュレーションなどの内容により変わり、ひとつのシステムにも各種のモデルが考えられる。その選択方法として特に定まったものはなく、経験や前例に従うことが多い。モデルを大別すると数式モデルとグラフないし図的モデルに分けられる。多くの場合、いずれのモデルで表すことも可能である。例えば動的システムでは数式モデルとしては微分方程式、状態方程式、入出力方程式、伝達関数、周波数応答などがあり、グラフモデルではブロック線図、シグナル・フロー線図などがある。グラフモデルではそのほか、ネットワークモデル、マルコフモデル、フローモデルなどがある。またあいまいな現象を対象にしたものにファジーモデルがある。

シミュレーションはシステムに適用する各種方式を検討したり、その性能を検証する際に欠かせないものである。もちろん実際のシステムとは異なるので、多少の誤差は有り得るので、それを認識の上、活用すれば有用な手段となる。

動的システムの数式モデル

2.2.1 微分方程式モデル

微分方程式モデルは動的システムのモデルの基本であるが、数学的な取扱いが厄介なこともあり、多くは以下の節で述べるモデルに変換して解析される。微分方程式モデルは電気理論、物理法則、力学、化学反応式などをもとに、微分方程式としてモデル化したものである。集中定数系では線形ないし非線形の常微分方程式で表し、非線形の場合は動作点の近傍で線形化して取扱うことが多い。また分布定数系は偏微分方程式で表されるが、これも集中定数系あるいは、むだ時間による近似で扱われることが多い。

線形常微分方程式の場合は一般に次の形式で表される。ここで $u(t)$ は入力を、 $y(t)$ は出力を表す (式では時間 t を省略)。

$$\begin{aligned} a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = \\ b_m \frac{d^m u}{dt^m} + b_{m-1} \frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u \end{aligned} \quad (2.1)$$

2.2.2 伝達関数モデル

微分方程式をラプラス変換により複素数の s 領域に変換し、初期値を零として、出力と入力との比をとったものが伝達関数 (transfer function) である。これはインパルス応答のラプラス変換でもある。実関数 $c(t)$ のラプラス変換 $C(s)$ はつぎの式により求める。

$$C(s) = \mathcal{L}[c(t)] = \int_0^{\infty} c(t) e^{-st} dt \quad (2.2)$$

伝達関数は通常、複素数 s の有理式で表され、式 (2.1) の微分方程式の場合の伝達関数 $G(s)$ はつぎの式になる。

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \quad (2.3)$$

ここで $U(s), Y(s)$ はそれぞれ入力 $u(t)$, 出力 $y(t)$ のラプラス変換である (初期値 $y(0) = 0$ とする)。また入力、出力がベクトルの時は伝達関数は行列となり、伝達関数行列 ($G(s) = Y(s)U(s)^{-1}$) という。

ラプラス変換の特徴は微分、積分演算が掛算、割算などの代数演算で扱えることであり、したがって伝達関数モデルでは代数的な方法で各種特性の解析ができる。

2.2.3 周波数応答モデル

動的システムの周波数領域での特性を見るのに用いられるのが周波数応答で、微分方程式のフーリエ変換により求められる。伝達関数が求まっている場合には、 $s = j\omega$ と置くこ

とにより簡単に求まる．周波数応答は周波数 $f = \omega/2\pi$ の正弦波入力を与えた時の出力と入力との振幅比と位相差で示され，応答特性や安定の検討に使用される．あらゆる波形の入力はフーリエ変換により各周波数の正弦波成分に分解できるので，各周波数の入力に対する特性が判明すれば，その動作を予測できる．

周波数応答をグラフに表わしたものにボード線図がある．周波数に対するゲインと位相との関係を図示したもので，周波数には対数目盛，ゲインにも対数に変換したデシベル (db)，位相は角度が用いられる．安定特性の解析に使用され，特にゲイン 1 (すなわちゼロデシベル) 及び位相 180 度付近の特性が安定と関係が深い．

2.2.4 状態変数モデル

伝達関数では入力と出力との関係を中心に考えてきたが，システムの内部状態という観点から見たのが，この状態変数モデルである．一般に内部状態を表す変数として n 次元の状態変数ベクトル

$$x(t) = [x_1(t), \dots, x_n(t)]^T \quad (2.4)$$

を考えると，内部状態ベクトルの変化は次式で表わせる．

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.5)$$

この式を状態方程式という．ここで $u(t)$ は制御入力を表す r 次元のベクトル， A, B はそれぞれ $n \times n, n \times r$ 次元の行列である．出力ベクトル (または観測ベクトル) $y(t)$ は状態変数ベクトル $x(t)$ にもとづくシステムの m 次元の出力ベクトルで，つぎのとおり求まる．

$$y(t) = Cx(t) \quad (2.6)$$

この式を出力 (または観測) 方程式という．ここで C は $m \times n$ 次元の行列となり，出力行列 (または観測行列) という．なおシステムの構造によっては出力 $y(t)$ に制御入力 $Du(t)$ (D は $m \times r$ 次元行列) が付加される場合もある．

以上をブロック線図に示したのが図 2.1 である．

つぎに伝達関数モデルとの関係を求める．まず式 (2.5) をラプラス変換し，変形すると

$$(sI - A)X(s) = x(0+) + B U(s)$$

したがって

$$X(s) = (sI - A)^{-1}x(0+) + (sI - A)^{-1}B U(s) \quad (2.7)$$

この式にラプラス逆変換を適用すると，第 2 項は畳込み積分となり，

$$x(t) = \Phi(t)x(0+) + \int_0^t \Phi(t - \tau)Bu(\tau)d\tau \quad (2.8)$$

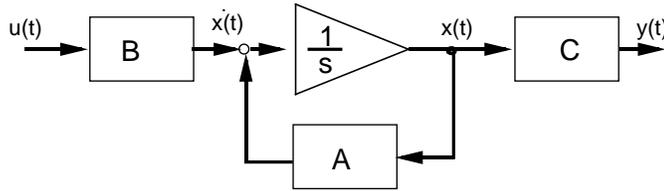


図 2.1: 連続系のブロック線図

ここで $\Phi(t)$ は状態推移行列 (state transition matrix) といわれ、次式で求められる。

$$\Phi(t) = \mathcal{L}^{-1}(sI - A)^{-1} = I + At + \frac{A^2 t^2}{2!} + \cdots + \frac{A^n t^n}{n!} + \cdots \quad (2.9)$$

$$= e^{At} \quad (2.10)$$

$\Phi(t)$ にはつぎの性質がある。

$$\Phi(0) = I$$

$$\Phi(t_2 - t_1)\Phi(t_1 - t_0) = \Phi(t_2 - t_0)$$

$$\Phi^{-1}(t) = \Phi(-t) \quad (2.11)$$

同様に式(2.6)をラプラス逆変換し、式(2.7)の関係を代入し、 $y(t)$ の初期値したがって $x(0+)$ を0とすると

$$Y(s) = CX(s) = C(sI - A)^{-1}B U(s) \quad (2.12)$$

が得られ、伝達関数行列 $G(s)$ は係数行列を用いて次式のように表わされる。

$$G(s) = C(sI - A)^{-1}B \quad (2.13)$$

2.2.5 入出力モデル

システムの入出力のみに注目して、この両者間の関係を示したモデルが入出力モデルであり、連続時間系でスカラーの入出力の場合は次式で表される。

$$\begin{aligned} a_n y^{(n)} + a_{n-1} y^{(n-1)} + \cdots + a_1 y^{(1)} + a_0 y = \\ b_m u^{(m)} + b_{m-1} u^{(m-1)} + \cdots + b_1 u^{(1)} + b_0 u \end{aligned} \quad (2.14)$$

ここで $y^{(n)}$ 、 $u^{(n)}$ などはそれぞれの n 次の時間微分を表わす。

2.2.6 離散系の状態変数モデル

計算機制御では一定のサンプリング時間 T ごとのデータに基づいて制御する。このような場合はシステムを離散系（離散時間系）として取扱う必要があるので、ここでは離散系のモデルを求める。

時間の開始を t_0 とすると、式(2.8)より

$$x(t) = \Phi(t - t_0)x(t_0) + \int_{t_0}^t \Phi(t - \tau)Bu(\tau)d\tau \quad (2.15)$$

ここで、あるサンプリング時間での状態を表わす式を求めるために、サンプリング時間を T とし、 $t_0 = kT$ 、 $t = (k + 1)T$ とすると

$$x[(k + 1)T] = \Phi(T)x(kT) + \int_{kT}^{(k+1)T} \Phi[(k + 1)T - \tau]Bu(\tau)d\tau \quad (2.16)$$

ここで制御入力 $u(\tau)$ は時間区間 $kT \leq \tau \leq (k + 1)T$ で一定で、 $u(\tau) = u(kT)$ とすると

$$x[(k + 1)T] = \Phi(T)x(kT) + G(T)u(kT) \quad (2.17)$$

となる。この式が離散時間系の状態方程式である。ここで

$$G(T) = \int_{kT}^{(k+1)T} \Phi[(k + 1)T - \tau]Bd\tau = \int_0^T \Phi(T - \tau)Bd\tau \quad (2.18)$$

観測方程式は式(2.6)より

$$y(kT) = Cx(kT) \quad (2.19)$$

以上をまとめ、表記の簡略化のため時間 T を省略して $x(kT)$ 、 $u(kT)$ 、 $y(kT)$ をそれぞれ $x(k)$ 、 $u(k)$ 、 $y(k)$ と表し、また $\Phi(T)$ 、 $G(T)$ も Φ 、 G とすると

$$x(k + 1) = \Phi x(k) + Gu(k) \quad (2.20)$$

$$y(k) = Cx(k) \quad (2.21)$$

これをブロック線図に示したのが図2.2である。

式(2.9)より、 T が小さい場合は近似的に次式が成立する。

$$\Phi(T) \simeq I + AT \quad (2.22)$$

$$G(T) \simeq \int_0^T [I + A(T - \tau)]Bd\tau = (T + \frac{1}{2}AT^2)B \simeq BT \quad (2.23)$$

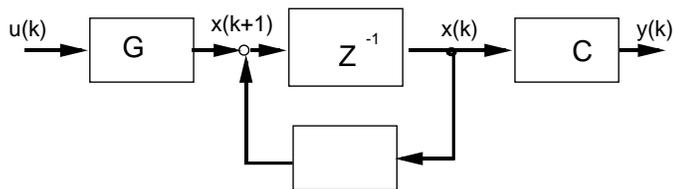


図 2.2: 離散系のブロック線図

2.2.7 パルス伝達関数

離散時間系においては現象はサンプリング時間 T の間隔で観測されるので, $u(t)$, $y(t)$ を入力, 出力とした場合の入出力モデルは一般に下記の差分方程式で表わされる.

$$a_n y(k-n) + a_{n-1} y(k-n+1) + \cdots + a_1 y(k-1) + a_0 y(k) = b_m u(k-m) + b_{m-1} u(k-m+1) + \cdots + b_1 u(k-1) + b_0 u(k) \quad (2.24)$$

ここで $y(i)$ (i は $k-n$ など) は $y(iT)$ を表わす. $u(i)$ も同様.

つぎに, このような差分方程式を代数的に解くために, つぎの Z 変換を導入する.

$$F(z) = Z[f(k)] = \sum_{k=0}^{\infty} f(k) z^{-k} \quad (2.25)$$

Z 変換の性質として, 現象を時間 T だけ遅らせることは z 領域では z^{-1} を乗ずることの相当する. この意味で z^{-1} を遅延演算子と呼んでいる.

$y(k)$, $u(k)$ の Z 変換をそれぞれ $Y(z)$, $U(z)$ で表わすと入出力式の Z 変換は

$$(a_n z^{-n} + a_{n-1} z^{-n+1} + \cdots + a_1 z^{-1} + a_0) Y(z) = (b_m z^{-m} + b_{m-1} z^{-m+1} + \cdots + b_1 z^{-1} + b_0) U(z) \quad (2.26)$$

となり, したがって

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_m z^{-m} + b_{m-1} z^{-m+1} + \cdots + b_1 z^{-1} + b_0}{a_n z^{-n} + a_{n-1} z^{-n+1} + \cdots + a_1 z^{-1} + a_0} \quad (2.27)$$

この $G(z)$ をパルス伝達関数と呼んでいる. 連続系の伝達関数と同様に離散時間系の安定性の解析などに利用されている.

2.2.8 確率的システムのモデル

確率的システムの取扱いに多く用いられている状態変数モデルを例として示す. 離散系確率的システムの状態方程式は普通, 次の式で表される.

$$x(k+1) = \Phi x(k) + Gu(k) + \xi(k) \quad (2.28)$$

$$y(k) = Hx(k) + \eta(k) \quad (2.29)$$

ここで、 x, u, y はそれぞれ n, r, m 次のベクトルである。 $\xi(k), \eta(k)$ はそれぞれシステム雑音、観測雑音を表し、正規性白色雑音である。それぞれの平均値と共分散行列は、 $\xi(k)$ に関しては $E\{\xi(k)\} = 0, E\{\xi(k) \cdot \xi^T(k)\} = Q\delta_{i,j}$ 、 $\eta(k)$ に関しては $E\{\eta(k)\} = 0, E\{\eta(k) \cdot \eta^T(k)\} = R\delta_{i,j}$ である。なお、 $\delta_{i,j}$ はクロネッカーのデルタである。すなわち

$$\begin{aligned} \delta_{i,j} &= 1 && i = j \text{ において} \\ &= 0 && i \neq j \text{ において} \end{aligned}$$

2.2.9 ブロック線図

ブロック線図はシステムの機能を構成要素とその間の信号の流れにより図示したものである。構成要素の表示には通常、伝達関数を使用される。このような伝達関数を用いた表示では、単なる表示のみならず図を使用して式の等価変換が可能であり、伝達関数に関する演算にも使用できる。まず、システムを構成する要素やサブシステムを矩形のブロックで示し、その中に伝達関数を記入する。信号の流れは矢線、信号の引出し点と合流点(加算または減算)とにより示す。表2.1に示す変換公式が適用できる。

ブロック線図による表示の特長にはつぎのものがある。

- 1) システムを構成する各要素をつなげてシステムを構築することができる。
- 2) 全体像を視覚的に把握できる。
- 3) 変換公式を適用して、図を使用した計算により、各要素をまとめてより大きなサブシステムとし、最終的には1つの伝達関数に簡単にまとめることができる。

2.2.10 シグナルフロ - 線図

ブロック線図の場合とは逆に信号の流れを中心に示した図である。信号を節(node)で、信号間の関係を枝(branch)で示す。枝はトランスミッタンス(transmittance)といい、その横に関係を表す値を記入する。この値として動的システムでは伝達関数を用いる。シグナルフロ - 線図にも表2.2に示す変換公式があり、これにより図的に等価変換を行い簡単化できる。シグナルフロ - 線図は動的システムのほか、静的システムに対しても使用できる。

2.2.11 各数式モデル間の関係

これまで述べてきた各数式モデル間は相互に変換可能であり、それを示したのが図2.3である。

表 2.1: ブロック線図の変換公式

変換方法	変換前	変換後
加算点の移動		
分岐点の移動		
並列		
直列		
フィードバック		

2.2.12 モデルと解析

動的システムの特長について解析する際、それぞれの目的に最も適したモデルを選定する必要がある。動的システムの解析では特にフィードバック制御システムの研究が最も盛んである。その詳細は制御関係の図書に譲り、ここでは解析内容とモデルとの関係について簡単に説明する。

1) 動的システムの構築：各要素を組み立てて動的システムを構築する際は伝達関数モデルとその図的表現であるブロック線図が適している。これらは入力と出力との関係で示されており、かつ代数的演算が適用できるため、要素間を簡単に接続できる。さらにブロック線図の変換公式を利用して各要素を統合して1つの伝達関数にまとめることができる。

表 2.2: シグナルフロー線図の変換公式

変換方法	変換前	変換後
加算点の移動		
分岐点の移動		
並列		
直列		
フィードバック		
自己ループ		

2) 安定性の解析: フィードバック制御系の解析の中心は安定性の解析であり, 多くの研究がなされてきた.

- 最も昔から愛用され, 現在もまだ現場で使用されているのが周波数モデルによる安定性の解析である. あらゆる入力はフーリエ変換により各周波数成分に分解できるので, 各周波数に対する安定性を解析することにより, システム全体の安定性を知ることができる. オープンループの周波数特性より求めたゲイン余有と位相余有が指標として用いられている. 最近では H_∞ ノルムを指標として用いた方法も行なわれて

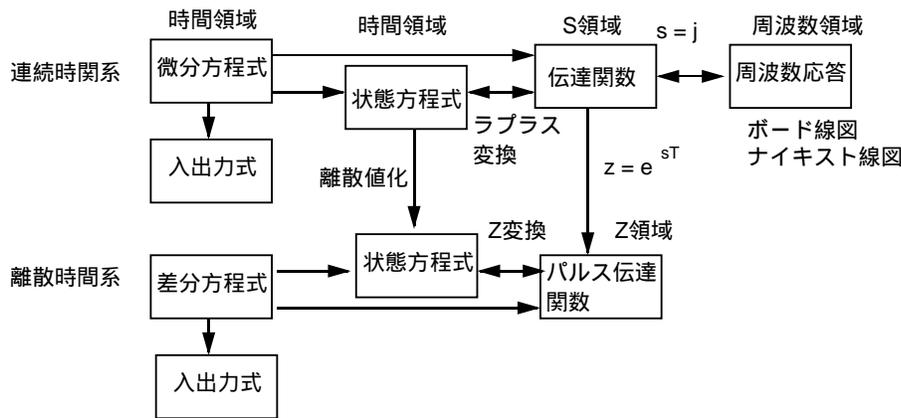


図 2.3: 数式モデル間の変換

いる。

- 伝達関数の代数的性質，特にオープンループ伝達関数の根の存在領域による安定性解析(根軌跡法，極配置法)などがある．最近は伝達関数の代数的性質に基づく方法が研究されている．
- 状態方程式モデルの係数行列間の性質，可制御性，可観測性に基づいての安定性解析も行なわれている．

3) 応答特性：過渡応答を始め各種入力に対するシステムの応答特性の解析には最終的にはコンピュータによるシミュレーションが必要となる．そのためには状態値を逐次的に計算できる状態変数モデルが適している．しかしコンピュータは離散時間系であるので離散時間系で扱う必要があり，連続系の解析の場合は適当なサンプリング時間を定めてモデルを離散時間系に変換して，離散時間系の状態方程式モデルを適用する．

グラフモデル

システム工学では各要素間の情報，信号の伝達を扱うことが多い．このようなシステムの表現方法としてグラフが適している．グラフの優れた点としては

1. 図による表示により，システムが視覚化され，解決策の検討において発想を豊かにできる．
2. グラフは行列との折合いがよく，グラフの内容を行列で表すことが容易であり，行列演算を用いて各種の解析ができる．

グラフは節点(または頂点)の集合と、枝(または辺)とも呼ぶ頂点との対からなる集合から構成される。各種の形態があり、その取扱いは一通りではない。まずグラフのひとつにツリ - (Tree, 木)がある。これはル - プのないグラフの総称で、事故樹解析 (F T A)などに用いられている。ツリ - の一種のバイナリツリ - (二分木)は常に二つに枝分れするツリ - で、イベント・ツリ - 解析 (E T A)などに使用される。

グラフには枝の向きを考慮した有向グラフと、向きを考慮しない無向グラフがある。システム工学では情報の流れを問題にするので、ほとんどは有向グラフが用いられる。前節で有向グラフの一種であるブロック線図やシグナルフロ - 線図について述べたので、ここでは他の例としてネットワークモデルとマルコフモデルについて述べる。

2.3.1 有向グラフによるネットワークモデル

最初に有向グラフ (ダイグラフ, digraph) に関するいくつかの定義を簡単に述べておく。まず、グラフの連結に関して、グラフ内の任意の1点から他の任意の1点に枝の方向に、たどって行ける場合に強連結 であるという。また任意の2点間で一方向にのみ、たどれるときに片方向連結 であるといい、さらに、方向を無視して任意の2点間がつながっている場合に弱連結 であるという。したがって、強連結なら片方向連結であり、片方向連結なら弱連結である。また2節点 i, j 間で、節点 i から節点 j にたどれるときに、節点 j は節点 i から可到達 であるという。

つぎに、有向グラフを行列により表示する方法について述べる。いま n 個の節点を持つ有向グラフがあるとして、この節点間の接続を表す行列として以下に示す $n \times n$ 正方行列を考える。 i 番目の節点から j 番目の節点に向かう枝が存在するとき、この行列の (i, j) 要素を1とし、存在しないとき0とする。これにより1と0よりなる行列、すなわちバイナリ - 行列が得られる。この行列を隣接行列という。一例として図2.4に示す有向グラフの隣接行列を求めると、つぎのとおりになる。

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

このような隣接行列を A として、 $A \cdot A = A^2$, $A^2 \cdot A = A^3$ を求めてみる。この場合の積の演算子“ \cdot ”はブ - ル代数演算則に基づく行列の積を表す。 A, B, C を $n \times n$ の正方バイナリ - 行列とすると、 A, B のブ - ル積 $C = A \cdot B$ はつぎのとおりになる。

$$c_{i,j} = \bigcup_k (a_{i,k} \cap b_{k,j}) \quad (2.30)$$

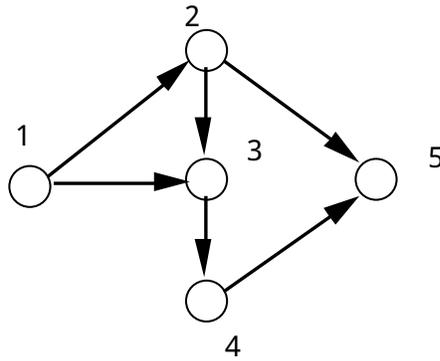


図 2.4: 有向グラフの例

こうして求めた A^2 および A^3 は各節点より枝を 2 回および 3 回たどって到達できる節点の有無を示している。図 2.4 の有向グラフでこれら行列を求めてみるとつぎのようになる。

$$A^2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad A^3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

例えば $A^2(1,3)$ は 1 であり、これは図 2.4 をみると分かるように節点 1 から 3 へ 1-2, 2-3 の 2 つの枝をたどって行けることを示している。もちろん、枝 1-3 により 1 回でも行けるが、これは関係ない。

次に A に単位行列 I をブール代数的に加えた行列, $T = A + I$, を考え, T^2, T^3 を求めると、それぞれ 2 回および 3 回までに、たどれるすべての節点を示している (各節点にとどまることも含め)。何故なら、例えば $T^3 = A^3 + A^2 + A + I$ の各項は順番に 3 回, 2 回, 1 回でたどれるか否かを示しており, I はその位置にとどまることを示しているからである。

したがって $T^{m+1} = T^m = R$ を満足する行列 R を求めると、その各要素は行番号の節点から列番号の節点に到達可能か否かを示している。つまり、 R の (i, j) 要素が 1 ならば節点 i から節点 j にたどれることを意味している。また、この式は m 回までたどれば、それ以上たどる回数を増やしても状況は変化しないことを示している。なお節点の数を n とすると、最大 n 回たどるとすでにたどった節点になるので、 $m < n$ である。こうして求めた行列 R を可到達行列 (reachability matrix) という。 R をみることによりグラフのいろいろな性質が明らかになる。例えば、 R の全要素が 1 の場合は、すべての節点間が互いに可到達であることを意味するので、強連結グラフであることが分る。図 2.4 で可到達行列を求めると、つぎのとおりになる ($m = 2$)。

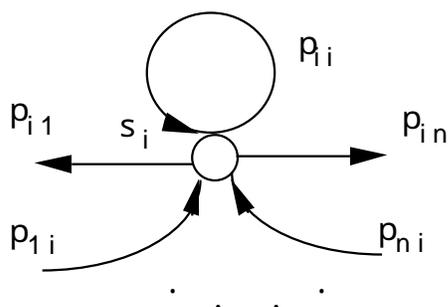


図 2.5: 状態遷移グラフ

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3.2 マルコフモデル

現在の状態が有限回の過去までの状態のみの影響を受け、それ以前の状態の影響は受けないような変化過程をマルコフ過程といい、このようなモデルをマルコフモデルという。特に一つ前の状態のみに支配されるものを単純マルコフという。ここでは単純マルコフ過程をとりあげる。特に断らない限りマルコフ過程とは単純マルコフ過程を意味する。いま、 n 個の状態、 s_1, \dots, s_n よりなるマルコフ過程を考え、状態 s_i より状態 s_j に移る確率を p_{ij} とする。この時つぎの関係が成立する。

$$\sum_{j=1}^n p_{ij} = 1.0, \quad i = 1, \dots, n \quad (2.31)$$

なぜなら現在の状態 s_i からはつぎの時点では s_1, \dots, s_n 中の一つの状態に必ず移るからである。このような確率を遷移確率 (transition probability) という。マルコフモデルは図 2.5に示す有向グラフで表わすことができる。これを状態遷移グラフという。

ここで簡単なマルコフモデルの例としてランダムウォークを取り上げ、どのように利用されるかをみる。いま、幅 16m の道路があり、そこを人が左右によるめきながら歩いているとする。よるめきの状況は、毎分 4m だけ左によるめく確率が 0.5、右によるめく確率が 0.3、その場所にとどまる確率が 0.2 であり、かつ道の両端に来ると溝に落ちてその位置に留まり戻れなくなるとする。ここではこのように 1 分間隔の離散的な事象として考える。

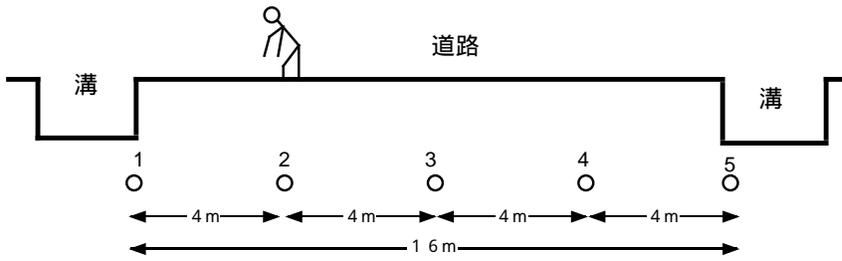


図 2.6: よろめきの状態

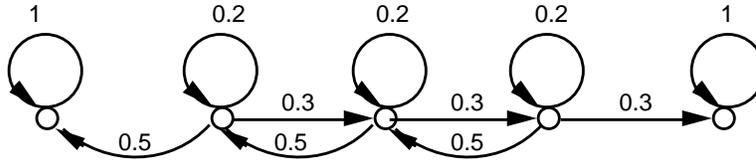


図 2.7: よろめきの状態遷移グラフ

この場合図 2.6 に示すように 5 つの位置で考えることができる．この状況を状態遷移グラフで表すと図 2.7 になる．このグラフを基に基づきのような行列 p を考える．

$$p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.5 & 0.2 & 0.3 & 0 & 0 \\ 0 & 0.5 & 0.2 & 0.3 & 0 \\ 0 & 0 & 0.5 & 0.2 & 0.3 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.32)$$

この行列の要素 $p(i, j)$ は位置 i より 位置 j への遷移確率を表わしている．このように遷移確率により構成した行列 p を遷移行列 (transition matrix) という．式 (2.31) の関係よりこの行列の各行要素の和は 1.0 になる．

つぎにこの人の位置の変化を遷移確率を基に求めてみる．位置としては 5 つあるので，ある時点でそれぞれの位置にある確率を考え，つぎの横ベクトルで表す．

$$x(k) = [x_1(k), x_2(k), x_3(k), x_4(k), x_5(k)]$$

ここで k は k 分後を示し， $x(k)$ の各要素はその時刻にそれぞれの位置にいる確率を表している．初期状態を道路の中央部とすると $x(0) = [0, 0, 1, 0, 0]$ となるので，1 分後に各状態にいる確率 $x(1)$ は遷移確率を使用して，つぎの式で求めることができる．

表 2.3: ランダウオークの経過

時間-k (分)	$x_1(k)$	$x_2(k)$	$x_3(k)$	$x_4(k)$	$x_5(k)$
0	0.0	0.0	1.0	0.0	0.0
1	0.0	0.5	0.2	0.3	0.0
2	0.25	0.2	0.34	0.12	0.09
3	0.35	0.21	0.188	0.126	0.126
4	0.455	0.136	0.164	0.082	0.164
5	0.523	0.109	0.114	0.065	0.188
10	0.686	0.025	0.027	0.015	0.247
20	0.733	0.001	0.001	0.001	0.264

$$x(1) = x(0) p \quad (2.33)$$

また 2 分後の確率 $x(2)$ は同様にして次式で求められる .

$$x(2) = x(1) p = x(0) p^2 \quad (2.34)$$

これを順次繰り返すことにより k 分後での各位置にいる確率は

$$x(k) = x(0) p^k \quad (2.35)$$

となる . 計算の結果の一例を表 2.3 に示す . 20 分後にほとんど , いずれかの溝に落ちてしまっていることが分かり , それぞれにいる確率も示されている . 以上の例に示したようにマルコフモデルによる解析は確率的に発生する事象の取扱いに適しており , 後述のシステム信頼性の解析における冗長系の解析にも使用している .

2.3.3 フローモデル

ここではフローモデルの代表としてシステムダイナミクスに使用されているフローダイヤグラムを取り上げて述べる . システムダイナミクス (SD, System Dynamics) はローマクラブが「成長の限界」[27] の本の中で発表した世界の各種経済社会活動の結果や環境への影響のシミュレーションに採用された方式として一躍有名になった . 社会における諸活動は物 , 金 , 設備 , 人 , 情報などの流れから成り立っており , その間に相互関係があり , フィードバックがなされて変化して行く . システムダイナミクスではこの考えの基にフローダイヤグラムを用いて相互関係を表示し , それをシミュレーション言語 DYNAMO (DYNAic MOdel) を用いてシミュレーションを行なっている [10] .

ここで使用されているフローダイヤグラムはグラフの一種であり , 図 2.8 に示す各種記号を用いてシステムの相互関係を表示する . まず , フローは物 , 資本設備 , 人 , 情報 , 金

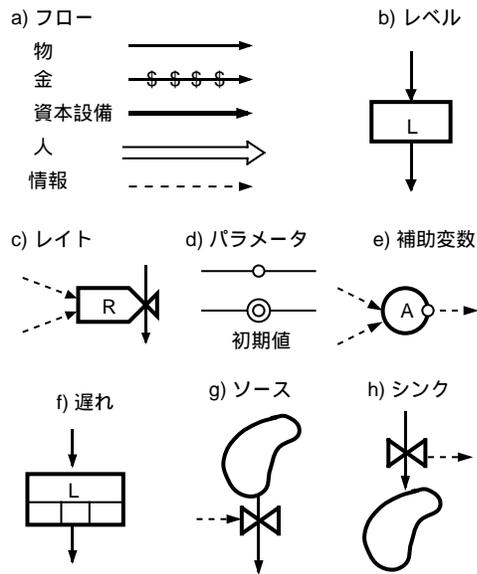


図 2.8: フローダイアグラム用記号

などの流れを示すもので，図に示す以外にも各種の記号が使用されている．システムの状態を表わす状態変数にはレベルを使用する．レートはフローの流れを制御するバルブのようなものである．パラメータは各種の設定（所期値を含む）のための信号を提供する．補助変数は変数の符号を変えたり，加算したりして新たな変数を作るのに使用される．遅れは1次遅れ要素で，これを複数組み合わせると高次の遅れも作ることができる．ソースとシンクはそれぞれ外部からの入力と外部への出力を表わす．

図2.9にフローダイアグラムの1例を示す．a)は1次の正のフィードバックを示す．フローがレベルに送り込まれ，その蓄積すなわち積分値が正のフィードバックとしてレートに送られている．これは，例えば人口が出生率をパラメータに正のフィードバックが行なわれている様の表現に使用できる．

また，b)は2次の負のフィードバックを示している．レート2個を使用して2次の遅れを作っている．補助変数Aは負のフィードバックを作るために符号の変換に使用されている．世界の資本，設備，金，物，人などの動きは相互に正または負のフィードバックをとおして影響し合っているので，これらを組み合わせるとその関係を明らかにできる．

こうして作られたモデルをもとにシミュレーション言語DYNAMOに移し替えてシミュレーションを行なっている．このようなプログラムとして必ずしもDYNAMOを使用する必要はなく，他の言語で作成することもできる．すでにBASIC言語で作成した例が発表されている[10]．

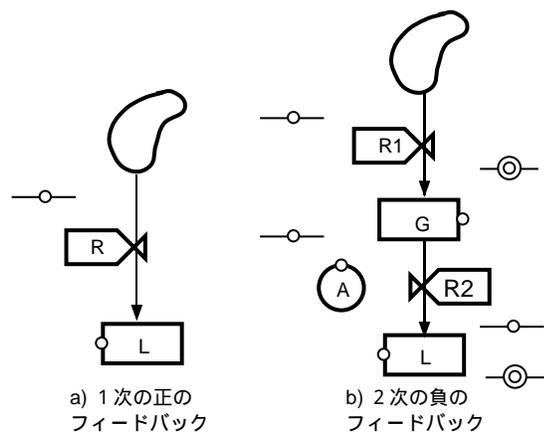


図 2.9: フローダイアグラムの例

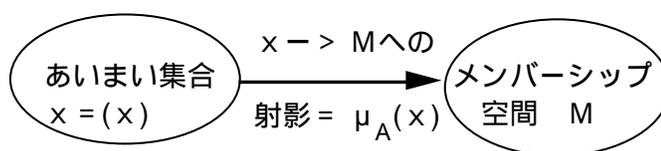


図 2.10: ファジー集合

ファジーモデル

人間の判断や社会システムなどでは従来の1,0の二値論理で割切れないものが多い。例えば、青年という集合を考えてみたとき、年齢30歳はこの中に入るかどうかと問われると、いずれともいい難い。このような場合、集合に入るか、入らぬかとの二値ではなく、その中間的な状態を認めることが考えられる。このような集合をファジー (fuzzy, あいまい) 集合という。そして、その集合に属する程度を表すものとして、0から1の間の値をとるメンバーシップ関数 (membership function) を定義する。その関係を図2.10に示す。ファジー論理はこのファジー集合間の関係を規定するもので、メンバーシップ関数を用いてつぎのように定義する。

$$\text{包含} \quad A \subseteq B \quad \iff \mu_A(x) \leq \mu_B(x) \quad (2.36)$$

$$\text{等値} \quad A = B \quad \iff \mu_A(x) = \mu_B(x) \quad (2.37)$$

$$\text{補集合} \quad \bar{A} \quad \iff \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.38)$$

$$\text{和} \quad C = A \cup B \quad \iff \mu_C(x) = \max[\mu_A(x), \mu_B(x)] \quad (2.39)$$

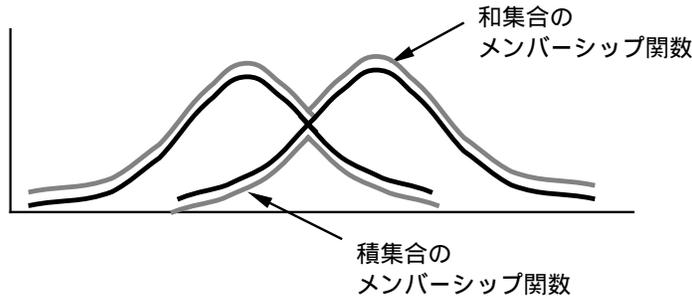


図 2.11: ファジー演算

$$\text{積} \quad C = A \cap B \iff \mu_C(x) = \min[\mu_A(x), \mu_B(x)] \quad (2.40)$$

この内、和集合と積集合の関係を図示したのが図 2.11 である。ここでメンバーシップ関数値 $\mu_A(x)$, $\mu_B(x)$ などは、いずれも 0 と 1 との間の値 $[0, 1]$ を取る。包含関係を例にとると、ファジー集合 A が集合 B に含まれるとは A のメンバーシップ関数値 $\mu_A(x)$ が B のメンバーシップ値 $\mu_B(x)$ 以下であることと定義している。このようにファジー集合ではその間の関係をメンバーシップ関数値の大小関係と補数関係で表わしている。

ファジーモデルの 1 例として、故障診断に用いられる因果モデルを考えてみる。まず、原因と症状との関係をつぎのとおり仮定する。

1. 原因がなければ症状は現れない。
2. 1 つの原因にはそれに関連したすべての症状が現れる。
3. 1 つの症状はそれに関連した多くの原因のうちの 1 つが生じても現れる。
4. 2 つ以上の原因が同時に起こった場合、症状は個々の原因がもたらす症状の最大値で決まる。

この仮定の基で故障の因果関係を以下のおりモデル化する。いま、 m 個の事故と n 個の症状があるとして、つぎのファジー集合、 X, Y を考える。

$$X = \{X_i \mid i = 1, \dots, m\}$$

$$Y = \{Y_j \mid j = 1, \dots, n\}$$

これらの原因と症状のメンバーシップ関数 $\mu_{f,i}, \mu_{s,i}$ をつぎのとおり $[0, 1]$ 間の値に定める。

$$\mu_{f,i} : X_i \rightarrow [0, 1] \quad i = 1, \dots, m$$

$$\mu_{s,i} : Y_i \rightarrow [0, 1] \quad i = 1, \dots, n$$

原因群のメンバーシップ関数を要素とするベクトルを a で表し、同様に症状のメンバーシップ関数のベクトルを b で表わす。つぎに各原因が各症状に与える影響の程度を表わすファジー関数を考え、それらのメンバーシップ関数を要素とする行列を R とする。これらを用いて先に仮定した因果関係はつぎのとおりモデル化できる。

$$a \cdot R = b \quad (2.41)$$

この演算子“ \cdot ”の意味は

$$\bigvee_i (a_i \wedge r_{ij}) = b_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n \quad (2.42)$$

但し \bigvee は最大、 \wedge は最小を意味する。式 (2.41) の逆演算を行えばファジー関係 R と症状 Y とより事故原因を求めることができる。この逆演算はかなり複雑になりこの本の範囲を超えるので、詳細は文献 [26] に譲る。

シミュレーション

システムを計画する際に、どのような方式を適用したらよいか、適用すべき方式についていくつかの候補がある場合はどれを採用すべきか、さらに方式が決定した場合に所期の性能が期待できるか、などの色々な場面でシミュレーションが行なわれる。一般に性能の確認には数式などを用いた解析で求めることもできるが、システム工学で扱うような複雑な問題の解析では限界があり、多くはシミュレーションに頼ることになる。

シミュレーションにはかつては実時間性に優れ、操作が簡単なアナログ計算機によるアナログシミュレーションが用いられたが、精度、プログラムの再利用性などに難点があり、今日ではほとんど使用されず、デジタル計算機によるデジタルシミュレーションが適用されている。したがってここではデジタルシミュレーションを対象にして述べる。

まずシミュレーションの利点を考えてみると次のようなものがある。

- 1) 実システムにおける実験などに比較し、費用、時間、人手が少なくすむ。
- 2) 時間の領域を変更でき、時間を短縮したり、伸ばしたりできる。年単位の現象を秒単位に実行したり、逆に 1 秒以下の短い現象を時間を伸ばして観測しやすくなる。
- 3) パラメータの範囲を大きく変えて、各種の状態でテストできる。
- 4) 安全問題における事故など、現実には不可能な現象をテストできる。
- 5) 乱数や雑音を含む確率システムの検証はシミュレーション以外は不可能である。

このような利点がある一方、取扱いを誤ると現実とは異なった状況を作りだし、その結果間違った結論を得る危険性もあるので、注意が必要である。シミュレーションを実施するに当たって注意すべき点としては次のものがある。

1) モデルの適切な選定. シミュレーションの目的に合ったモデルを選定する必要がある. 実際のシステムを忠実に再現したモデルが最良とは限らない. そのようなモデルでは場合によってはシミュレーションに多大の時間が掛かることも有り得る.

2) モデルには程度の差はあるが, 近似化がなされている. 例えば, 非線形システムの線形化, システム次数の低下や一部のパラメータの無視などがある. これがシミュレーションの結果に如何に影響するかを予想して対策を立てて置く必要がある.

3) デジタル計算機では離散的に計算が行なわれるので, 連続現象のシミュレーションでは連続システムとしてシミュレーションできるように数学的な変換を行なうか, あるいは離散時間化して扱う必要がある. 後者の場合, 離散時間を如何に選定するかの検討を大事であり, 特に確率的な現象の場合に注意が必要である. また使用している乱数系列が果たして本当に乱数として働いているかなど, その性質にまで配慮する必要がある場合がある.

さて, 実際にシミュレーションを行なうときの手段について考えてみよう. 方法としては次のようなものがある.

1) 比較的簡単な場合はシミュレーションに適した計算機言語を使用して行なう. 例えば, Mathematica, REDUCE などの数式処理言語を使用して直接, 数式を実行する. また行列演算の場合は MATLAB が適している. MATLAB にはプログラムパッケージとして各種のツールボックスが揃っているので, それが利用できる.

2) シミュレーション用として開発された各種のシミュレーション言語を利用する. 例えば, 離散系のシミュレーションでは GPSS(待ち行列を含むモデル用) や SIMSCRIPT(経営の意志決定用) など. また, 連続系では CSMP(動的システム用) や DYNAMO(経済, 資源, エネルギーなどのフィードバック機能をもつシステム用) などがある.

3) その他は目的に応じて作成する.

いずれを採用するかはシミュレーションの目的, 規模, 内容に応じて決定すればよい.

演習問題

2.1 数式モデルとグラフモデルの長短を比較してみよ.

2.2 つぎの行列で表される有向グラフを図で示し, 可到達行列を求めよ.

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2.3 本文のランダムウオーク問題で最初の位置が道路の中央ではなく左側, すなわち $x(0) = [0, 1, 0, 0, 0]$ のときの 1, 2, 3, 4, 5, 10, 20 分後の各位置にいる確率を求めよ.

2.4 つぎのランダムウオークの問題に答えよ.

幅 8m の道路があり，そこに人が左右によろめきながら歩いている．中央では毎分 4m だけ左によろめく確率が 0.2，右によろめく確率が 0.4，その位置にいる確率が 0.4 とし，両端では 4m 戻る確率が 0.6，その位置にいる確率が 0.4 とする．なお，道路上の位置としては中央，左および右端の 3 箇所とする．

- 1) 状態遷移グラフを作成せよ．
- 2) 遷移確率を求めよ．
- 3) 最初に中央にいたとして 1, 2, 3, 10 分後の中央，左端，右端にいる確率を求めよ．

システムの計画技法

[概要] システムの計画に使用される各種の技法について説明する．先ず K J 法，シナリオ・ライティング，デルファイ法などの発想・予測技法，システム評価技法，そして最後にテロテクノロジーについて述べる．

発想・予測技法

3.1.1 K J 法

川喜田二郎氏により発想法として開発された技法であり，システムの分析などにも有効な方法である [7]．システムには構造が明確でない，いわゆる悪構造のものが多い．このような対象での問題点の発見，分析には通常の論理的方法とは異なった方式，例えば，ここで説明する K J 法が有効である．

K J 法はつぎのステップで進められる．

1. 情報の収集：自由な発想のもとで，考えられるすべての情報を集め，簡潔ではあるがあまり抽象化し過ぎない表現で，1 件 1 葉のカ - ドに記入する．ブレ - ンスト - ミング などにより，できるだけ多く集めるのが重要である．ブレ - ンスト - ミングは
 - 他の人のアイデアは批判しない．
 - 他の人のアイデアをさらに発展させるのがよい．
 - 短時間でできるだけ多くのアイデアを出す．

などのルールに基づき 10 人内外の人々でアイデアを出し合う方法である．

2. グル - プ化：上記により作成したカ - ドをテ - ブル上に広げ，それを眺めつつ互いに似たもの，親近感のありそうなもの同士を一つにまとめてゆく．このまとめ方は論理的ではない点が重要であり，こうして発想をうながす．まとめ方も小分けから大分けに進むのが大切である．こうしてグル - プ化したものに 1 行見出しをつける．なお，すべてをグル - プ化する必要はなく，グル - プ化できないものはそのまま残して置く．このような離れ小島になったものも新しいアイデアを生む上で大切であると言われている．

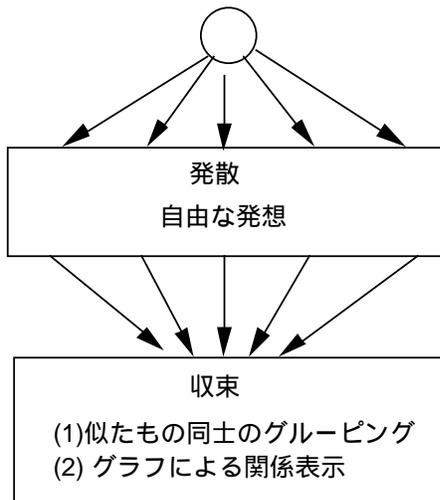


図 3.1: K J法の考え方

3. 図解：グル - プ化したカ - ドを大きな白紙のうえに配置し，グル - プ同士は輪で囲んだり，関係のあるもの同士の間は線で結ぶなどして図で表す．
4. 文章化：図解したものをもとに文章にまとめる．

この方法は図 3.1 に示すように大きく分けて二段階より成立っている．最初が発散段階で，できるだけ広く，自由な発想のもとで情報を集める．第二段階は収束段階で，ここではグル - プ化による整理と相互関係の図による視角化が行われる．このグル - プ化の際にも自由な発想を働かせている．図 3.2 に一例として就職先選定に関して考えるべきことを K J 法でまとめたものを示す．

近年，この方法をコンピュータ上に実現して発想支援システムとして利用する研究が進んでいる．カードの作成，グループ化なども C R T 画面上で簡単に行なうことができる．発想のためには全部のカードを見渡せることが大事であり，そのためには大型画面が必要となるが，これが可能となりつつあり，今後に期待できる．

3.1.2 シナリオライティング

将来の予測を文章により記述する手法で，ハ - マン・カ - ン氏が紀元 2 0 0 0 年のシナリオで使用して有名になった．シナリオを書く際，現在からの類推とともに，今後予想される科学・技術や社会情勢の変化などを考え，筋道の立ったひとつの物語としてまとめる．文章化する過程で論理的に練られ，整合性を確認などが行われ，さらに，その結果が

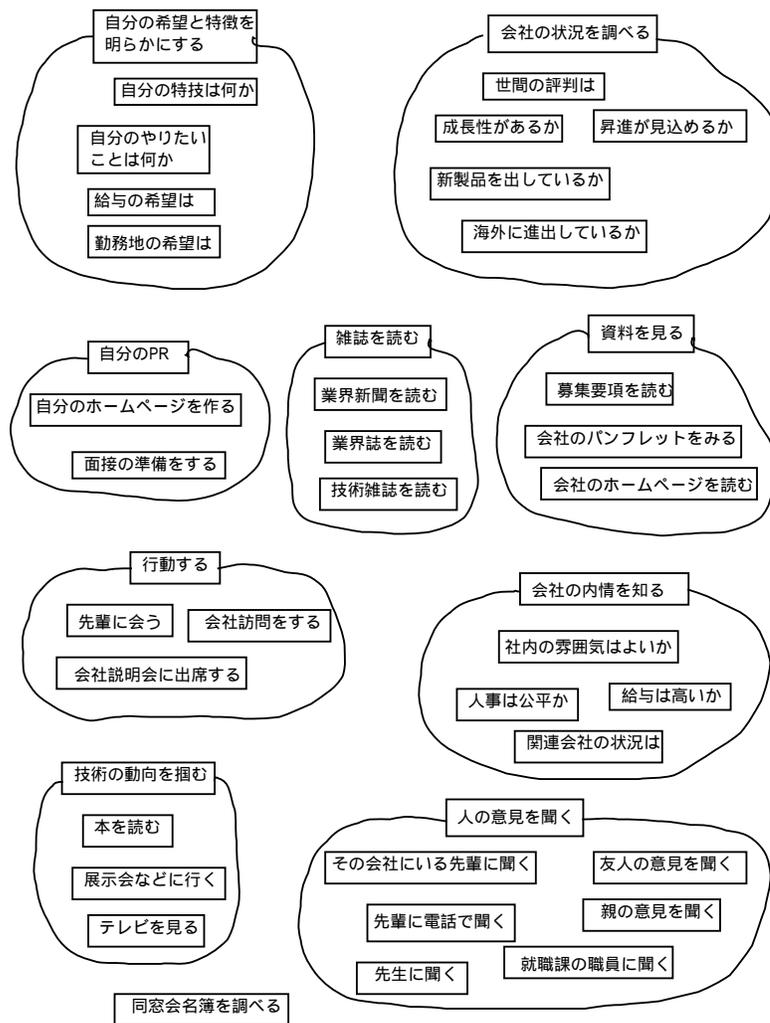


図 3.2: 就職先選定の図解 (K J 法)

シナリオとしてイメージ化されるので強力な説得力を持つ。時にはシナリオは一つだけではなく、いくつかの情勢の変化を折込んだ複数のシナリオを作ることのできるなどの融通性もある。いずれにしても文書を用いるので画一的ではなく、問題に応じて自由に記述できるのも強みである。

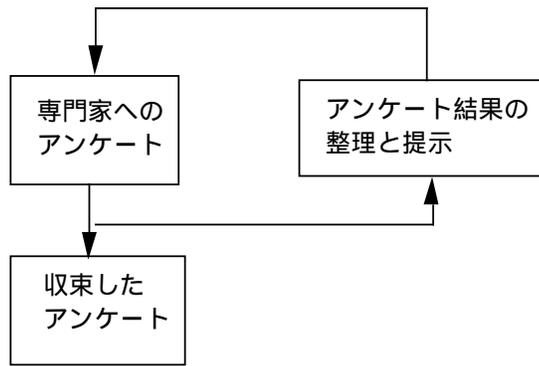


図 3.3: デルファイ法

3.1.3 デルファイ法

多数の専門家により将来を予測する方法であり、通常のアンケート法にブレ-ンスト-ミングの長所を取入れたものといえよう。図 3.3 に示すように、先ず多数の専門家に同一のアンケートを行い、その結果を整理する。つぎに、その結果を呈示してそれを参考にして貰いながら、再び同じアンケートの回答を求める。これを数回繰返すことにより、最終的な収束した結果を得る方法である。

この方法の長所は、回答結果のフィ-ドバックにより、他の専門家の意見を取入れたり、自説の不十分なところを補ったり、あるいは、気付かなかった点を教えられる、などにより修正が行われることであり、さらに、会議方式などで陥りがちな、権威者の意見に引きずられることや自説への固執をなくして自由に意見を出せること、などが挙げられている。将来の予測のようなアンケートでは最初は意見のばらつきが大きいのが、アンケートを繰り返すことにより、やがて意見がかなり一致し、参加者の納得も得やすい。一方、この方法の短所は何回もアンケートをしなければならぬため時間と費用が掛かることであり、それを考慮しても効果のある場合に適している。

評価技法

3.2.1 一対比較

評価においてはいくつかの代替案の中から最適なものを選択することが行われる。このように多くの代案の中から一つの案を選ぶ場合、各種の評価尺度があり、なかなか選び難い。一般に、3 個以上から 1 個を選んだり、あるいは順序づけるのは互いに目移りして難しいが、2 個のものから 1 個を選ぶのは比較的易しい。従って、この性質を利用し、多

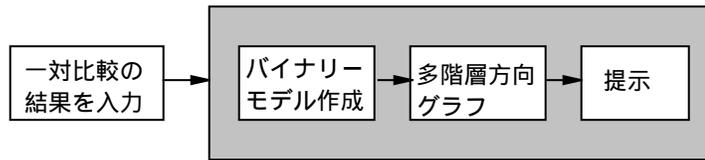


図 3.4: ISM手法の流れ

数のものから1個を選んだり、あるいは順序づけをする場合に、これを2個のものから1個の選択することの組合わせに変えることにより、円滑に行わせようというのが、この一対比較の考え方である。構造の明確でない複雑な社会問題などを扱う際にはこのような方法が必要になる。

一対比較の組合わせによる構造化を支援して、この作業を円滑にさせるものとして、バッテリー・コロンバス研究所で開発された、ISM手法(Interpretive Structural Modeling)がある。このISM手法はグラフ理論をもとに作られたもので、コンピュータより次々に2つの要素間の比較が質問されるので、一対比較の結果を入力することにより、各要素間の構造を方向性グラフとして構築してゆく。これにより不必要な質問をなくすることができる。その結果、全体の構造がグラフとして図示されるので、それにより、さらに検討ができる。これらの過程を図3.4に示す。

つぎに簡単に数学的な背景を説明する。いま対象とするシステムが n 個の要素より成立しているとし、その n 個の要素よりなる集合 S を考える。 S の中の2個の要素 s_i, s_j 間に2項関係 R が成立つとして、それを

$$s_i R s_j \quad (3.1)$$

で表す。例えば R が“好み”の関係を表す場合、要素 s_i より s_j の方が好ましいというのを上記の通りに表せる。ここで一つ的前提として推移律が成立するものとする。これは任意の3要素 s_i, s_j, s_k の間で

$$s_i R s_j \text{ および } s_j R s_k \Rightarrow s_i R s_k \quad (3.2)$$

を意味している。

グラフ理論を用いると、これらの関係は要素を頂点に、関係を辺にした有向グラフで表せ、さらに行列で表すこともできる。集合 S の要素数に等しい行数と列数の正方行列 C を考え、要素 s_i, s_j 間に関係 R が成立すれば C の要素 $C_{i,j} = 1$ とし、成立しない場合は $C_{i,j} = 0$ とする。

まず、次式を満たす行列、すなわち可到達行列 T を求める。

$$(C + I)^{m+1} = (C + I)^m = T \quad (3.3)$$

ここで I は単位行列を表す。

つぎに行列 T から多階層方向性グラフを求める方法を述べる。まず最初につぎの集合を定義する。

$$R_i = \{s_j \in S \mid T_{i,j} = 1\}, \quad i = 1, 2, \dots, n \quad (3.4)$$

$$A_i = \{s_j \in S \mid T_{j,i} = 1\}, \quad i = 1, 2, \dots, n \quad (3.5)$$

ここで集合 R_i は要素 s_i から到達できる、自分自身を含むすべての要素から成り立っており、行列 T の i 行より求めることができる。集合 A_i は要素 s_i に到達できる、自分自身を含むすべての要素から成り立っており、行列 T の i 列より求められる。

つぎに R_i と A_i のブール積 $R_i \cap A_i$ を作り、

$$R_i \cap A_i = R_i \quad (3.6)$$

を満足する要素 s_i の集合を考えると、これは他のどの要素にも到達できない要素、すなわちトップの要素であることが分る。

つぎに、行列 T からトップの要素に相当する行と列を除いたのち、以上の計算を行うとトップから2番目の要素が得られる。これを繰返すことにより、全要素の順序付けができる。一例として図3.5に示すグラフについて計算すると C, T は次式のとおりになり、 T より求めた表3.1に式(3.6)を適用することによりトップが要素4であることが分る。

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad C + I = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(C + I)^2 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T = (C + I)^3 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

以上に述べたように一対比較法は推移律が成立するとの前提の基に作られている。一般に多数の評価尺度をもつものに対し推移律を期待するのは難しい場合が多く、この方法の適用を困難にしている。ISM法では推移律が成り立つものとしてコンピュータ内の処理を行ない、その結果により質問を出しているため、このような問題は生じない。

3.2.2 マトリックス法

多数の代替案から選択する方法として一般に使用されているものにマトリックス法がある。代替案を評価すべき項目をあらかじめ定め、代替案とともに表3.2を作成し、各欄に

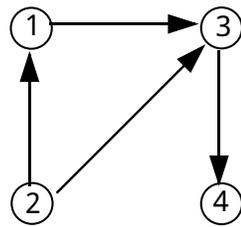


図 3.5: グラフの例

表 3.1: 計算結果

S_i	R_i	A_i	$R_i \cap A_i$
1	1 3 4	1 2	1
2	1 2 3 4	2	2
3	3 4	1 2 3	3
4	4	1 2 3 4	4

表 3.2: マトリックス法

	評価値				
	項目 1	項目 2	項目 3	項目 4	合計
代替案 1					
代替案 2					
代替案 3					

評価値を記入してゆく．そして全評価値の合計値の高いものを採用する方法である．評価値の決め方としては評価項目の重要性をどのように考慮するかによりつぎの二つの方法がある．

1. 方法 1 : 評価値自体に評価項目の重要性を反映させ，重要な項目ほど評価値を高く設定する．重みを考慮しながら自由に設定できる利点がある．
2. 方法 2 : 各項目の評価値は同一基準（例えば $[0, 1]$ ）で定める．別に各項目ごとに重要度に応じた重みを設定し，これを用いて合計値を重み付き加算で求める方法である．重みの設定により採用される代替案が異なってくるので，しばしば重みを変え

とすることが行なわれる。この際、重みにより表を作り直さなくてすむ利点がある。

いずれの方法でも各項目の評価値や重みは代替案の採用に大きく影響し、かつ評価者の意向に基づくので、会議などでは参加者の合議で決めて行く場合が多い。

3.2.3 デシジョンツリ -

意思決定は多くの場合、不確実な事実に基づいて行わねばならない。この際、できるだけ利益を大きく、危険や負担を小さくするように決定するのが合理的であろう。このような考えに基づいて、不確実な事象に対して生起確率を予想し、価値の期待値を求めて、常に期待値の大きい方を採用するよう意思決定する統計・確率的な方法がこのデシジョンツリ - である。

デシジョンツリ - の一例を図3.6に示す。これはある商品を「一旦、試作してから商品化するか」、あるいは「試作なしに商品とするか」の意思決定の例である。後者の場合は試作費は助かるが、失敗した場合の損失は大きくなる。図において四角は意思決定を示し諸否により二つに枝分れしており、丸印は確率的な事象を示し推定確率値により同じく二つに枝分れしている。ツリ - の末端(図の右端)の数値は最終段階の価値を示し、マイナスはそれだけの損失を生じることを意味する。この例では試作費と失敗した商品の費用がマイナス要因となっている。

意思決定の手順としてはツリ - の末端より頂点に向かってたどり、意思決定では期待値の大きい方を採用する。先ずC点での期待値を、確率値をもとに計算すると $1億9000万円 \times 0.2 - 6000万円 \times 0.8 = -1000万円$ となる。したがって、意思決定点2ではこの金額と一方の案「商品化を中止」の価値 $-6000万円$ (実際は損失)と比較して、その大きい方(損失の少ない方)の「試作をする」案を採用する。したがってこの点での期待値は $-1000万円$ になる。以下括弧()の中に期待値を示す。つぎのA点では期待値は前と同様に計算すると $1億1600万円$ ($2億円 \times 0.6 - 1000万円 \times 0.4$) となる。一方、B点の期待値は $1億3000万円$ ($1億9000万円 \times 0.7 - 1000万円 \times 0.3$) となるので意志決定点1での意思決定は「試作をする」案を採用することになる。こうして意志決定の手順が求められる。

現実の問題では時間や人員、設備そのほか種々、考慮せねばならない事項があり単純ではなく、また各確率値をいかに設定するかの難しい問題もあるが、このような費用に換算して決定するのも合理的な方法のひとつである。

以上に述べた方式では意思決定者の好みを反映していない。平均的な行き方を好む人のほか、危険をできるだけ回避したい人や、逆に勝負に賭けたい人もある。勝負にかける人は金額の大きな事項に対してはより大きなウエイトを置きたいであろう。このような意思決定者の好みを考慮した選好曲線(金額対選好尺度)を作成して、期待値を求める際にこの選好尺度を使用して、金額を意志決定者の好みに応じて補正し、それに沿った決定がで

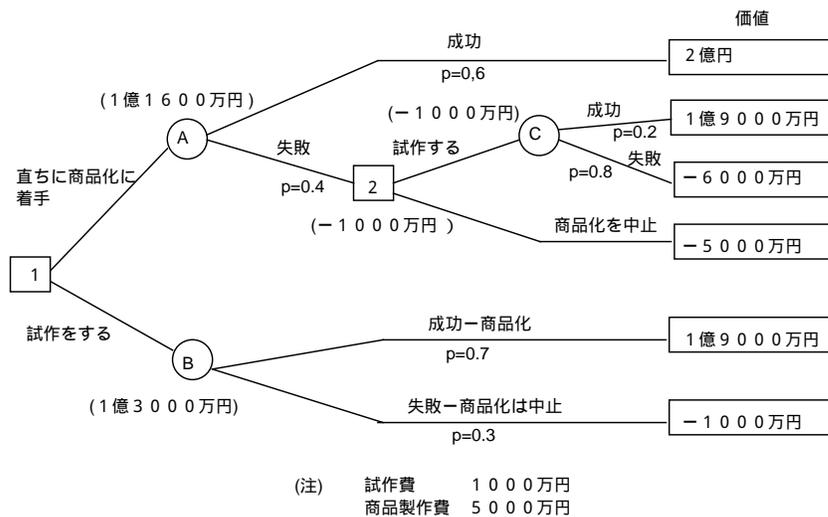


図 3.6: デシジョンツリーの例

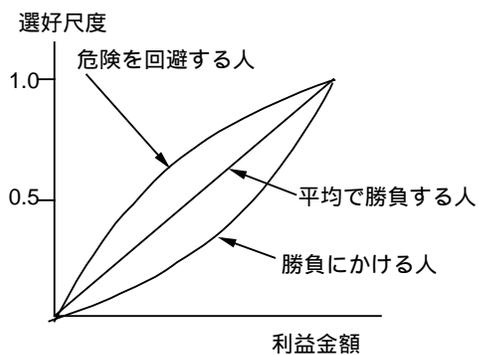


図 3.7: 選好曲線

きるようにしている。選好曲線の一例を図3.7に示す。勝負に賭けたい人は金額の高い方より比重を置くことを示している。

3.2.4 テクノロジ - アセスメント

最近のシステムは大形化により社会や自然に対し大きな影響を及ぼし、それを解決するのに多大の努力を必要とするようになってきた。このような事態を未然に防止することを目的とした事前評価がテクノロジー - アセスメントである。したがって、システムの及ぼ

す負の効果を中心に分析し、その結果によっては必要な対策を施したり代替案に変えたりする。テクノロジー - アセスメントは次の手順で行われる。

1. 対象とするシステムの範囲とその目的を明らかにする。
2. システムの詳細な内容を分析する。
3. システムの与える影響、特に負の影響を分析・評価する。
4. 対策案や代替案を検討する。
5. 総合的に評価する。

最近では巨大システムに対してはテクノロジー - アセスメントを行うのが普通になってきた。

ライフサイクルコスト

システムを構築する際、性能もさることながら、コストの最小化も図らねばならない。そのコストとして建設費などの初期投資にのみ目を奪われ、動力費や保全費などが軽視される場合が多いが、投資コストに対する効果の点からはそのシステムの計画から廃棄されるまでの全期間、すなわちライフサイクルでのコストを最小にすべきであろう。これを主張するのがテロテクノロジー (Terotechnology) であり、ロジスティックス (Logistics) である。システムの計画時には基本思想としてこれらを持って置くべきだろう。

テロテクノロジー - の用語は 1970 年に英国で定められたもので

ライフサイクルコストを最も経済的にするために有形資産に適用されるエンジニアリング、マネ - ジメント、財務そのほかの実務を総合したテクノロジー -

とされている。また、ロジスティックスは米国で提唱されたもので

システム、製品の使用中の効果を最大にする諸資源の費用を最小にすること

と定義されている。ここでもライフサイクルコストを対象としている。

さて、ライフサイクルコストは、図 3.8 に示すようにシステムの研究・開発段階に始まり、設計・製作・試験・据付け・配線・試運転を経て運用・保全段階に入り、廃棄されるまでの全期間の費用で、それには機器、建物の費用に始まり土木費、人件費、予備品費、修繕費などすべてが含まれる。また、費用の発生時点がそれぞれ異なるので、各種の代案の費用を比較する際は金利に対する補正を必要とする。これには 2 案があり、第 1 案は金利と稼働年数とから計画初期時点での総価格に換算する方法であり、代案は逆に廃棄時点の総価格に換算する方法である。

各種の代案を比較する際に建設費用のみを比較する場合が多いが、このライフサイクルコストにより評価する方法がより合理的である。

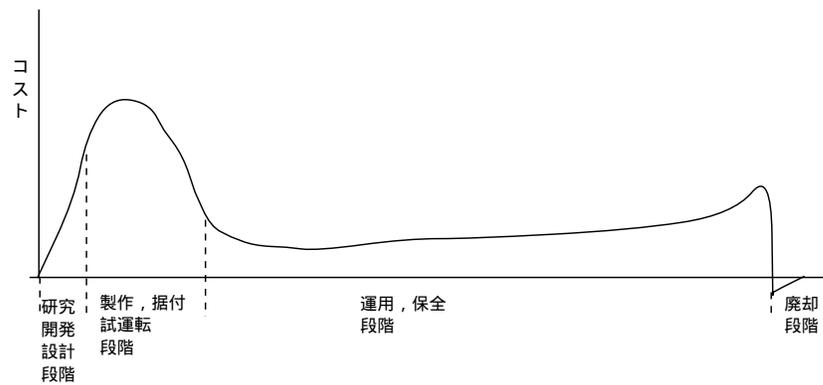


図 3.8: ライフサイクルコスト

演習問題

- 3.1 地球の環境問題について我々ができることをKJ法で考えてみよう。
- 3.2 現在住んでいる地域の50年後の状態に関してシナリオ・ライティングにより予測して見よ。
- 3.3 就職会社の選定に際し評価すべき事項を挙げて、一対比較により評価項目を構造化し、図に表わせ。
- 3.4 主要な海外旅行先をあげ、一対比較により順序づけを行え。

線形計画法

[概要] システム工学の中でも最も重要な位置を占めているのが最適化手法である。最適化手法には各種あるので、ここではまず線形計画法について述べる。基本の理解に適した図的解法、実用的なシンプレックス法、初期値の状況によっては必要となる2段階法、および双対問題について記述する。

はじめに - 最適化手法について

システムは共通の目的をもって構成されたものであるので、その目的を最も効果的、すなわち最適に達成することが求められる。その最適の内容はシステムの種類により異なり、例えば最小の費用、最短の時間、最高の効率、最大の生産量、あるいは、これらを組合わせた総合的な指標などが考えられる。このように最適の内容が各種あることと、システムの種類が多様であることから、最適化の手法には多くの種類がある。ここではその内の代表的ないくつかについて説明する。最適化を論ずる時には、その指標として何を選ぶかを明確にしておく必要がある。それを表すものとして目的関数（あるいは評価関数）を定め、システムの行動を規制する制約条件のもとで、その最小値あるいは最大値を得る方策を求めることにより最適化を達成する。

まず、最適化手法の代表的なものに数理計画法がある。これは「与えられた制約条件のもとで、ある目的関数を最大（または最小）にする解を見出だす方法」と定義されており、これには線形計画法、非線形計画法、動的計画法などがある。線形計画法は制約条件式と目的関数が線形代数で表せられるもので、比較的簡単に解けるところから広く用いられている。これに対し条件式と目的関数のいずれか、あるいは両者が非線形であるものが非線形計画法であり、その非線形の式の種類により整数計画法、二次計画法など各種の方式に分れる。本書では非線形計画法のひとつの勾配法について述べる。動的システムのように時間とともに状態が推移して行くシステムにおいて最適化を達成するためには、時間の進行に応じて次々に決定をして行かねばならない。この目的に開発されたのが動的計画法である。

また最近はニューラルネットワークを用いた方法、遺伝的アルゴリズムなど生物の機能を模倣した方式が研究されている。特に後者はモデル化できない対象に適している。これ

表 4.1: 合金製造計画

合金の種類	原料の組成比			合金1トン当りの 利益(千円)
	M_1	M_2	M_3	
A_1	0.4	0.4	0.2	4
A_2	0.2	0.4	0.4	5
原料の在庫(トン)	8	10	8	

らについても述べている。

最後に多目的最適化問題について述べる。

線形計画法とは

線形計画法(LP:Linear Programming)は「いくつかの線形の等式または不等式の制約条件式のもとで1次の目的関数の値を最大(または最小)にする手法」である。ここでは、まず図的解法について述べた後に、その代表的な解法であるシンプレックス法(simplex method)について説明する。つぎに初期解を与えることができない場合に必要となる2段階法について述べる。最後に双対問題について触れている。1984年にカーマーカー(Karmarkar)により内点法が発表され、現在活発な研究が進められているが、一般的に広く用いられているのはシンプレックス法であるので、ここでは内点法には触れない。

図的解法

合金の製造計画を例に説明する。3種類の金属原料 M_1, M_2, M_3 を混ぜて製品の合金を製造する際に、各組成比を変えることにより2種類の合金 A_1, A_2 を作ることができる。その各々の組成比、各合金の1トン当りの利益および各原料の在庫量を表4.1のとおりとする時、利益を最大にするには合金 A_1, A_2 をいくらずつ製造すればよいかという問題を考えてみよう。製品の合金 A_1, A_2 の重量をおのおの x_1, x_2 トンとすると、次の条件式と目的関数が得られる。この目的関数を最大にする解が求める解となる。これらの式はいずれも線形式となっており、したがって線形計画法の問題である。

$$\text{制約条件式 } 0.4x_1 + 0.2x_2 \leq 8$$

$$0.4x_1 + 0.4x_2 \leq 10$$

$$0.2x_1 + 0.4x_2 \leq 8$$

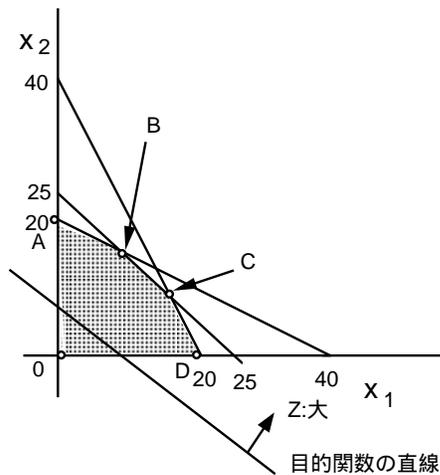


図 4.1: 制約条件の範囲

$$x_1 \geq 0, x_2 \geq 0$$

$$\text{目的関数 } z = 4x_1 + 5x_2$$

上記の関係を図示したのが図 4.1 である。図で斜線で示した範囲が制約条件を満足する領域を示している。また目的関数は x_1 軸, x_2 軸との交点がそれぞれ $z/4$, $z/5$ である直線になるので, z の増加により直線は右上に移動する。したがって目的関数を最大にする状態は目的関数を表す直線が斜線の領域と接する B 点であることが分かり, 最適解は $x_1 = 10$, $x_2 = 15$ となる。このように作図により最適解を求める方法が図的解法である。図的解法は視覚に訴えるので理解し易いがせいぜい三次元までしかできない。したがって変数の数が 4 個以上になると適用できないので, 実用上はつぎのシンプレックス法が使用される。

シンプレックス法

線形計画法を適用する問題は一般に次の形をとる。

$$\text{制約条件式 } a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n (\leq)(\geq)(=)b_1 \quad (4.1)$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n (\leq)(\geq)(=)b_2$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n (\leq)(\geq)(=)b_m$$

$$x_i \geq 0, i = 1, \cdots, n$$

$$\text{目的関数 } z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \quad (4.2)$$

において目的関数の値を最大または最小にする $x_i, i = 1, \dots, n$ を求めることである．ここで $(\leq), (\geq), (=)$ はこれらの演算子のいずれかを適用する意味を表す．

なお非負条件のない変数 x_i に対しては2つの非負変数 $x_i^+ (\geq 0), x_i^- (\geq 0)$ を導入し，それらの差 $x_i = x_i^+ - x_i^-$ に置き換えればよい．

この問題はつぎの標準形式の問題に変形することができる．ただし n や各係数 $a_{i,j}, c_i$ などの値は上式とは異なる．

$$\text{制約条件式} \quad a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \quad (4.3)$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

$$x_i \geq 0, i = 1, \dots, n, \quad \text{但し } b_j \geq 0, j = 1, \dots, m$$

$$\text{目的関数} \quad z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (4.4)$$

において目的関数の値を最大にする $x_i, i = 1, \dots, n$ を求める問題．

[標準形式への変換] この標準形式への変形はつぎの最大4ステップにより行うことができる．

ステップ1 目的関数を最小にする問題の場合は， z の代わりに次式のようにその符号を逆にした z' を目的関数に選び，目的関数を最大にする問題に変える．

$$z' = -z$$

ステップ2 制約条件式において不等号 (\geq) が用いられている式には，その式の両辺に (-1) を乗じることにより，すべての式の不等号の向きを \leq に統一する．

ステップ3 非負のスラック変数を導入して不等号 (\leq) をなくして等号にする．例えば

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$$

なら非負のスラック変数 (slack variable) x_{n+1} を追加して

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} = b_1$$

とすることにより等号をなくせる．またこの時の目的関数は

$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n + 0 \times x_{n+1}$$

のようにスラック変数についての係数を零にしておけば目的関数への影響はない．

ステップ4 右辺の b_j を非負にする．もし負のものがあれば両辺に (-1) を乗じて正にする．

以上の結果，線形計画法の問題は行列表示を用いて

$$\text{制約条件} \quad Ax = b \quad (4.5)$$

$$x \geq 0 \text{ 要素のすべてが非負, } b \text{ の各要素は非負}$$

のもとで

$$\text{目的関数} \quad z = c^T x \quad (4.6)$$

を最大にする x を求めることになる．ここで $x \in R^n, b \in R^m, c \in R^n, A \in R^{m \times n}$

[解の存在] 上記で与えられた問題の解はつぎのいずれかになる．

1. 制約条件を満足する領域が存在せず解が得られない．
2. 得られた解に限界がなく，目的関数の値がいくらでも大きくなる．すなわち発散する（非有界）．
3. 少なくとも一つの最適解が存在する．

1, 2 について例を挙げると．

まず場合1については制約条件が次式で与えられた場合である．図4.2にこれらの条件を満足する領域を示している．図より分かるようにこれらすべての条件を満足する領域は存在しない．したがって，この場合は解が存在しない．

$$\begin{aligned} \text{制約条件} \quad 3x_1 - 2x_2 &\leq 6 \\ x_1 + 2x_2 &\leq 10 \\ x_1 + x_2 &\geq 9 \\ x_1, x_2 &\geq 0 \end{aligned}$$

次に場合2については

$$\begin{aligned} \text{制約条件} \quad 2x_1 + x_2 &\geq 10 \\ -2x_1 + x_2 &\leq 3 \\ x_1 - 2x_2 &\leq 0 \\ x_1, x_2 &\geq 0 \end{aligned}$$

$$\text{目的関数} \quad z = 2x_1 + x_2 \text{ を最大にする}$$

を満足する x_1, x_2 を求める場合である．この場合は図4.3に示すように，条件を満足する領域は存在するが，目的関数はいくらでも大きくなり得て限界がなく，発散する．

[解の性質] ここで線形計画法の解の性質について調べてみる．

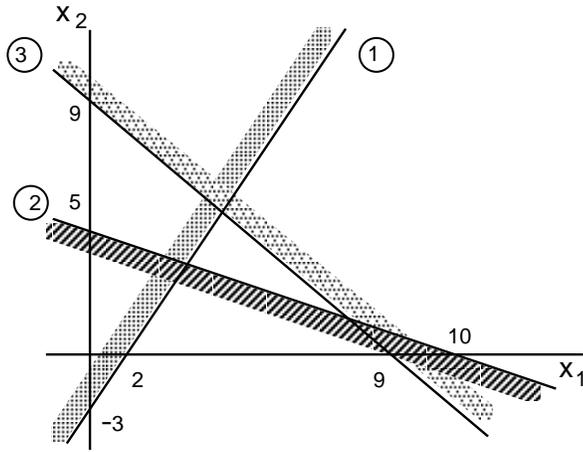


図 4.2: 解の存在しない場合

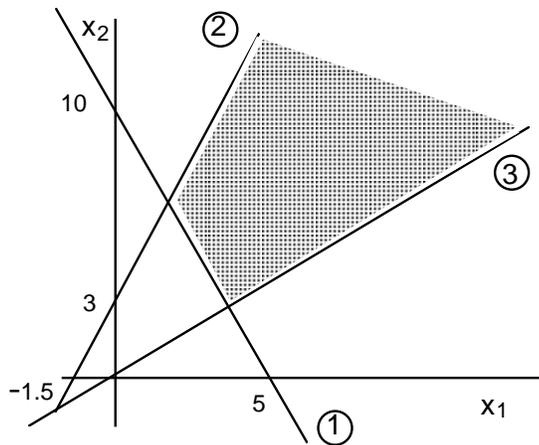


図 4.3: 解に限界がない場合

- 線形の制約条件式で与えられる解の存在可能な領域は凸集合である．ここで凸集合とはその集合内の任意の二つ点を結んだ線分上にあるすべての点はその集合に含まれるものをいう．
- 最適解が存在する場合は，それは凸集合の端点の一つである．ここで端点とはその集合の任意の二つの点の中央になることのない点をいう．

したがって線形計画法を解くには各端点について順次目的関数の値を求め，それを最大に

する端点を求めればよいことがわかる．

さて制約条件式の式 (4.3) に戻り，これを満足する x を求めることを考える． x が得られるためには少なくとも条件式の数 m が変数の数 n より少ないこと，すなわち $m < n$ が必要である．したがって x の n 個の変数中 $n - m$ 個については任意に与えることができ，それに応じて m 個の変数の値が定まる．まず容易に求まる端点として $n - m$ 個の変数は零に固定したものを選び，残りの m 個の独立変数の値を求める．このようにして得られた解を基底解という．また変数については，零に固定される変数を非基底変数，残りの変数を基底変数と呼ぶ．制約条件式を解いて得られた変数が非負なら制約条件すべてを満たしているので，これを基本許容解 (basic feasible solution) と呼び，さらに目的関数を最大にするものが最適解になる．

[シンプレックス法 (Simplex Method) による求解例] 上に述べた原理に基づくシンプレックス法の解き方を具体的な例で説明し，その後でこの方法の根拠について述べる．例として図的解法で述べた合金の製造計画の例を取り上げる．この問題を再録すると，次の条件の基で目的関数を最大にする x_1, x_2 を求めることである．

$$\text{制約条件式 } 0.4x_1 + 0.2x_2 \leq 8$$

$$0.4x_1 + 0.4x_2 \leq 10$$

$$0.2x_1 + 0.4x_2 \leq 8$$

$$x_1, x_2 \geq 0$$

$$\text{目的関数 } z = 4x_1 + 5x_2$$

1) 標準形式に変換

まず準備として標準形式に変換する．そのためにスラック変数 x_3, x_4, x_5 を導入して制約条件式を等式に変換すると次式を得る．

$$\text{制約条件式 } 0.4x_1 + 0.2x_2 + x_3 = 8$$

$$0.4x_1 + 0.4x_2 + x_4 = 10$$

$$0.2x_1 + 0.4x_2 + x_5 = 8$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

$$\text{目的関数 } z = 4x_1 + 5x_2 + 0 \times x_3 + 0 \times x_4 + 0 \times x_5$$

$$\text{または } -4x_1 - 5x_2 - 0 \times x_3 - 0 \times x_4 - 0 \times x_5 + z = 0$$

上記のように目的関数の式は変数の項をすべて左辺に，定数項を右辺にした式に変形して用いる．また目的関数の式にスラック変数の項を入れるため，その係数をすべて零にしている．

2) 初期テーブルの作成

表 4.2: テーブル-1

x_1	x_2	x_3	x_4	x_5	z	b	$b_i/a_{i,2}$
0.4	0.2	1	0	0	0	8 = x_3	8/0.2 = 40
0.4	0.4	0	1	0	0	10 = x_4	10/0.4 = 25
0.2	0.4	0	0	1	0	8 = x_5	8/0.4 = 20 ←
-4	-5	0	0	0	1	0 = z	

↑

制約条件式と目的関数の式の係数と定数からなるテーブルを作る．そのテーブルを表 4.2 に示す．表の作成方法は以下のとおり．各行がそれぞれの式に対応する．第 1 行が制約条件式の第 1 式．以下，第 2 式，第 3 式，目的関数の式となる．目的関数の式の前には他と区別するために横線を引いて置く．各変数の列にはそれぞれの係数を記入する．定数は b の列に記入する．また制約条件式には変数 z の項はないが，計算上必要なので係数を零の項として入れて置く．

3) 最初の基本許容解を求める (サイクル 1)

制約条件式は 3 式であるので，変数 5 個中独立変数は 3 個となり，残り 2 個は任意に選べる．したがって，まず x_1, x_2 を零と置くと

$$x_1 = 0, x_2 = 0, x_3 = 8, x_4 = 10, x_5 = 8$$

となる．その理由を述べる．表から分かるように， x_1, x_2 を零と置くと，各式においてこの 2 つの変数の列は考えなくてよくなる．残りの変数 x_3, x_4, x_5, z においては，各行をみると係数 1 のものが各々 1 個所であるので，その位置の変数の値が定数項の値となる．参考のために，表の定数項の右側の欄に該当する変数の記号を記入してある．

これら x の値は制約条件のすべてを満足しているのでこれが基本許容解の一つである．この時，目的関数の値は $z = 0$ である．

この基本許容解は目的の最適解であろうか．その判断は第 4 行の目的関数の行の係数の符号により判断する．係数が負のものがあれば最適解ではない．理由については後で述べる．表より分かるように 2 個所に負のものがあるので，この解は最適解ではない．

4) つぎの基本許容解を求める (サイクル 2)

4 - 1) 入れ換える基底変数と非基底変数の決定：他の基本許容解を求めるには，基底変数の一つと非基底変数の一つを選んで，入れ換えて他の端点を選んで上記と同じ方法で調べる．そのためにはそれぞれの変数を選ばねばならない．その方法について述べる．

まず非基底変数 x_1, x_2 のなかから基底変数に変えるものを選ぶ．それには目的関数の係

数の中で負で絶対値が最大なものを持つ変数を選べばよい(理由は後述)。この場合 -5 の係数を持つ x_2 を基底変数に変える。この x_2 の列をピボット列と呼ぶ。

つぎに基底変数 x_3, x_4, x_5 の中から非基底変数に変えるものを選ぶ。このためには上記で求めたピボット列の各係数で各右辺の値を割り、その値が最小になる行にある変数(係数が1になっている変数)にすればよい(理由は後述)。この行をピボット行と呼ぶ。今の場合

$$\text{第1行: } 8/0.2=40, \text{ 第2行: } 10/0.4=25, \text{ 第3行: } 8/0.4=20$$

より第3行が最も小さいので、変数 x_5 を非基底変数とする。表の最右欄にこれらの値を記入している。

4 - 2) 新しいテーブルを作る(ピボット操作): 以上により入れ換えるべき基底変数と非基底変数が得られたのでそれをもとに新しいテーブルを作る。このためにはつぎに述べるピボット操作を行なう。

ピボット操作とはピボット列の係数をつぎのとおり変換する操作である。

1. ピボット(ピボット行とピボット列の交点)の係数を1にする。そのためにはこの行のすべての数値(定数項を含み)をピボットの値で割る。
2. ピボット行以外の行(目的関数の行も含む)ではピボット列の係数を零にする。そのためには他の行の数値に適当な値を乗じて加算すればよい。

いまの場合、ピボット列は第2列、ピボット行は第3行であるので、つぎの操作を行えばよい。

- 第3行を0.4で割って新しい第3行とする。
- 新第3行に -0.2 を乗じて第1行に加える。
- 新第3行に -0.4 を乗じて第2行に加える。
- 新第3行に5を乗じて第4行に加える。

このような変換を行うと表4.3が得られる。

4 - 3) 基本許容解: 今度は非基底変数の x_1, x_5 を零に置くと、先に述べたように基本許容解として $x_1 = 0, x_2 = 20, x_3 = 4, x_4 = 2, x_5 = 0$ が得られ、目的関数値は $z = 100$ となる。表4.3を見ると目的関数の行(第4行)に負の係数が存在するので最適解ではないので、つぎの基本許容解を求める。

5) 第3の基本許容解を求める(サイクル3)

今回はサイクル2で述べたと同様の方法を繰り返すことになる。したがって、非基底変数の内 x_1 を基底変数に移す。 x_1 の係数で各式の右辺を割ると

$$\text{第1行: } 4/0.3=13.3, \text{ 第2行: } 2/0.2=10, \text{ 第3行: } 20/0.5=40$$

表 4.3: テーブル-2

x_1	x_2	x_3	x_4	x_5	z	b	$b_i/a_{i,1}$
0.3	0	1	0	-0.5	0	4 = x_3	$4/0.3 = 13.2$
0.2	0	0	1	-1.0	0	2 = x_4	$2/0.2 = 10 \leftarrow$
0.5	1	0	0	2.5	0	20 = x_2	$20/0.5 = 40$
-1.5	0	0	0	12.5	1	100 = z	

↑

表 4.4: テーブル-3

x_1	x_2	x_3	x_4	x_5	z	b
0	0	1	-1.5	1.0	0	1 = x_3
1	0	0	5.0	-5.0	0	10 = x_1
0	1	0	-2.5	2.75	0	15 = x_2
0	0	0	7.5	5.0	1	115 = z

したがって、第2行にある基底変数 x_4 を非基底変数とする。これでピボット行とピボット列が決まったのでつぎのピボット操作を行なう。

- 第2行を0.2で割って新しい第2行とする。
- 新第2行に-0.3を乗じて第1行に加える。
- 新第2行に-0.5を乗じて第3行に加える。
- 新第2行に1.5を乗じて第4行に加える。

以上を行って表4.4を得る。非基底変数の x_4, x_5 を零と置くと基本許容解は

$$x_1 = 10, x_2 = 15, x_3 = 1, x_4 = 0, x_5 = 0$$

となり、 $z = 115$ となる。今回は表を見ると第4行の係数に負のものが存在しないので、ここで求めた基本許容解が最適解であることが分かる。従って

最適解としてつぎのものが得られた。

$$x_1 = 10, x_2 = 15, z = 115$$

ここでは説明の都合上、各サイクルのテーブルは別々に作成したが、実際は後述の2段階法の表のように一連の表とする。シンプレックス法はこのように表を中心に計算が行なわれるので別名タブロー法ともいわれている。

[シンプレックス法の手順]以上に述べたようにシンプレックス法では次の手順で計算が行われる(図4.4を参照)。

ステップ1 初期テーブルを作る。

ステップ2 最適性のテストを行う。最適ならここで停止。最適でない場合はつぎのステップに進む。

最適解の条件:目的関数のすべての係数が非負となった時、非基底変数を零とし求めた基底変数の値および目的関数の値が求める最適解である。

ステップ3 ピボット列を決定する(非基底変数のうち基底変数に変えるもの1つを決定する)。

ステップ4 ピボット行を決定する(基底変数のうち非基底変数に変えるもの1つを決定する)。

ステップ5 ピボット列についてピボット行のみ1とし、同列の他の行はすべて0にするよう行操作(ピボット操作)を行いステップ2に戻る。

[シンプレックス法の原理]ピボット行とピボット列を決定する方法についてその理由を説明する[32]。今、ある時点のテーブルが表4.5のとおりであったとする。表で y_{11} , c_1 など はそれぞれの係数を表す。非基底変数から基底変数に移す変数 x_s (ピボット列が s 列)、基底変数から非基底変数に移す変数が x_{pr} (ピボット行が r 行)とする。これにステップ5の演算(ピボット操作)を行った結果が表4.6のとおりとする。表から明かなようにピボット列でピボットの係数が1で他はすべて零になっている。これらの表からつぎのことが分かる。

(1) 表4.5は1つの基本許容解を与え、基底変数は非負でなければならないので

$$b_i \geq 0, i = 1, \dots, m$$

でなければならない。何故ならこれらの値は基底変数の値になるので。したがって、もし

$$b_j < 0, j \in [1, \dots, m]$$

が存在すれば解は存在しないことになる。しかし制約条件によっては解が存在するにも関わらずこの状態になることがあるので、その場合は次節で述べる2段階法を試みる必要がある。

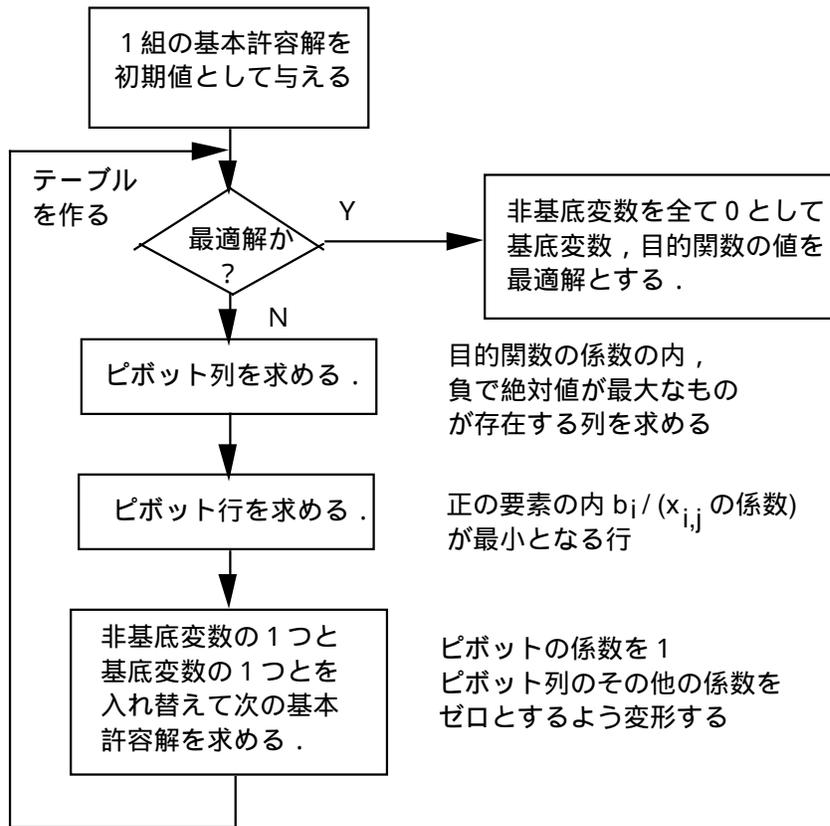


図 4.4: シンプレックス法の系統図

(2) 新しいピボットの条件: x_{pr} は基底変数であるから $x_{pr} \geq 0$, 従って x_{pr} の値である b_r について $b_r \geq 0$ である. 新しいテーブルに解があるためには新しい基底変数 x_s について $x_s \geq 0$ でなければならない. $x_s = b_r / y_{rs}$ であるから $y_{rs} > 0$ でなければならない

(3) ピボット行の決定: 同様に新しいテーブルの残りの変数は

$$x_{qi} = b_i - y_{is} b_r / y_{rs}, i = 1, \dots, m; i \neq s \quad (4.7)$$

新テーブルに解が存在するためには前項と同じく $x_{qi} \geq 0, i = 1, \dots, m, i \neq s$ でなければならない.

まず $y_{is} \leq 0$ の時は, $b_i \geq 0, b_r \geq 0, y_{rs} > 0$ となり, $x_{qi} \geq 0$ となるので解の存在条件は満足する. 全ての $y_{is} \leq 0, i = 1, \dots, m$ の時は解は限界がない(発散する).

つぎに $y_{is} > 0$ の時は $x_{qi} \geq 0$ であるためには

$$b_i - y_{is} b_r / y_{rs} \geq 0$$

でなければならないので

$$\frac{b_r}{y_{rs}} \leq \frac{b_i}{y_{is}} \quad (4.8)$$

となる．すなわち $y_{is} > 0$ のすべての i について上記の条件を満足せねばならない．

以上よりピボット列において，正の要素 y_{js} に対して右辺とこの要素との比 b_j/y_{js} が最小となる行をピボット行とせねばならない．

(4) ピボット列の決定:新しいテーブルでの目的関数の値を w_* とすると

$$w_* = w - c_s b_r / y_{rs} \quad (4.9)$$

となる．したがって目的関数の増加は

$$w_* - w = -c_s b_r / y_{rs}$$

となるので，目的関数の増加をできるだけ大きくするには

$$-c_s b_r / y_{rs}$$

が正で，かつ極力大きい r と s を選ばばよい．しかし実際これを行うには多数の計算を必要とするので，その代わりに c_s が負でかつ絶対値が極力大になるものを選ぶ方法を用いる．これにより

$$w_* - w \geq 0$$

となり目的関数の増加が保証される．

以上よりピボット列としては各係数のうち，負でかつ絶対値が最大の列を選ばばよい．

(5) 最適解の条件: 前項より目的関数の行の係数がすべて正になると目的関数の値はそれ以上には増加しないことが分かる．したがって，この段階で最適解に達したことが分かる．

2 段階法

前節で述べたシンプレックス法では最初に一つの基本許容解を求め，それに対してピボット操作を行って最適解を求めているが，最適解があるにも関わらずそこで述べた方法では基本許容解が得られない場合がある．これは最初的基本許容解が得られない場合であり，基底変数になるべき変数が負となり制約条件外になる時に起こる．このような場合に対する解法として考え出されたのが2段階法である．この方法では次の2段階で解を求める．

1. 一つの基本許容解を求める．それにシンプレックス法に類似した方法を用いる．

表 4.5: テーブル-1

	x_1	x_2	\cdots	x_s	\cdots	x_n	z	b	
	y_{11}	y_{12}	\cdots	y_{1s}	\cdots	y_{1n}	0	b_1	$= x_{p1}$
	y_{21}	y_{22}	\cdots	y_{2s}	\cdots	y_{2n}	0	b_2	$= x_{p2}$
	\vdots	\vdots		\vdots		\vdots	\vdots	\vdots	\vdots
$x_{pr} \leftarrow$	y_{r1}	y_{r2}	\cdots	y_{rs}	\cdots	y_{rn}	0	b_r	$= x_{pr}$
	\vdots	\vdots		\vdots		\vdots	\vdots	\vdots	\vdots
	y_{m1}	y_{m2}	\cdots	y_{ms}	\cdots	y_{mn}	0	b_m	$= x_{pm}$
	c_1	c_2	\cdots	c_s	\cdots	c_n	1	w	$= z$
				\uparrow					
				x_s					

表 4.6: テーブル-2

x_1	x_2	\cdots	x_s	\cdots	x_n	z	b	
y_{11}^*	y_{12}^*	\cdots	0	\cdots	y_{1n}^*	0	$b_1 - y_{1s}b_r/y_{rs}$	$= x_{q1}$
y_{21}^*	y_{22}^*	\cdots	0	\cdots	y_{2n}^*	0	$b_2 - y_{2s}b_r/y_{rs}$	$= x_{q2}$
\vdots	\vdots		\vdots		\vdots	\vdots	\vdots	\vdots
y_{r1}^*	y_{r2}^*	\cdots	1	\cdots	y_{rn}^*	0	b_r/y_{rs}	$= x_s$
\vdots	\vdots		\vdots		\vdots	\vdots	\vdots	\vdots
y_{m1}^*	y_{m2}^*	\cdots	0	\cdots	y_{mn}^*	0	$b_m - y_{ms}b_r/y_{rs}$	$= x_{qm}$
c_1^*	c_2^*	\cdots	0	\cdots	c_n^*	1	$w - c_s b_r/y_{rs}$	$= z$

2. 上記で求めた基本許容解を使用して最適解を求める．これはシンプレックス法そのものである．

実際はこれら二つを組み合わせた方法となっている．以下，具体的にその方法について説明する．

先ず標準形式で右辺に負の定数があればその式の両辺に -1 を乗じて，すべての制約条件式の右辺の定数項が正になるようにする．

新しく非負の変数（人為変数という） x_{n+1}, \dots, x_{n+m} を導入すると次の式が得られる．

$$\text{制約条件式 } a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + x_{n+1} = b_1 \quad (4.10)$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + x_{n+2} = b_2$$

$$\begin{aligned}
 & \vdots \\
 & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + x_{n+m} = b_m \\
 & x_i \geq 0, i = 1, \cdots, n+m \\
 \text{目的関数} \quad & z = c_1x_1 + c_2x_2 + \cdots + c_{n+m}x_{n+m}
 \end{aligned} \tag{4.11}$$

ただし目的関数の式で人為変数に関する係数 c_{n+i} , $i = 1, \cdots, m$ は零とする.

ここで人為変数を基底変数に選べば最初の基本許容解は

$$x_1 = \cdots = x_n = 0, x_{n+1} = b_1, \cdots, x_{n+m} = b_m$$

となる. しかし実際はこれら人為変数は零でなければならないので, シンプレックス同様の方法によりすべての人為変数を零にする解が求めれば, それを第2段階のシンプレックス法の初期値に用いることができる. したがって, ここで次の変数 w を考えてみる.

$$w = x_{n+1} + x_{n+2} + \cdots + x_{n+m} \tag{4.12}$$

人為変数は非負であり, かつ上記状態では零になることから, 上記解を求めることは, w を最小にしかつそれが零とする解を求めることに相当する. すなわち, 目的関数 $-w$ を最大とする解をシンプレックス法で求めればよいことが分かる.

ここで前記制約条件式より

$$\begin{aligned}
 x_{n+i} &= b_i - a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n \\
 & i = 1, \cdots, m
 \end{aligned} \tag{4.13}$$

となるので, これを w の式に代入してつぎの式が得られる.

$$-w = -d_1x_1 - d_2x_2 - \cdots - d_nx_n + w_0 \tag{4.14}$$

または

$$d_1x_1 + d_2x_2 + \cdots + d_nx_n - w = w_0 \tag{4.15}$$

ここで,

$$d_i = -(a_{i1} + a_{i2} + \cdots + a_{in}), i = 1, \cdots, n \tag{4.16}$$

$$w_0 = -(b_1 + b_2 + \cdots + b_m) \tag{4.17}$$

目的関数としてはここで求めた $-w$ を使用する.

なお, w を零にする解が得られない場合はこの線形計画問題は解が存在しないことになる.

以下では具体的な例でその方法を説明する．ここでは次の問題を解くことを例として取り上げる．

$$\begin{aligned} \text{制約条件} \quad & x_1 + x_2 \geq 4 \\ & x_1 + x_2 \leq 10 \\ & x_1 - x_2 \geq -2 \\ & x_1 - x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

$$\text{目的関数} \quad z = x_1 + 2x_2 \quad \text{を最大にする}$$

先ず通常のスラック変数を導入して標準形式に変換すると

$$\begin{aligned} \text{制約条件} \quad & x_1 + x_2 - x_3 = 4 \\ & x_1 + x_2 + x_4 = 10 \\ & -x_1 + x_2 + x_5 = 2 \\ & x_1 - x_2 + x_6 = 2 \\ & x_1, x_2, \dots, x_6 \geq 0 \end{aligned}$$

$$\text{目的関数} \quad z = x_1 + 2x_2 + 0 \times x_3 + \dots + 0 \times x_6$$

通常のシンプレックス法を適用すると最初の基本許容解は x_1, x_2 を非基底変数にし x_3, \dots, x_6 を基底変数にすることになるが，基底変数のひとつが $x_3 = -4$ と負になり制約条件を満足しない．この問題の制約範囲を示す図 4.5 によると明らかのように $x_1 = 0, x_2 = 0$ はこの制約条件の中には入っていない．したがってこの点は基本許容解にはなり得ない．したがって，シンプレックス法はそのままでは適用できないので 2 段階法を適用する．

まず 2 段階法のための人為変数 x_7, \dots, x_{10} を導入すると次式が得られる．

$$\begin{aligned} \text{制約条件} \quad & x_1 + x_2 - x_3 + x_7 = 4 \\ & x_1 + x_2 + x_4 + x_8 = 10 \\ & -x_1 + x_2 + x_5 + x_9 = 2 \\ & x_1 - x_2 + x_6 + x_{10} = 2 \\ & x_1, x_2, \dots, x_{10} \geq 0 \end{aligned}$$

$$\text{目的関数} \quad -w = 2x_1 + 2x_2 - x_3 + x_4 + x_5 + x_6 + 0 \times x_7 + \dots + 0 \times x_{10} - 18$$

の最大値が零のもとで

$$z = x_1 + 2x_2 + 0 \times x_3 + \dots + 0 \times x_{10} \text{を最大にする}$$

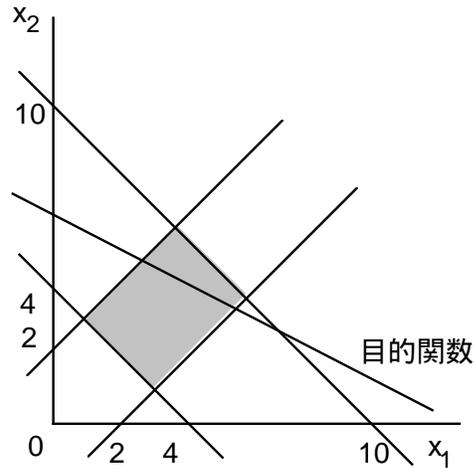


図 4.5: 制約条件の範囲 (2段階法)

2段階法による計算経過は表 4.7, 表 4.8 に示す. シプレックス法に比べて $-w$ の行が増えているほかは変わるところはない. x_7, \dots, x_{10} の列を陽には示していないのは計算上特に必要がないからである. 基底の列に基底変数として示しており, 後はピボット操作のつど入れ替えれば追跡できる.

サイクル 1 の初期テーブルにおける w の行について補足すると, x_1, \dots, x_6 の各係数は式 (4.15) より分かるように, 上 4 行の各係数の合計値に $(-)$ を乗じたものであり, 定数項も同じく上 4 行の合計値に $(-)$ を乗じたものとなっている.

第 1 段階では $-w$ を目的関数とし, かつ最初の基底変数に x_7, \dots, x_{10} を選んでシプレックスと同様の方法を適用する. $-w$ の行が目的関数の行として追加されたこと以外は変わるところはない. この $-w$ 行の初期テーブルの値には, 上記のように式 (4.15) を利用して求めた値を代入し, 以後はピボット操作で変えてゆく. 同様に z の行もピボット操作により変えてゆく. 表より分かるように 5 サイクルで $w = 0$ となり, 第 1 段階の最適解すなわち基本許容解が得られた.

第 2 段階に入り今度は目的関数に z を選んでシプレックス法を適用し 6 サイクル目で最適解

$$x_1 = 4, x_2 = 6, z = 16$$

が得られた.

表 4.7: 2 段階法の例 (1 / 2)

サイクル	基底	x_1	x_2	x_3	x_4	x_5	x_6	z	b	b/a
1	x_7	1	1	-1	0	0	0	0	4	4
	x_8	1	1	0	1	0	0	0	10	10
	x_9	-1	1	0	0	1	0	0	2	
	x_{10}	1	-1	0	0	0	1	0	2	$2 \Leftarrow$
	z	-1	-2	0	0	0	0	1	0	
	$-w$	-2	-2	1	-1	-1	-1	0	-18	
		↑								
2	x_7	0	2	-1	0	0	-1	0	2	$1 \Leftarrow$
	x_8	0	2	0	1	0	-1	0	8	4
	x_9	0	0	0	0	1	1	0	4	
	x_1	1	-1	0	0	0	1	0	2	
	z	0	-3	0	0	0	1	1	2	
	$-w$	0	-4	1	-1	-1	1	0	-14	
		↑								
3	x_2	0	1	-0.5	0	0	-0.5	0	1	
	x_8	0	0	1	1	0	0	0	6	$6 \Leftarrow$
	x_9	0	0	0	0	1	1	0	4	
	x_1	1	0	-0.5	0	0	0.5	0	3	
	z	0	0	-1.5	0	0	-0.5	1	5	
	$-w$	0	0	-1	-1	-1	-1	0	-10	
			↑							
4	x_2	0	1	0	0.5	0	-0.5	0	4	
	x_3	0	0	1	1	0	0	0	6	
	x_9	0	0	0	0	1	1	0	4	$4 \Leftarrow$
	x_1	1	0	0	0.5	0	0.5	0	6	
	z	0	0	0	1.5	0	-0.5	1	14	
	$-w$	0	0	0	0	-1	-1	0	-4	
				↑						
5	x_2	0	1	0	0.5	0	-0.5	0	4	
	x_3	0	0	1	1	0	0	0	6	
	x_5	0	0	0	0	1	1	0	4	$4 \Leftarrow$
	x_1	1	0	0	0.5	0	0.5	0	6	12
	z	0	0	0	1.5	0	-0.5	1	14	
	$-w$	0	0	0	0	0	0	0	0	
					↑					

表 4.8: 2 段階法の例 (2 / 2)

サイクル	基底	x_1	x_2	x_3	x_4	x_5	x_6	z	b	b/a
6	x_2	0	1	0	0.5	0.5	0	0	6	
	x_3	0	0	1	1	0	0	0	6	
	x_6	0	0	0	0	1	1	0	4	
	x_1	1	0	0	0.5	-0.5	0	0	4	
	z	0	0	0	1.5	0.5	0	1	16	

双対問題

線形計画問題では双対問題と言われる互いに裏と表の関係になる問題があるので、それについて述べる。まず、行列で表わしたつぎの二つの線形計画問題を考える。

[線形計画問題 1]

$$\begin{aligned} \text{制約条件 } Ax &\leq b \\ x &\geq 0 \end{aligned} \quad (4.18)$$

のもとで

$$\text{目的関数 } z = c^T x \text{ を最大にする} \quad (4.19)$$

[線形計画問題 2]

$$\begin{aligned} \text{制約条件 } A^T y &\geq c \\ y &\geq 0 \end{aligned} \quad (4.20)$$

のもとで

$$\text{目的関数 } w = b^T y \text{ を最小にする} \quad (4.21)$$

ここで $x \in R^n, y \in R^m, b \in R^m, c \in R^n, A \in R^{m \times n}, (n \geq m)$ である。

この二つの問題の式を見ると、制約条件を表わす式では係数行列は互いに転置の関係にあり、その定数項と目的関数の係数とが互いに入れ替わっており、不等号の向きが逆であり、かつ一方は最大問題、他方は最小問題であるというように異なっているが、同じ最適解が得られることが分かっている。このような後者の問題を前者（主問題という）の双対（そうつい）問題という。主問題で最適解が得られれば双対問題でも最適解が得られ、それは主問題の解と同じになる。

ここで簡単に同一解が得られることを証明する。式 (4.18) の両辺に左から y^T を乗ずる。 $y \geq 0$ であることを考えると、不等号の向きは変わらないので、

$$\mathbf{y}^T \mathbf{A} \mathbf{x} \leq \mathbf{y}^T \mathbf{b}$$

同様に式(4.20)の転置に右から \mathbf{x} を乗ずると, $\mathbf{x} \geq 0$ であるので,

$$\mathbf{y}^T \mathbf{A} \mathbf{x} \geq \mathbf{c}^T \mathbf{x}$$

この両式よりつぎの関係が得られる.

$$\mathbf{y}^T \mathbf{b} \geq \mathbf{c}^T \mathbf{x}$$

したがって, $\mathbf{c}^T \mathbf{x}$ の最大値を求めることは $\mathbf{y}^T \mathbf{b}$ の最小値を求めることになる. 実際は最適状態では両者が一致することの証明が必要ではあるが, ここでは省略する. 詳細は文献 [18],[28]などを参照のこと.

本章の始めで取り上げた合金の製造計画問題の双対問題はつぎのようになる.

$$\text{制約条件式 } 0.4y_1 + 0.4y_2 + 0.2y_3 \geq 4$$

$$0.2y_1 + 0.4y_2 + 0.4y_3 \geq 5$$

$$y_1 \geq 0, y_2 \geq 0; y_3 \geq 0$$

$$\text{目的関数 } w = 8y_1 + 10y_2 + 8y_3 \text{ を最小にする}$$

この最適解は $y_1 = 0, y_2 = 5, y_3 = 7.5, w = 115$ となり, 最適値 115 が主問題のものと同じことが分かる.

演習問題

4.1 つぎの制約条件式及び目的関数が与えられている線形計画問題について以下に答えよ.

- 1) 図的解法で答を求めよ.
- 2) シンプレックス法で解け.

$$\text{制約条件式 } 4x_1 + x_2 \leq 20$$

$$x_1 + 3x_2 \leq 15$$

$$x_1 + x_2 \leq 7$$

$$x_1, x_2 \geq 0$$

$$\text{目的関数 } z = 7x_1 + 12x_2 \text{ を最大にする.}$$

動的計画法

[概要] 動的計画法は最適化手法のひとつで、時間の経過に応じて決定を行なうものに適用される方法であり、再帰的な手法であるのでコンピュータによる演算にも適している。また本法は定型的な手法ではなく考え方であり、多くの事例を通じて修得する必要があるので、各種の事例を挙げて説明している。

はじめに

動的計画法 (DP: Dynamic Programming) は動的システムの最適化手法として研究され、確立された方法である。ある一時点の決定ではなく、時間の経過に応じてつぎつぎと決定を行い、この一連の決定の結果として得られた全期間の効果を最大にする方策を決定する問題を扱う。例えば、ある地点より別の地点へ行く経路が多数ある場合に、最短時間で行く経路を発見する問題などがこれに相当する。このような多段階の決定過程の解析法としてベルマン (Richard Bellman) により 1950 年代に開発された。

多段階決定問題を全体の組み合わせ問題として解けば多数の組み合わせのため、一般に多量の計算を必要とする。しかしこれを次元の低い問題に分解し、最適性の原理を用いて、その一連の段階の組合せとして解けば計算量も少なくなる。これが動的計画法の大きな特徴といえる。

動的計画法は各種の問題に適用できる応用分野の広い手法である。

動的計画法の一般解

最適性の原理は

最適方策では、最初の状態および決定がどのようなものであっても、残りの決定は先行の決定から生じた状態に関して最適方策になっていなければならない

と規定している。以下数式によりこの原理を表す。

図 5.1 に示す N 段階の多段階決定過程を考える。第 1 段階より最適な決定を下しつつ、 k 段階まで決定をしたとして、つぎの $k + 1$ 段階の決定をいかに下せばよいかを考える。 $k + 1$ 段階では

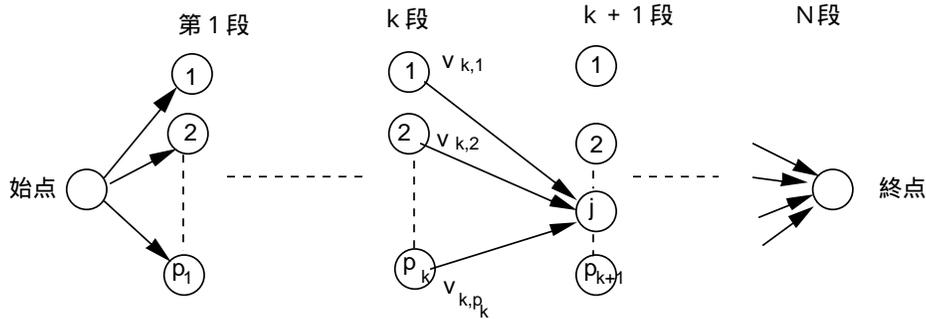


図 5.1: 多段決定過程

p_{k+1} 個の状態があるとして、各状態への最適方策を求める。この問題は、まず $k+1$ 段の 1 つの状態をとり、この状態へは k 段の p_k 個の状態のうちどの状態より移行するのが最適であるかを求め、これを $k+1$ 段の全状態 p_{k+1} 個について行うことに等しい。

さて、最適方策を決めるには目的関数を用意し、この値を最大(または最小)にする方法を適用する。 $k+1$ 段の j 番目の状態を表す変数として $v_{k+1,j}$ を用いて、この状態での目的関数を $f(v_{k+1,j})$ と表す。また k 段の i 番目の状態より $k+1$ 段の j 番目の状態に移る際の目的関数の増加を関数 $g(v_{k,i}, v_{k+1,j})$ で表す。これは最短経路問題での 2 点間の通行時間に相当する。以上の式を用いて、 $k+1$ 段での最適方策の決定方針はつぎの式により表せる(目的関数を最大にする場合)。

$$f(v_{k+1,j}) = \max_{i \in P_k} [f(v_{k,i}) + g(v_{k,i}, v_{k+1,j})] \quad (5.1)$$

$$j = 1, \dots, p_{k+1}$$

この式はつぎのことを意味している。すなわち $k+1$ 段の j 番目の状態には k 段のどの状態から移行するのが最適かは k 段の i 番目の状態の目的関数の値と、そこから $k+1$ 段の j 番目の状態に移る際の目的関数の増加との和を最大にする i をつけることにより求められる。なお上式では目的関数を最大(max)にすることを求めているが、問題によっては最小(min)を求めることもある。たとえば利益最大問題では max, 最短経路問題では min を用いる。

動的計画法では上式を用いて、つぎの手順で行う。

1. $k=1$ より開始し、 $k=N$ までこの式により各段の決定を行う。この場合、決定全体に対する制約条件がある場合には、この時点で状態値を具体的に決定することはできないので、変数としておく。

2. 最終段階 ($k = N$) に達したら、全体の制約条件 (たとえば使用可能な資源の制限など) がある場合には、この制約条件に照らしつつ、今度は始点に向かって後向きに進めて各段の方策を定め、変数としていた状態値を具体的に決定する。

このように動的計画法では始点より開始して終点に達したら、再び始点に向かって後戻りしつつ最終の値を決めて行く。また問題によっては上記とは逆に終点より開始して後向きに戻り、始点に達したら今度は終点に向かって前向きに進みながら決定する方法を採用している。この際の最適方針を決める式はつぎのとおりになる。

$$f(v_{k,j}) = \max_{i \in P_{k+1}} [f(v_{k+1,i}) + g(v_{k,j}, v_{k+1,i})] \quad (5.2)$$

$$j = 1, \dots, p_k$$

以上に述べた手順を図 5.2 に系統図として示している。

最適方針の決定のための演算法としてはつぎの各種の解決法がある。

1. 解析的な方法
2. 図式的解法
3. 数値計算による解法

動的計画法は目的関数のモデル化により各種の問題に応用できる。その例として最短経路問題、資源配分問題、倉庫運営問題および資金配分問題について考える。

最短経路決定問題 [応用例 - 1]

出発点と目的地間の道路と途中の各二点間の所要時間が図 5.3 のように与えられているとき、最短時間で行くにはどのような経路を取ればよいかを考える。最初の解法は目的地より逆向きにたどる方法である。目的地 G より逆向きに所要時間を加算してゆく。図 5.4 はこのようにして計算した最短時間を 図 5.3 の中に示している。例えば、 B_2 点に行くには A_1 点からでは 8 分、 A_2 点からでは 19 分かかるので、短い方を採用して 8 分としている。このようにして出発地 S に達すると、今度は逆向きに戻りながら、所要時間の短い方を採用して経路を決定すると、図で太線で示す経路 $S \rightarrow C_1 \rightarrow B_2 \rightarrow A_1 \rightarrow G$ が求まり、その所要時間は S での値の 18 分となる。なお各点の時間はその地点から目的地 G までの最短所要時間を示している。

第 2 の解法は図 5.5 に示すように、出発地 S より開始して各地点までの最短所要時間を計算し、終点に達したら逆に元に戻りながら最短経路を捜す方法もある。この場合は各地点の時間はその点に到達するのに必要な最短時間を示している。上記のいずれの方法がよいかは問題により決り、上記の問題では両者の差はないが、例えば目的地に到達すべき時刻

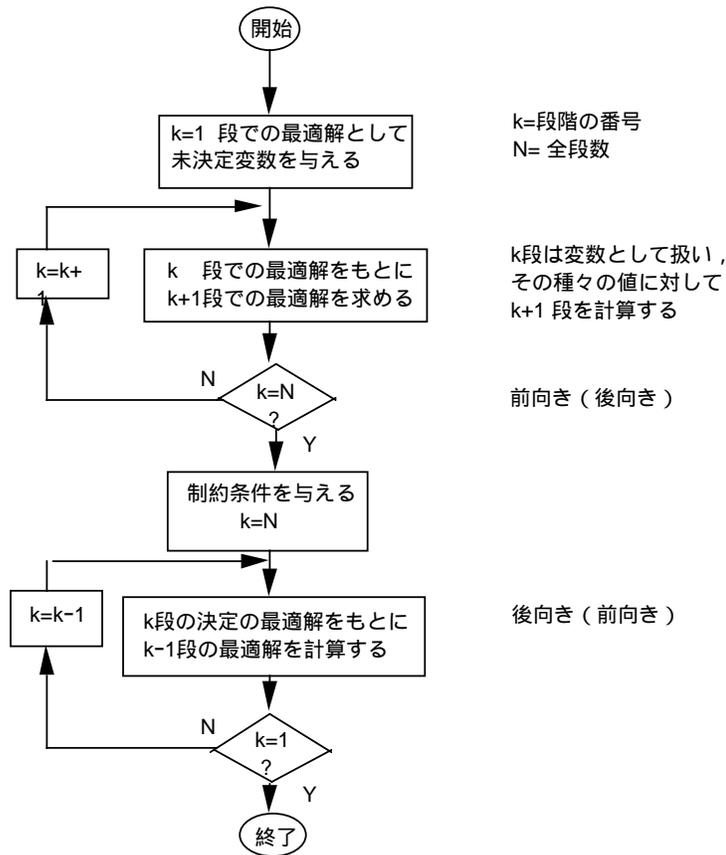


図 5.2: 動的計画法の系統図

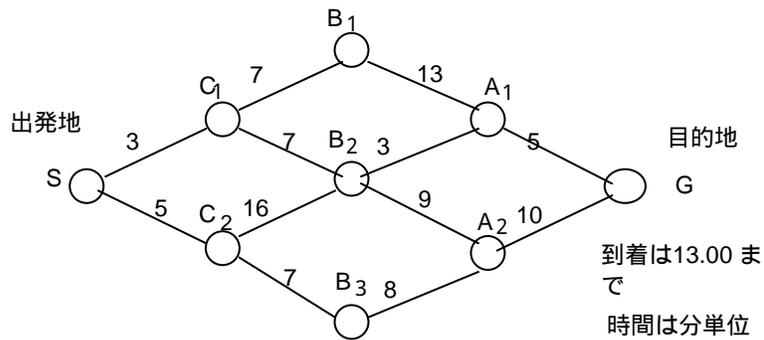


図 5.3: 経路図

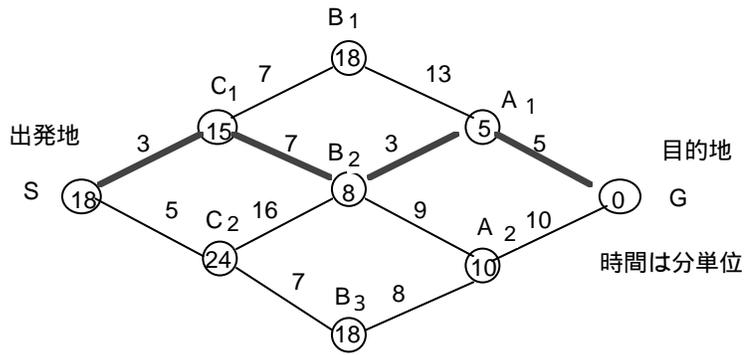


図 5.4: 目的地への最短時間

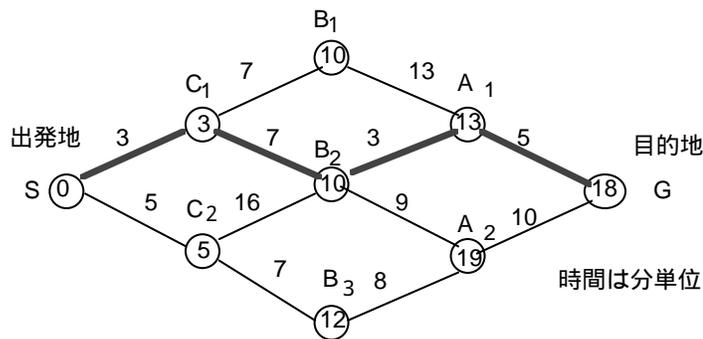


図 5.5: 出発地よりの最短時間

が指定され、それに間に合うようにするための最短経路と出発時刻を求める場合には第一の方法がよい。

資源配分問題 [応用例 - 2]

静的な多元問題に多段決定方を適用して動的計画法で解くことができる。その例として資源配分問題を考えてみる。これは一定の資源をどの分野にどのように配分すれば最大の効果(利益)を期待できるかを決定する問題である。その一例としてつぎの問題を考えてみる。

[問題] 農地が 3 箇所あり、それぞれ別の作物を作っている。農地の状況と作物の種類から、それぞれに配置した農業従事者の人数と各農地の収益(単位千万円)との間に表 5.1

表 5.1: 農業従事者の人数と収益の関係

人数	農地の収益		
	1	2	3
0	0	0	0
1	1.5	1.0	0.8
2	2.3	1.9	1.6
3	3.0	3.3	3.6
4	3.6	4.2	4.0
5	4.0	5.0	4.8

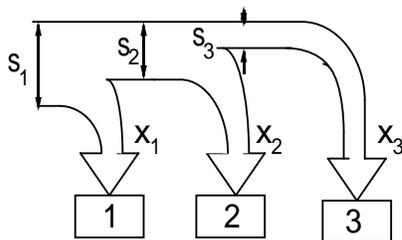


図 5.6: 人数の配分

に示す関係があるとする．農業従事者の総数が 5 名とした時に各農地にどの様に配置すれば最大の収益を上げることができるか．

[解答] 農業従事者の各農地への配置を $1 \Rightarrow 2 \Rightarrow 3$ の順に配置したとして 3 ステップの動的計画法を適用することができる．つぎのとおり記号を定める (図 5.6 参照) ．

- 各農地への配置人数 : $x_j, j = 1, 2, 3, j =$ 農地を表す
- 配置可能な人数 : $s_j, j = 1, 2, 3$
- 各農地の収益 : $g_j(x_j), j = 1, 2, 3$
- 人数 s_i による収益 : $f_j(s_j), j = 1, 2, 3$

これにより本問題は

$$\sum_{j=1}^3 x_j = 5 \text{ の制約のもとで}$$

$$\sum_{j=1}^3 g_j(x_j) \text{ を最大にする}$$

表 5.2: 農地への人員配置：第 1 ステップ

s_3	$f_3(s_3)$	x_3^*
0	0	0
1	0.8	1
2	1.6	2
3	3.6	3
4	4.0	4
5	4.8	5

問題となる。

従って、さきに挙げた動的計画法の式 (5.2) より次のように表すことができる。

$$f_j(s_j) = \max_{x_j \leq s_j} [f_{j+1}(s_j - x_j) + g_j(x_j)] \quad (5.3)$$

$$j = 1, 2, 3$$

ここで終点から始点に向かって値を決めて行く。

[第 1 ステップ] 農地 3 への割当：上式において $j = 3$ に相当する。この段階以後の配置はないので、残っている従事者はすべてこの農地に配置してよい。従って $x_3 = s_3$ かつ $f_4(0) = 0$ となり最適方策の式はつぎのとおり。

$$f_3(s_3) = g_3(s_3) \quad (5.4)$$

ここで $s_3 = 0, \dots, 5$ の 6 つの場合があるので、それぞれに対する $f_3(s_3)$ の値、 x_3 の最適解 x_3^* は表 5.2 に示すとおりになる。

[第 2 ステップ] 農地 2 への割当： $j = 2$ に相当するので、このステップの最適方策の式はつぎのとおり。

$$f_2(s_2) = \max_{x_2 \leq s_2} [f_3(s_2 - x_2) + g_2(x_2)] \quad (5.5)$$

ここでも $s_2 = 0, \dots, 5$ の場合があるのでそれぞれに対する、 x_2 を変えた場合の $f_3(s_2 - x_2) + g_2(x_2)$ 、 $f_2(s_2)$ 及び最適解 x_2^* は表 5.3 のとおりになる。

[第 3 ステップ] 農地 1 への割当： $j = 1$ に相当する。この場合は初期値として $s_1 = 5$ となる。このステップの最適方策の式はつぎのとおり。

$$f_1(s_1) = \max_{x_1 \leq s_1} [f_2(s_1 - x_1) + g_1(x_1)] \quad (5.6)$$

従って x_1 を変えた場合の $f_2(s_1 - x_1) + g_1(x_1)$ 、 $f_1(s_1)$ 、及び最適解 x_1^* は表 5.4 のとおり $x_1^* = 1$ となり、全利益は 6,100 万円となる。

表 5.3: 農地への人員配置：第 2 ステップ

s_2	$f_3(s_2 - x_2) + g_2(x_2)$						$f_2(s_2)$	x_2^*
	0	1	2	3	4	5		
0	0	—	—	—	—	—	0	0
1	0.8	1.0	—	—	—	—	1.0	1
2	1.6	1.8	1.9	—	—	—	1.9	2
3	3.6	2.6	2.7	3.3	—	—	3.6	0
4	4.0	4.6	3.5	4.1	4.2	—	4.6	1
5	4.8	5.0	5.5	4.9	5.0	5.0	5.5	2

表 5.4: 農地への人員配置：第 3 ステップ

s_1	$f_2(s_1 - x_1) + g_1(x_1)$						$f_1(s_1)$	x_1^*
	0	1	2	3	4	5		
5	5.5	6.1	5.9	4.9	4.6	4.0	6.1	1

つぎに、以上の表を逆にたどることにより、次の結論が得られる。

まず $s_2 = s_1 - x_1^* = 4$ であるから、表 5.3 より $x_2^* = 1$ となり、 $s_3 = s_2 - x_2^* = 3$ であるから表 5.2 より $x_3^* = 3$ となる。これをまとめると、

農地 1 : 1 人、農地 2 : 1 人、農地 3 : 3 人、全収益 : 6,100 万円

倉庫運営問題 [応用例 - 3]

ここではある大きさの倉庫を利用し、価格の安いときに大量に購入し、高いときに販売して利益を挙げるための方策を動的計画法で求めてみる。制約条件としては倉庫の大きさである。また方策を決めるためには購入及び販売価格の予想が必要であり、これらのものは与えられているとする。

まず倉庫の大きさは 300 個保管できるものとする。ここでは簡単のため 3 カ月の期間で考えてみる。3 カ月間の購入及び販売価格の予想値は表 5.5 で与えられているとする。また購入は販売後に行なうものとする。最初の在庫量は 100 個とし、3 カ月経った時には零にしてもよいとする。

[解答] k 月の販売量を x_k 、購入量を y_k 、販売価格を p_k 、購入価格を c_k 、倉庫での在庫量を s_k とし、倉庫の能力を $H (= 300)$ とすると、動的計画法の式はつぎのように表わすことができる。ここで $f_k(s_k)$ は k での最大利益 (最終月よりの利益の合計の最大値) とする。

表 5.5: 購入価格と販売価格の推移

月 (i)	購入単価 (1 個) (c_i)	販売単価 (1 個) (p^i)
1	162	160
2	160	165
3	165	185

$$f_k(s_k) = \max_{x_k, y_k} [f_{k+1}(s_k - x_k + y_k) + (p_k x_k - c_k y_k)] \quad (5.7)$$

$$k = 1, 2, 3; \text{ ただし } 0 \leq x_k \leq s_k, 0 \leq y_k \leq H - (s_k - x_k)$$

[第 1 ステップ] 最終月 $k = 3$ の最大利益の式 $f_3(s_3)$ を求める : 最終月では在庫を零にしてもよい, すなわち $s_4 = 0$ でよいので, 在庫量はすべて売りきり, 新しく購入する必要もない. 従って,

$$x_3 = s_3, y_3 = 0$$

これから

$$f_3(s_3) = p_3 s_3 = 185 s_3 \quad (5.8)$$

[第 2 ステップ] $k = 2$ での式 $f_2(s_2)$ を求める : 最大利益の式に第 1 ステップの結果を入れる, $s_3 = s_2 - x_2 + y_2$ の関係を代入すると, つぎの式が得られる.

$$\begin{aligned} f_2(s_2) &= \max_{x_2, y_2} [f_3(s_2 - x_2 + y_2) + (p_2 x_2 - c_2 y_2)] \\ &= \max_{x_2, y_2} [p_3(s_2 - x_2 + y_2) + (p_2 x_2 - c_2 y_2)] \\ &= \max_{x_2, y_2} [(p_2 - p_3)x_2 + (p_3 - c_2)y_2 + p_3 s_2] \end{aligned} \quad (5.9)$$

$$0 \leq x_2 \leq s_2, 0 \leq y_2 \leq H - (s_2 - x_2)$$

ここで

$$p_2 - p_3 = 165 - 185 = -20 \leq 0$$

$$p_3 - c_2 = 185 - 160 = 25 \geq 0$$

の関係から $f_2(s_2)$ を最大にするには x_2 を最小に, y_2 を最大にすればよいことが分かる. したがって

$$x_2 = 0, y_2 = H - s_2 + x_2 = H - s_2$$

となり

$$\begin{aligned} f_2(s_2) &= (p_3 - c_2)(H - s_2) + p_3 s_2 \\ &= c_2 s_2 + (p_3 - c_2)H = 160s_2 + 25H \end{aligned} \quad (5.10)$$

が得られる。

[第3ステップ] $k = 1$ として $f_1(s_1)$ を求める：第2ステップと同様にして次式が得られる。

$$\begin{aligned} f_1(s_1) &= \max_{x_1, y_1} [f_2(s_1 - x_1 + y_1) + (p_1 x_1 - c_1 y_1)] \\ &= \max_{x_1, y_1} [c_2(s_1 - x_1 + y_1) + (p_3 - c_2)H + (p_1 x_1 - c_1 y_1)] \\ &= \max_{x_1, y_1} [(p_1 - c_2)x_1 + (c_2 - c_1)y_1 + (p_3 - c_2)H + c_2 s_1] \end{aligned} \quad (5.11)$$

$$0 \leq x_1 \leq s_1, \quad 0 \leq y_1 \leq H - (s_1 - x_1)$$

また

$$p_1 - c_2 = 160 - 160 = 0$$

$$c_2 - c_1 = 160 - 162 = -2 \leq 0$$

の関係から $f_1(s_1)$ を最大にするには x_1 は不定(何でも可), y_1 は最小にすればよい。したがって,

$$x_1(\text{不定}) \leq s_1, \quad y_1 = 0$$

となり,

$$f_1(s_1) = (p_3 - c_2)H + c_2 s_1 = 25H + 160s_1 \quad (5.12)$$

また与えられた条件より

$$H = 300, \quad s_1 = 100$$

[第4ステップ] 第3ステップの $k = 1$ にて初期条件が与えられたので, 今度は逆に $k = 1 \rightarrow 3$ と計算すれば各月の値が求められる。

$$f_1(s_1) = 25H + 160s_1 = 25 \times 300 + 160 \times 100 = 23,500$$

$$x_1(\text{不定}) \leq 100, \quad y_1 = 0$$

$$s_2 = s_1 - x_1 + y_1 = s_1 - x_1 = 100 - x_1$$

$$f_2(s_2) = 160s_2 + 25H = 23,500 - 160x_1$$

表 5.6: 最適運営計画

月	在庫量 s_k	販売量 x_k	購入量 y_k
1	100	$x_1 (\leq 100)$	0
2	$100 - x_1$	0	$200 + x_1$
3	300	300	0

$$x_2 = 0, y_2 = H - s_2 = 200 + x_1$$

$$s_3 = s_2 - x_2 + y_2 = H = 300$$

$$f_3(s_3) = 185s_3 = 55,500$$

$$x_3 = 300, y_3 = 0$$

以上を表にまとめると表5.6が得られ、総利益は23,500円となる。

資金配分問題 [応用例 - 4]

これまでは数値が離散値で与えられている問題を扱ったが、ここでは連続値で与えられる場合について見てみる。

総計 A 円の資金を N 個のプロジェクトに配分する際に、利益を最大にする配分方法を求める問題を考える。その際、各プロジェクトの挙げる利益は配分された資金の p 乗 ($0 < p < 1$) に比例するとする。

[解答] N 個のプロジェクトに配分する方法をプロジェクト 1 から順次 N まで配分すると考えると N 段の動的計画法の問題として扱うことができる。まず記号をつぎのように定める。

- 各プロジェクトへの配分資金 : $x_k, k = 1, 2, \dots, N$
- k プロジェクトに来たときに残っている資金 : $s_k, k = 1, 2, \dots, N$

最適性の原理はつぎのように表わせる。

$$f_k(s_k) = \max_{0 \leq x_k \leq s_k} [f_{k+1}(s_k - x_k) + g(x_k)] \quad (5.13)$$

$$k = 1, 2, \dots, N$$

ただし、 $g(x_k) = x_k^p, 0 < p < 1, s_1 = A$ 。

これを解くには最後のプロジェクトへの配分 $k = N$ より開始して逆向きに $k = 1$ まで配分する方法を採用する。

[第 1 ステップ] $k = N$ のとき : 残っている資金は残さず , すべてこのプロジェクトに配分できるので , $f_N(s_N)$ はつぎのようになる .

$$\begin{aligned} f_N(s_N) &= \max_{0 \leq x_N \leq s_N} [g(x_N)] \\ &= \max_{0 \leq x_N \leq s_N} [x_N^p] = s_N^p \end{aligned} \quad (5.14)$$

$$x_N = s_N \quad (5.15)$$

[第 2 ステップ] $k = N - 1$ のとき : 最適方策の式はつぎのようになる .

$$\begin{aligned} f_{N-1}(s_{N-1}) &= \max_{0 \leq x_{N-1} \leq s_{N-1}} [f_N(s_{N-1} - x_{N-1}) + g(x_{N-1})] \\ &= \max_{0 \leq x_{N-1} \leq s_{N-1}} [(s_{N-1} - x_{N-1})^p + x_{N-1}^p] \end{aligned} \quad (5.16)$$

これを解くために $y = (s - x)^p + x^p$ と置いて y の最大値を求める .

$y' = -p(s - x)^{p-1} + px^{p-1}$ にて $y' = 0$ と置いて $x = (1/2)s$.

$y'' = p(p - 1)((s - x)^{p-2} + x^{p-2})$ は $0 < p < 1$ において常に $y'' < 0$ となるので , 上記は最大値となる .

以上より

$$x_{N-1} = \frac{1}{2}s_{N-1} \quad (5.17)$$

$$f_{N-1}(s_{N-1}) = 2\left(\frac{1}{2}s_{N-1}\right)^p \quad (5.18)$$

[第 3 ステップ] $k = N - 2$ のとき : 最適方策の式はつぎのようになる .

$$\begin{aligned} f_{N-2}(s_{N-2}) &= \max_{0 \leq x_{N-2} \leq s_{N-2}} [f_{N-1}(s_{N-2} - x_{N-2}) + g(x_{N-2})] \\ &= \max_{0 \leq x_{N-2} \leq s_{N-2}} \left[2\left(\frac{1}{2}(s_{N-2} - x_{N-2})\right)^p + x_{N-2}^p\right] \end{aligned} \quad (5.19)$$

前のステップと同様にして微分及び二次微分を採って最大値を求めると , 次式が得られる .

$$x_{N-1} = \frac{1}{3}s_{N-2} \quad (5.20)$$

$$f_{N-2}(s_{N-2}) = 3\left(\frac{1}{3}s_{N-2}\right)^p \quad (5.21)$$

[第 4 ステップ] 同様にして一般に $k = N - i$ のときの解は数学的帰納法を適用してつぎのように得られる .

$$x_{N-i} = \frac{1}{i+1}s_{N-i} \quad (5.22)$$

$$f_{N-i}(s_{N-i}) = (i+1)\left(\frac{1}{i+1}s_{N-i}\right)^p \quad (5.23)$$

表 5.7: 在庫管理

月 i	販売価格 p_i	購入価格 c_i
1	160	162
2	165	160
3	185	165

[第5ステップ] $k = 1$ のときの解は以下になる .

$$x_1 = \frac{1}{N} s_1 = \frac{1}{N} A \quad (5.24)$$

$$f_1(s_1) = N \left(\frac{1}{N} s_1 \right)^p, \quad s_1 = A \quad (5.25)$$

[第6ステップ] $k = 1$ より $k = N$ まで順次計算することにより ,

$$x_1 = x_2 = \dots = x_N = \frac{1}{N} A \quad (5.26)$$

となり , 最適解は均等配分であることが分かる . また , このときの総利益は $f_1(A) = N \left(\frac{1}{N} A \right)^p$ である .

演習問題

5.1 ある製品 A を 300 個まで収容できる倉庫がある . この製品の販売価格と購入価格の 3 カ月間にわたっての各月の変動が表 5.7 のとおりであるとする . 倉庫の最初の在庫が 100 個として , 3 カ月を終わった時点での利益を最大にするには毎月どのように販売し , 購入すればよいか . なお購入は販売後に行うものとする .

その他の最適化手法

[概要] 線形計画法，動的計画法以外の最適化手法について述べる．非線形システムの最適化手法として代表的な勾配法，最近のニューラルネットワークによる方法，遺伝的アルゴリズム，および多目的最適化について説明している．

勾配法

勾配法 (gradient method) は山登り法あるいは最大傾斜法 ともいわれ，非線形システムの最適化手法として，よく使用される．目的関数とその変数に関して微分可能の条件のほかには適用上の制約はない．基本的には目的関数の変数に関する勾配方向，すなわち，最大変化をする方向にたどって極値に近づく方法である．ただ，極値解が最適解であるとの保証はない．いま，状態値を時間の関数の $x(t)$ とし，目的関数を $f(x)$ で表す．ある点 $x_0 = x(t_0)$ の近傍に $f(x)$ の極大値があることが知られているとして，最短時間で極大値に近づく経路を考える．その場合 $f(x)$ の x に関する最大の傾斜，すなわち勾配方向に進むのが最適方策と考えられる．この考えに基づく運動方程式は次の式で表される．

$$\frac{dx(t)}{dt} = \nabla f(x) \quad (6.1)$$

なお，極小値に近づく場合は $-\nabla f(x)$ (∇ は nabla と読む) と負符号をつける．ここで

$$\nabla f(x) = \text{grad} f(x) \quad (6.2)$$

$$= \left[\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right]^T \quad (6.3)$$

$$x \in R^n \quad (6.4)$$

数値計算を行う時には時間は離散値になるので，式 (6.1) を差分式に変換すると

$$\begin{aligned} x(t + \Delta t) &= x(t) + \Delta x(\Delta t) \\ &= x(t) + \Delta t \cdot \nabla f(x) \end{aligned} \quad (6.5)$$

となる． $t = t_0$ より開始し，上記式に基づいて $x(t)$ を進めればよい． Δt を大きくし過ぎると乱調をおこし収束をしなくなるので，適正值に選定せねばならない， Δt として一定値に固定する方法が最も簡単ではあるが，適正值に選定するのが難しい．その対策として以下に述べる方法により，可変として最適化を図る方法が考えられている．

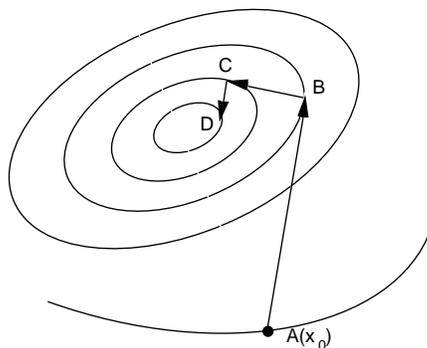


図 6.1: 最適勾配法

6.1.1 最適勾配法

最適勾配法 (optimal gradient method) は最急降下法 (steepest descent method) とも言われ、 $f(x(t + \Delta t))$ が最大となるように $x(t)$ を $\nabla f(x(t))$ の方向に進める方法である。すなわち

$$\max_{\Delta t} f(x(t + \Delta t)) = \max_{\Delta t} [f(x(t) + \Delta t \cdot \nabla f(x(t)))] \quad (6.6)$$

となる Δt を求め式 (6.5) により $x(t + \Delta t)$ を計算して、その方向に進める。つぎに、その点での勾配 $\nabla f(x)$ を求め、上記と同じ方法を繰り返す。この方法を用いた場合の $x(t)$ の軌跡を図 6.1 に示す。初期点 x_0 より開始し、 $\nabla f(x_0)$ の方向に進み、等高線 (同一の $f(x)$ の点を結んだ線) に接するまで行く。つぎにこの点の勾配の方向に再び等高線に接するまで進む。従って図のように直角に折れ曲がった軌跡をたどることになる。

Δt の具体的な決め方としては、例えば $\Delta t = T, 2T, \dots, MT$ (T は適当な定数) に対する目的関数 $f(x(t + \Delta t))$ の値を計算して、これが最大になる Δt を選べばよい。この方法の欠点は収束がよくない点で、1 回目で極大値に到達しない場合は無限回の繰返しになる恐れがある。その対策として、次の共役勾配法がある。

6.1.2 共役勾配法

共役勾配法 (conjugate gradient method) は共役方向に向かって直線探索を行う方法である。最初は最適勾配法により進み、以後はこれを基に互いに共役となる方向に探索を行う。共役の作り方により各種の方式があるので以下では Fletcher-Reeves 法と言われる方式

について説明する .

この方式ではつぎの手順で解の探索をすすめる . 極大値を求める例で示す .

1) 所期値として x_0 を選ぶ .

$$p_0 = \nabla f(x_0) \quad (6.7)$$

極小値を求める場合には負符号をつけ $-\nabla f(x_0)$ とする . $f(x_0 + \Delta t \cdot p_0)$ を最大とする Δt を計算して , これを Δt_0 とすると

$$x_1 = x_0 + \Delta t_0 \cdot p_0 \quad (6.8)$$

まで進める . ここまでは最適勾配法と同じである .

2) つぎに

$$p_1 = \nabla f(x_1) + \beta_1 \cdot p_0 \quad (6.9)$$

$$\beta_1 = \frac{|\nabla f(x_1)|^2}{|\nabla f(x_0)|^2} \quad (6.10)$$

を求め $f(x_1 + \Delta t \cdot p_1)$ を最大にする Δt を計算して Δt_1 とすると

$$x_2 = x_1 + \Delta t_1 \cdot p_1 \quad (6.11)$$

まですすめる .

3) 一般に i 回目まで繰返して , $i = 1, 2, \dots$

$$p_i = \nabla f(x_i) + \beta_i \cdot p_{i-1} \quad (6.12)$$

$$\beta_i = \frac{|\nabla f(x_i)|^2}{|\nabla f(x_{i-1})|^2} \quad (6.13)$$

を求め $f(x_i + \Delta t p_i)$ を最大にする Δt を計算して Δt_i とすると

$$x_{i+1} = x_i + \Delta t_i \cdot p_i \quad (6.14)$$

まですすめる .

上の関係を図で示したのが図 6.2 である . なお , この方式の特徴は収束の速いことである .

[註] 共役とは : k 個のベクトル $v_i, i = 1, \dots, k$ において , 対称行列 $Q \in R^{n \times n}$ に対して $v_i^T Q v_j = 0, i \neq j$ を満たす時 , 互いに共役であるという . $Q = I$ (単位行列) とすると , 前記の関係は直交性を意味するので , 共役は直交性を一般化したものである .

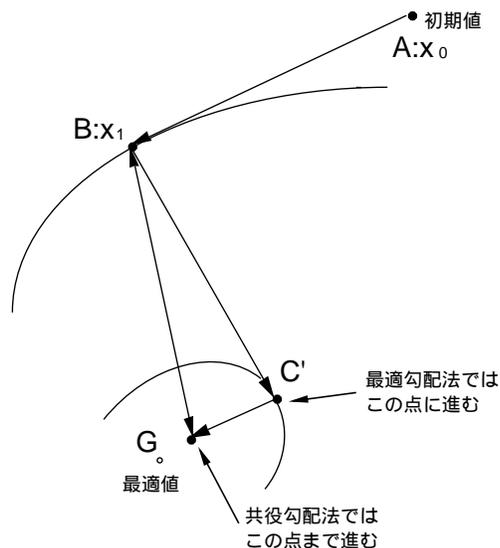


図 6.2: 共役勾配法

6.1.3 ニュートン法

関数 $f(x)$ をある x_k の近傍でテーラ展開し, つぎのように 2 次関数で近似して最適勾配法を適用する方式をニュートン法という.

$$f(x) \simeq f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k) \quad (6.15)$$

目的関数が正定の二次形式の場合は 1 回の探索で最小値が得られる. しかし一般の非線形の目的関数では二階微分の行列 (Hessian) を求め, さらに逆行列を計算しなければならないので時間がかかる欠点がある.

生物学的方法

近年, ニューラルネットワークや遺伝的アルゴリズムなど生物の機能を応用した方式が注目されている. ここでは最適問題の解法としてのこれらの応用について簡単に説明する.

6.2.1 ニューラルネットワーク

ニューラルネットワーク (NN) には多層形, 相互結合形など各種の方式が研究されているがここでは, 最適問題の解法に適している相互結合形 NN, 別名ホップフィールドネットワークについて述べる.

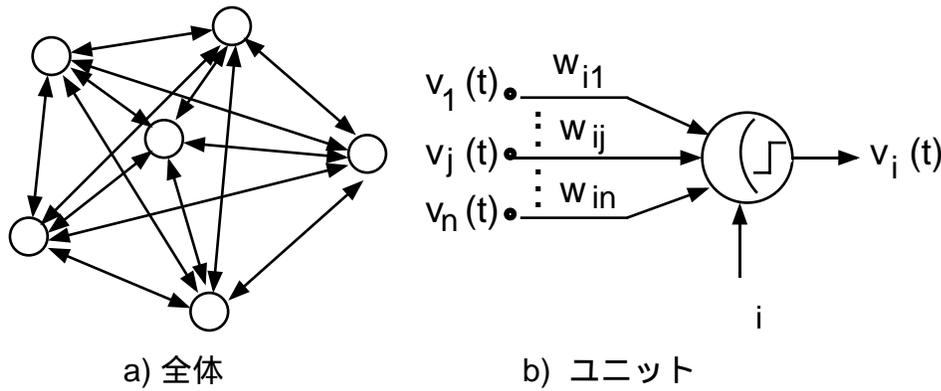


図 6.3: 相互結合形ニューラルネットワーク

図 6.3a) に示すよう n 個のユニットよりなる相互結合形 NN を考える．各ユニット間は相互に結合されており，結合の強さは対称であるとする．したがって，任意のユニット j ユニットから i ユニットへの結合の強さを w_{ij} で表わすと $w_{ij} = w_{ji}$ の関係が成り立つことになる．また，このネットワークは離散時間的に動作するとして離散時間 t における i ユニットの出力を $v_i(t)$ とし，しきい値を θ_i ，入力の合計を $u_i(t)$ とすると図 6.3b) より分かるようにつぎの関係が成り立つ．

$$u_i(t) = \sum_{j \neq i} w_{ij} v_j(t) + \theta_i \quad (6.16)$$

ここで次式で表わされるネットワークエネルギー を考える．

$$E(t) = -\frac{1}{2} \sum_i \sum_{j \neq i} w_{ij} v_i(t) v_j(t) - \sum_i \theta_i v_i(t) \quad (6.17)$$

ネットワークがつぎに示す時間発展規則で動作すると，このネットワークエネルギーが時間と共に減少することが証明できる．

- ステップ 0：最初にすべてのユニットの出力を乱数により 0 または 1 に設定して置く．
- ステップ 1：ネットワークよりランダムに 1 つのユニット i を選び出し，そのユニットへの入力の総和 $u_i(t)$ を求める．
- ステップ 2： $u_i(t)$ の値に応じてユニット i の $t+1$ 時間での出力 $v_i(t+1)$ を次式に従って更新する．

$$\begin{aligned} u_i(t) &\geq 0 \quad \text{なら} \quad v_i(t+1) = 1 \\ &< \quad \text{なら} \quad v_i(t+1) = 0 \end{aligned} \quad (6.18)$$

- ステップ 3 : ユニット i 以外のユニットの出力は変えない . すなわち

$$v_j(t+1) = v_j(t), \quad j = 1, 2, \dots, N, \quad j \neq i \quad (6.19)$$

- ステップ 4 : 所定の動作回数に達すれば終了 , 達しなければステップ 1 に戻る .

以上を行えばネットワークエネルギーが減少することは次のように証明できる .

$v_i(t+1)$ を上記規則で変化させた場合のエネルギーの変化の式を求めると

$$\begin{aligned} E(t+1) - E(t) &= -(v_i(t+1) - v_i(t)) \left(\sum_{i \neq j} w_{ij} v_j(t) + \theta_i \right) \\ &= -(v_i(t+1) - v_i(t)) u_i(t) \end{aligned} \quad (6.20)$$

式 (6.18) の関係からつぎが成り立つ .

$$\begin{aligned} u_i(t) \geq 0 \quad \text{の時} \quad v_i(t+1) - v_i(t) &\geq 0 \\ < \quad \text{の時} \quad v_i(t+1) - v_i(t) &\leq 0 \end{aligned} \quad (6.21)$$

したがって式 (6.20) により常に $E(t+1) - E(t) \leq 0$ すなわち

$$E(t+1) \leq E(t) \quad (6.22)$$

となって , エネルギーは常に減少または一定値の方向に動作し , 極小値に向かうことが分かる .

このホップフィールドネットワークを使用して最適問題を解くには , 求める解をユニットの出力 v_i で表現し , それに必要な数のユニットを設ける . 目的関数をエネルギー関数に選び , 最適値を求めるための条件や制約条件などにより相互結合係数 w_{ij} としきい値 θ_i を定めればよい . 後はユニットの出力を最初に乱数で設定し , 適当な回数だけ動作させればよい . 適用例としてはセールスマン巡回問題や棒材の最適切断問題 [6] などがある .

この方式の適用の際 , 考慮すべき事項はどのようなユニットを設けるか , 結合係数やしきい値をどのように定めるかである . その考え方の例としてセールスマン巡回問題を例に説明する .

[セールスマン巡回問題への適用]

セールスマン巡回問題は与えられた n 都市を順番にまわり , 出発点に戻るルートで巡回距離が最小になる巡回経路を求める問題である . その解き方について順に述べる .

[ニューラルネットワーク構成] 都市数を n とするとユニットは図 6.4 に示すように n^2 個を設ける . 縦方向は都市を , 横方向は巡回順序を表わしている . 発火しているユニット (出力が 1) を結んだ線が巡回経路になる . したがって発火しているユニットは各列 , 各行とも 1 つになっていなければならない .

[エネルギー関数] ネットワークのエネルギー関数はつぎのように定義する .

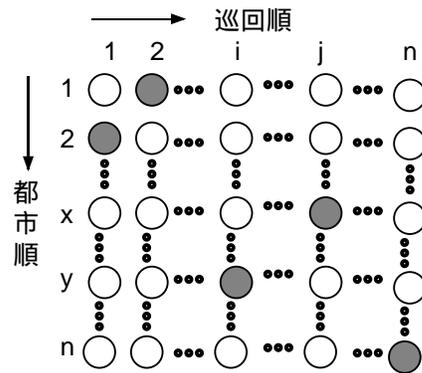


図 6.4: セールスマン巡回問題

$$E = -\frac{1}{2} \sum_x \sum_{y \neq x} \sum_i \sum_{j \neq i} w_{x,i,y,j} v_{x,i} v_{y,j} - \sum_x \sum_i \theta_{x,i} v_{x,i} \quad (6.23)$$

ここで x, y は都市を, i, j は順番を表わし, 時間の表示は省略している. また $w_{x,i,y,j}$ はユニット (x, i) と (y, j) 間の結合係数, $\theta_{x,i}$ はユニット (x, i) のしきい値をそれぞれ表わす.

[目的関数] 都市 x と y との間の距離を d_{xy} で表わすと巡回経路の全距離 E_1 は次式で表わされ, これを目的関数とする. したがって, 最小化問題となる.

$$E_1 = \frac{1}{2} \sum_x \sum_{y \neq x} \sum_i d_{x,y} v_{x,i} (v_{y,i+1} + v_{y,i-1}) \quad (6.24)$$

[制約条件] 制約条件を表わすものとしてつぎのエネルギー関数を導入する. これらのエネルギー関数の値はいずれも非負で, ゼロになった時に制約条件を満足していることが分かる.

$$\text{同じ都市を 2 回通らない: } \frac{1}{2} \sum_x \sum_i \sum_{j \neq i} v_{x,i} v_{x,j} \quad (6.25)$$

$$\text{同時に 2 つの都市を通らない: } \frac{1}{2} \sum_i \sum_x \sum_{y \neq x} v_{x,i} v_{y,i} \quad (6.26)$$

$$\text{通過する都市の数は } n: \frac{1}{2} \left(\sum_x \sum_i v_{x,i} - n \right)^2 \quad (6.27)$$

[ネットワーク定数の決定] 目的条件のエネルギーと制約条件のエネルギーを加算したエネルギーを求め, これですべてネットワークを動作させる. なおこの加算の際に 3 つの制約条件のに対してはそれぞれ重み A, B, C を設定する. これら制約条件を決める重みは目的関数に対する重み D より相対的に大きく設定する. この式を式 (6.23) の形になるように変

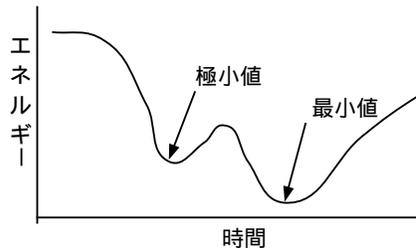


図 6.5: ローカルミニマム

形する．その際， $v_{*,*}$ の値は 0 か 1 であるので， $v_{*,*}^2 = v_{*,*}$ の関係を利用する．その結果，結合係数，しきい値がつぎのように得られる．

$$w_{x,i,y,j} = -A\delta_{x,y}(1 - \delta_{i,j}) \quad (6.28)$$

$$-B\delta_{i,j}(1 - \delta_{x,y})$$

$$-C$$

$$-Dd_{xy}(\delta_{j,i+1} + \delta_{j,i-1})(1 - \delta_{x,y})$$

$$\theta_{xi} = Cn \quad (6.29)$$

ここで $\delta_{i,j}$ はデルタ関数である．

(セールスマン巡回問題終了)

ホップフィールドニューラルネットワーク方式の問題点のひとつにローカルミニマム，すなわち極小解に陥る場合があることである．図 6.5 に示すように，初期値の状況によっては，そこから最小値に向かう途中に極小値が存在するとそこに陥って出てこれない場合がある．その対策としてはユニットの出力信号に雑音信号を重畳し，収束に近づくにつれてその値を小さくするシミュレーテッド・アニーリングの考えを導入することも行なわれている．

6.2.2 遺伝的アルゴリズム

遺伝的アルゴリズム (GA: Genetic Algorithms) は生物における遺伝情報の伝達をとおしての進化の機構を取り入れたアルゴリズムであり，特に最適問題に適している．各種の方式があるがここではシンプル GA とされる基本的な方式を中心に概要を説明する．

まず所定数の個体からなる集団を考える．各個体には必要数の遺伝子が収納されている．解くべき問題を遺伝子で表現する．遺伝子には各種の表現方法があるが，基本的なものはバイナリビットで表現したストリングである．それより問題に応じて適合値を計算する方法すなわち適合関数を定めて置く．以上に対し GA はつぎのように動作する (図 6.6 を参照)．

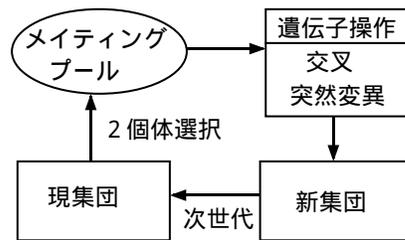


図 6.6: 遺伝的アルゴリズム

- ステップ 0 : 準備段階として、現集団にあるすべての個体の遺伝子を乱数を用いてランダムに設定して置く。
- ステップ 1 : 集団より 1 組を選ぶ。この際、各個体の適合値に比例した確率で選択する(ルーレットホイール選択方式)。
- ステップ 2 : この 1 組に対して交叉と突然変異の遺伝子操作を行なう。交叉はこの 1 組の間で一部の遺伝子を交換するもので 1 点で交叉、2 点で交叉、多点で交叉など各種の方式がある。突然変異は遺伝子の一部を変更するものである。これらを行なうか否か、およびその位置は確率的に決定する。
- ステップ 3 : こうして得られた新しい遺伝子はつぎの世代のための新集団に入れる。
- ステップ 4 : 新集団の個体数が所定の数になるまで、ステップ 1 から繰り返す。
- ステップ 5 : 新集団を現集団に移す。
- ステップ 6 : ステップ 1 に戻り、つぎの世代の実行を行なう。所定世代実行したら終了。

以上の動作により世代の進行に応じて集団全体として適合値の最大値すなわち最適解に向かって進む。最良の個体を残すエリート戦略を採用すると各世代の最良値は常に非減少で増加する。

GA を最適問題に適用する際は、先ず問題を遺伝子で表現し、それより目的値を求める方策すなわち目的関数を定め、これを適合関数に変換する。適合値は増加方向に進行するので、目的値を最小にする問題の場合はこの変換が必要となる。遺伝子表現としてはバイナリビット表現のほか整数、実数表現などもあり、さらに整数の順序表現などもある。例えば、セールスマン巡回問題への適用の場合は 1 から都市数までの整数の順序表現とし、この順序を変えて各種の巡回経路を表現する。順序表現での交叉や突然変異の方法は他のものと異なり、各種の方式が研究されている。詳しくは文献 [8] などを参照されたい。

遺伝的アルゴリズムの特長は対象のシステムのモデルがなくても適用できることであり、目的関数さえ求められておればよい。また確率的に動作するので、ローカルミニマムに陥ることがないのも大きな特長である。各種の問題に適用されている [8] [9] [30]。

多目的最適化

これまでは単一の目的関数を最適化する問題を主として扱ってきた。しかし、実際問題では複数の目的関数を最適化したいとの問題が多い。例えば性能を高めてかつ信頼性も挙げたい、コストを最小にして性能も挙げたいなどである。これらすべての要求を満たす解が得られる場合もあるが、多くの場合すべてを満足させることはできず、妥協を必要とする。ここではこのような問題の解決方法について検討する [3], [9]。

6.3.1 問題の定義と解の性質

[多目的最適化問題の定義] $\mathbf{x} \in R^n$ を n 次のベクトルとしたとき、多目的最適化問題はつぎのように表わせる。

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad (6.30)$$

$$\text{subj. to } \mathbf{x} \in X = \{\mathbf{x} | g(\mathbf{x}) \leq 0\} \quad (6.31)$$

ここで $f(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_p(\mathbf{x})]$, $g(\mathbf{x}) = [g_1(\mathbf{x}), \dots, g_m(\mathbf{x})]$ とする。また $f(\mathbf{x})$ は最適化しようとする多目的関数を表わし、 $g(\mathbf{x})$ は制約条件を表わしている。 (定義終わり)

ここでは最小化問題として表現しているが、実際は最大化問題もある。しかし以下の議論では最小化問題として扱っても一般性を失うことはないだろう。

さて、以下の論議のため用語を定義して置く。

[優越関係の定義] 二つの変数 $\mathbf{x}^1, \mathbf{x}^2$ 間において

* $f(\mathbf{x}^1) \leq f(\mathbf{x}^2)$ のとき、 \mathbf{x}^1 は \mathbf{x}^2 に優越するという。

* $f(\mathbf{x}^1) < f(\mathbf{x}^2)$ のとき、 \mathbf{x}^1 は \mathbf{x}^2 に強い意味で優越するという。

ここで \leq は各要素間に \leq の関係があるとともに、少なくとも1個の要素は $<$ の関係があることを意味する。すなわち

$$\mathbf{a} \leq \mathbf{b} \Leftrightarrow a_i \leq b_i (\forall i = 1, \dots, p) \text{ かつ } a_i < b_i (\exists i = 1, \dots, p)$$

とする (定義終わり)

[解の定義] \mathbf{x}^* を最適解とするとき、つぎの種類の最適解が存在する。

- \mathbf{x}^* に強い意味で優越する $\mathbf{x} \in X$ が存在しないとき、 \mathbf{x}^* を弱パレート最適解という。
- \mathbf{x}^* に優越する $\mathbf{x} \in X$ が存在しないとき、 \mathbf{x}^* を (強)パレート最適解という。

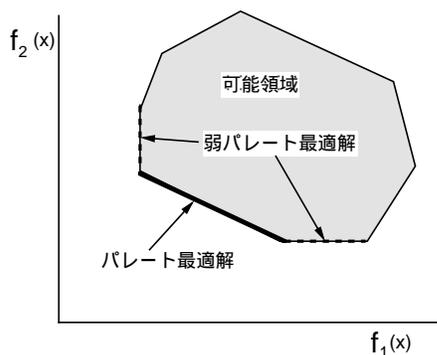


図 6.7: パレート最適解

c. x^* が任意の $x \in X$ に対して優越するとき, x^* を (完全) 最適解という.

(定義終わり)

簡単な例として目的関数が二つのときのパレート最適解の状況を図6.7に示す. 図では完全最適解が得られない場合を示しており, このような場合はパレート最適解が合理的な解であることが分かる. パレート最適解はこのようにひとつの解ではなく, 解の集合となっている.

6.3.2 多目的最適化の解の求め方

完全最適解が得られる場合は問題がないので, ここではパレート解しか得られない場合を対象に説明する. パレート解の場合はひとつの解ではなく, 解集合であるので問題によってはそれをさらに絞り込む必要がある. その方法として各種の方式が研究されているが, ここでは代表的なものについて説明する.

1) 加重和により単一目的化する方法

以下のように重みパラメータを使用して加重和を求め, 目的関数をひとつの関数に集約して単一目的関数として解く方法である. すなわち

$$\min_{\mathbf{x}} \sum_{i=1}^p w_i f_i(\mathbf{x}) \quad (6.32)$$

ただし, 重みは正すなわち $w_i \geq 0, i = 1, \dots, p$ とする. また重みは正規化して $\sum_{i=1}^p w_i = 1$ としてもよい.

これによりパレート解の一つが最適解として得られる. 重みを変えることにより選択するパレート解が変わってくるので, 目的を考えながら重みを設定すればよい. したがって重みの決め方に意志決定者の考えが入ることになる. この方法は簡単であるので, 使用されることが多い.

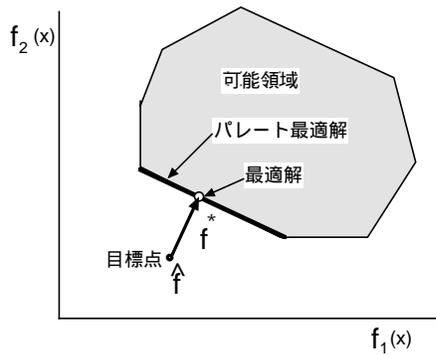


図 6.8: 目標設定法

2) 目標を設定しそれに近い解を求める方法

予め各目標関数ごとに目標点 \hat{f}_i , $i = 1, \dots, p$ を定め, できるだけそれに近い最適解を求める方法. 目標点としては可能領域の外に設け, 距離としては以下のような q 次のノルムを用いる.

$$\|\mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x})\|_q = \left\{ \sum_{i=1}^p |f_i(\mathbf{x}) - \hat{f}_i(\mathbf{x})|^q \right\}^{1/q} \quad (6.33)$$

q は問題の性質を考えながら適当に設定する必要がある. $q = 2$ の時は通常の距離を最小にする解となる. 図 6.8 に 2 目的関数の場合を示す.

3) 優先順位を定め順次決定する法

目的関数に優先順位をつけ, その順位の高いものから順に最適条件を満足する解を求めて行く方法である. 順位の決め方により解が異なってくるので, そこに意志決定者の考えが入ることになる.

4) パレート解を求め, 別の方法で絞り込む方法

最初の段階で最適解を絞り込むことはせずに, パレート解を求め, それに対して例えばエキスパートシステムなど別の方法により, 意志決定者の意向を考えつつ最適解に到達する方法である. パレート解を求める方法としては例えば遺伝的アルゴリズムを使用する方法などが研究されている.

制約条件つき最適化

これまでも最適問題における制約条件についてその都度述べてきたが, この制約条件の取り扱いについて今一度考えてみる. 制約条件付きの最適化問題は多くの研究がある [18][21] が, ここでは関数で表わされた等号制約条件のものと制約条件などが関数ではな

い一般のものの二つについて述べる．特に前者についてはラグランジュ乗数法について述べる．

6.4.1 ラグランジュ乗数法

目的関数および制約条件が微分可能で，かつ制約条件が等号で与えられる次式の最適問題を考える．ここで制約条件としては m 個の等号式で与えられているとする．

$$\text{目的関数： } f(x_1, \dots, x_n) \text{ を最大 (または最小) にする} \quad (6.34)$$

$$\text{制約条件： } g_i(x_1, \dots, x_n) = 0, \quad i = 1, \dots, m \quad (6.35)$$

これに対してはラグランジュ乗数 $\lambda_i, i = 1, \dots, m$ を導入して，つぎの制約なしの最適問題として解くことができる．

$$f(x_1, \dots, x_n) + \sum_{i=1}^m \lambda_i g_i(x_1, \dots, x_n) \text{ を最大 (または最小) にする．} \quad (6.36)$$

これを解くには，これを新しい関数 f として

$$\frac{\partial f}{\partial x_i} = 0, \quad i = 1, \dots, n \quad (6.37)$$

$$\frac{\partial f}{\partial \lambda_i} = 0, \quad i = 1, \dots, m \quad (6.38)$$

の連立方程式を解き， λ を消去して， $x_i, i = 1, \dots, n$ を求めればよい．何故なら式 (6.38) より制約条件の $g_i(x_1, \dots, x_n) = 0, i = 1, \dots, m$ が出てくるので，制約条件を満足することになる．

6.4.2 一般の場合

実際の問題では制約が不等号条件であったり，微分可能な関数式で表わされない場合や離散的な場合が多い．このような場合に用いられる方法としては

- 制約条件を目的関数に含め，多目的最適化問題として扱う．
- アルゴリズムにより制約条件を扱う．例えば制約条件を織り込み，これを常に満足するような方式を考える，あるいは制約条件を満足するように変数を修正する修正アルゴリズムを導入するなどがある．

ここでは前者の方法について考えてみる．問題を最小化問題と仮定し，制約条件がつぎの不等号で与えられているとする．

$$\text{制約条件： } g_i(x_1, \dots, x_n) > 0, \quad i = 1, \dots, m; \text{ かつできるだけ小さい} \quad (6.39)$$

この場合目的関数としてつぎのものを考え，その最小化を行なう．

$$\text{目的関数： } f(x_1, \dots, x_n) + \sum_{i=1}^m w_i g_i^2(x_1, \dots, x_n) \quad (6.40)$$

ここで $g_i(*)$ の二乗を使用するのはこの値を常に零以上にするためである．またパラメータ w_i , $i = 1, \dots, m$ は重みであり，上記式で制約条件の項の値が目的関数の項に比べて十分に大きくなるように大きな値に選ぶ．その具体的な値はシミュレーションの結果などを基に決定する．これは先に述べた多目的最適化における加重法を採用したことになる．場合によっては $g_i(*)$ の値がマイナスになる可能性があるため，その場合は取り除くか修正するアルゴリズムを別に必要とする．

演習問題

6.1 つぎの関数の最小値を 1) 最適勾配法と 2) 共役勾配法の両方で求めて結果を比較せよ．

$$f(x) = (x_1 - 5)^2 + 2(x_2 - 4)^2$$

システム管理

[概要] システムの計画，設計，製作の工程の管理技法として広く用いられている P E R T ， C P M について述べる．

概要

ここで紹介する P E R T ， C P M は大規模システムの工程管理の手法として米国で開発されたものである．工程管理は大別して工程の作成とその実施状況の管理に分けられる．まず，前者の工程の作成では各単位作業の工程の積上げにより全体工程を作成するとともに，所定の期間で，経済的に作業を完成するように計画の練直しが必要になる．作業の待ち時間を減らし，人員のピ - クをならして平均化を図り，もし工程遅れがあれば，あいろとなる部分の強化を行い，余裕のある作業は人員を減らして費用の低減を図るなどの検討を行う．一方，実施状況の管理の面では，なんらかの異常事態により部分的に工程遅れが生じた場合には，全体工程への影響を調査し，必要な強化策を採って全体への影響を最小限に抑える必要がある．

このような検討は，システムが大規模になればなるほど各作業の相互の関係が複雑になり，容易ではない．これを計算機の支援のもとに行う技法が P E R T と C P M である．P E R T (Program Evaluation and Review Technique) は工程管理用で，個々の単位作業の工程の全体工程への影響を評価する技法である．C P M (Critical Path Method) は P E R T とともに用いられ，最も経済的な工程計画を作成するための技法である．以下では主として P E R T について説明する．

P E R T

図 7.1 に示す建築工程を例に説明する．データとしては文献 [16] のものを使用した．P E R T を実施するには表 7.1 に示すような各作業の所要日数などのデータを用意する．なお，所要人員は P E R T では必要がないが，人員の山積みを使用するので，ここに挙げて置いた．これを基に P E R T では，まず図 7.2 に示すアロ - ダイアグラムを作成する．これは作業を示す矢印 (arrow) と作業の開始点，終了点などの区切りを表す結合点 (node) よ

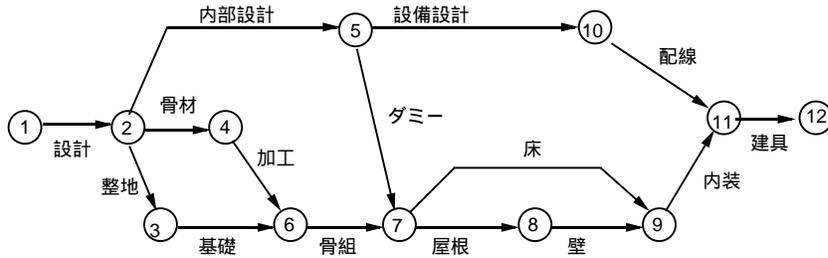


図 7.1: 建築工程の例

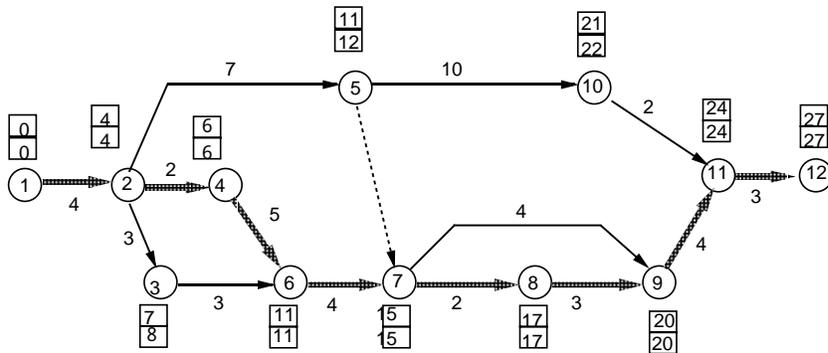


図 7.2: アローダイヤグラムの例

り成立っている．このように結合点は直前の作業の終了点でもあり，後続の作業の開始点でもある．矢印の横には作業名とその所要時間を記入している．また，点線で示す作業はダミー（dummy）作業といわれ，実質的な作業ではなく，作業手順上の条件を示している．すなわち，ダミー作業の開始点の条件が整わなければ終了点につながる，後続の作業が開始できないことを意味している．

このようにして作成したアロ - ダイヤグラムをもとに次の検討を行う．

1) 最早結合点時刻の計算

最早結合点時刻（earliest node time）とは，各結合点について最も早く取掛かれる時刻である．つまり，この時刻より早くはその作業に取掛かれない．これは全作業の開始点よりアロ - ダイヤグラムの矢印に沿って前向きにたどり，それ以前の各作業時間の和として求める．並行作業がある場合には，その区間の作業の最長の時間を採用する．したがって，作業 (i, j) を考え，作業時間を D_{ij} ，結合点 i の最早結合点時刻を ET_i と表すと，

$$ET_1 = 0 \quad (7.1)$$

表 7.1: PERT用入力データ

k	(i,j)		作業名	所要日	所要人員
	始点 (i)	終点 (j)			
1	1	2	設計	4	3
2	2	3	整地	3	5
3	2	4	骨材	2	4
4	2	5	内部設計	7	2
5	3	6	基礎	3	6
6	4	6	加工	5	5
7	5	7	ダミー	0	
8	5	10	設備設計	10	3
9	6	7	骨組	4	4
10	7	8	屋根	2	4
11	7	9	床	4	3
12	8	9	壁	3	2
13	9	11	内装	4	3
14	10	11	配線	2	2
15	11	12	建具	3	3

- (註) 1. $i < j$ とする .
 2. (i,j)は小さい値から大きい値になるように並べる .
 3. ダミー作業も入れる .

$$ET_j = \max_i \{ET_i + D_{ij}\} \quad (7.2)$$

ここで i は並行作業を示している .

2) 最遅結合点時刻の計算

最遅結合点時刻 (latest node time) とは, 各結合点で遅くとも到達していなければならない時刻である . 換言すると, システムの完成期限よりみて, 遅くともその時刻までには作業を完了していなければならない . 全作業の終了点よりアロ - ダイアグラムを後向きにたどりながら各作業時間の和として求める . 並行作業がある場合は前項と同じく最長の時間を採用する . したがって, 結合点 i の最遅結合点時刻を LT_i と表すと,

$$LT_n = ET_n \quad (7.3)$$

$$LT_i = \min_j \{LT_j - D_{ij}\} \quad (7.4)$$

ここで j は並行作業を示している .

3) 各種時刻の計算

最早結合点時刻 , 最遅結合点時刻 , 作業時間をもとに次の各種時刻を計算する .

今 , 作業 (i, j) (i, j は結合点の番号を表す) を考えると ,

- 最早開始時刻 (earliest start time) ES_{ij}

作業 (i, j) が最も早く取りかけられる時刻であり , 次式で求める .

$$ES_{ij} = ET_i \quad (7.5)$$

- 最早終了時刻 (earliest finish time) EF_{ij}

作業 (i, j) が最も早く終わる時刻で次式で求める .

$$EF_{ij} = ES_{ij} + D_{ij} \quad (7.6)$$

- 最遅開始時刻 (latest start time) LS_{ij}

作業 (i, j) について完成時刻よりみて最も遅らせて開始できる時刻 .

$$LS_{ij} = LT_j - D_{ij} \quad (7.7)$$

- 最遅終了時刻 (latest finish time) LF_{ij}

作業 (i, j) について完成時刻よりみて最も遅らせて終了させてよい時刻 .

$$LF_{ij} = LT_j \quad (7.8)$$

4) 余裕時間の計算

上記各時刻を基につぎの余裕時間を計算する .

a) 全余裕時間 (total float)

他の作業を極限までに前後にずらした時に得られる余裕時間であり , 作業 (i, j) の全余裕時間 TF_{ij} は次式で求められる .

$$TF_{ij} = LS_{ij} - ES_{ij} \quad (7.9)$$

b) 自由余裕時間 (free float)

後続の作業に影響なくずらすことのできる余裕時間であり , 作業 (i, j) の自由余裕時間 FF_{ij} は次式で求められる .

$$FF_{ij} = ET_j - EF_{ij} \quad (7.10)$$

表 7.2: 各種時間

作業 i-j	D_{ij}	LT_i	ET_j	ES_{ij}	EF_{ij}	LS_{ij}	LF_{ij}	TF_{ij}	FF_{ij}
1-2	4	0	4	0	4	0	4	0	0
2-3	3	4	7	4	7	5	8	1	0
2-4	2	4	6	4	6	4	6	0	0
2-5	7	4	11	4	11	5	12	1	0
3-6	3	8	11	7	10	8	11	1	1
4-6	5	6	11	6	11	6	11	0	0
5-10	10	12	21	11	21	12	22	1	0
6-7	4	11	15	11	15	11	15	0	0
7-8	2	15	17	15	17	15	17	0	0
7-9	4	15	20	15	19	16	20	1	1
8-9	3	17	20	17	20	17	20	0	0
9-11	4	20	24	20	24	20	24	0	0
10-11	2	22	24	21	23	22	24	1	1
11-12	3	24	27	24	27	24	27	0	0

以上に述べた各種時間を先に挙げた建築工程について求めたのが表 7.2 である。

5) クリティカルパスの探索

クリティカルパス (critical path) とは、全作業の開始点から終了点の間で、最も時間を要する各作業を結んだ経路をいい、この経路の作業に要する時間より早くは全作業を終えることはできない。前項で計算した全余裕時間が零になる作業を結んだ経路がこれに相当する。図 7.2 (文献 [16]) では太線で示している。全体の工程を管理するには、このクリティカルパス中の作業に注目すればよい。また、工程の短縮を必要とする場合にはこのパスの作業について短縮を図ればよい。

6) 山くずしによるスケジュールの調整

人員などのピークを平坦化して経済性を図るため、余裕時間などを利用して作業計画を見直しをする。まず図 7.3 に示すような横軸に日時、縦軸に各作業に必要な人員を積算して記入した山積み図を作成する。この図を参考にしながら、ピークになっている部分の平坦化を図る。そのためには余裕時間を参照しながら作業を前後にずらすことを行なう。

7) ガントチャートの作成

アロ - ダイアグラムは前記の検討には有効であるが、日常の工程の管理には必ずしも適していない。図 7.4 に示すガントチャート (Gantt chart) は時間を横軸にして作業を示して

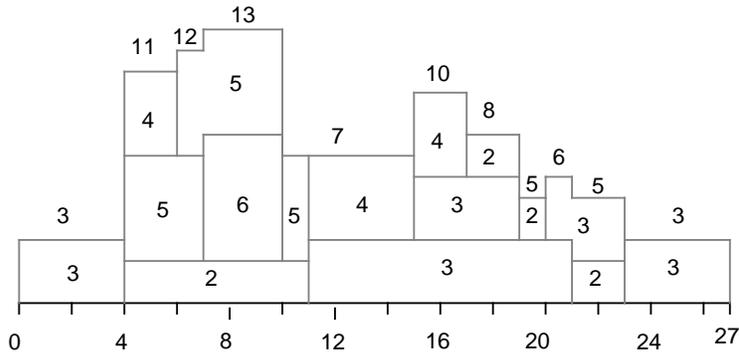


図 7.3: 山積み図

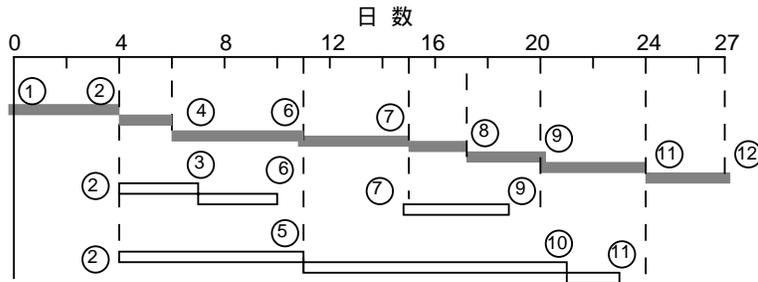


図 7.4: ガントチャート

いるので、各作業の進捗管理など日常の管理には便利であり、アロ - ダイアグラムに代って用いられている。アロ - ダイアグラムをもとに計算機を用いて簡単にガントチャートを作成することができる。

以上がPERTの概要である。PERTの効用はもちろん工程計画作成への支援であるが、日常の工程管理の面での支援も大きい。一度、アロ - ダイアグラムを計算機内に作成しておけば、ある作業時間に変更を生じた場合にも、新しい時間を入力することにより、最早結合点時刻、最遅結合点時刻、余裕時間、クリティカルパスなどが簡単に算出でき、工程の再検討が直ちにできる。

PERTを実際に適用する際の問題点としては各单位作業への分割、作業間の手順関係の検討や作業時間の見積りがある。特に作業の手順は、例で示したほど単純ではなく、部分的な並行作業や先発作業などによる工程短縮手段をとっているため、アロ - ダイアグラムは複雑になる。したがって多少の割り切りは必要であろう。

作業時間の見積りに対しては統計的なばらつきを考えて、次のようにして求めた期待

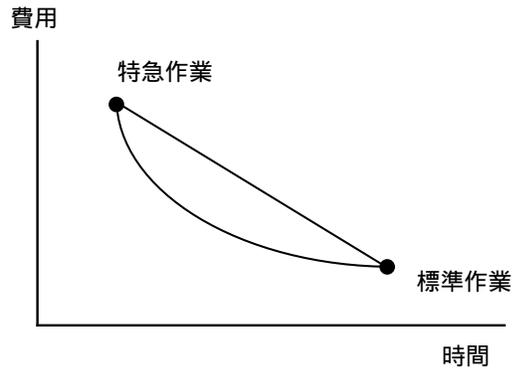


図 7.5: 費用勾配

時間 T_e を使用する場合が多い .

$$T_e = (T_o + 4T_m + T_p)/6 \quad (7.11)$$

T_o = 最も楽観的な見積時間

T_p = 最も悲観的な見積時間

T_m = 平均的な見積時間

C P M

PERTにより工程をたてた結果、クリティカルパスの工程が所定の期限を超過した場合には、各作業を見直して、いずれかの作業工程を短縮する必要がある。これを経済的に行う手法がC P Mである。一般に作業時間は投入資源（設備や人員）と関係があり、資源を投入することにより作業時間を短縮できる場合が多い。一般に図 7.5に示すような曲線になる。各作業の標準作業と最短の特急作業とにおける、作業時間と費用との関係を図のように直線近似した時の勾配をその作業の費用勾配といい、これを指標にしてつぎのようにして経済的な工期短縮を図る。

$$\text{費用勾配} = \frac{\text{特急費用} - \text{標準費用}}{\text{標準時間} - \text{特急時間}} \quad (7.12)$$

クリティカルパスにある作業の内、費用勾配が最小な作業を優先して短縮をおこなうのが最も経済的である。個々の作業で短縮できる時間に限界（特急時間）があるので、これを考慮して目的の時間短縮ができるまで、費用勾配が小さい作業から順に各作業の時間短縮を行えばよい。これは数理計画法の問題として解くことができる。ただ、平行作業が

ある場合にはクリティカルパスの作業を短縮して行けば、やがて、他の平行作業の方がクリティカルパスになる場合があるので、この点を折込んでの検討が必要である。

演習問題

7.1 卒業研究課題についてのPERTを行え。アローダイヤグラムを作成し、クリティカルパスを求め、各種時刻を計算し、そしてガントチャートを作れ。

システム信頼性

[概要] システムの重要な機能である信頼性について，信頼性に関する定義や指標，故障モデル，システムの信頼度の算定，信頼性解析技法などを中心にのべる．

信頼性の概念と評価指標

信頼性を論ずるに際しては，その概念や評価の尺度を明確にしておく必要がある．本節ではこれらに関する基礎的な事項について説明する．信頼性に関する用語は日本工業規格 JIS Z8115-1970 に規定されている．まず信頼性は

系，機器，部品などの機能の時間的安定性を表す度合または性質と定められている．一方，信頼度については

系，機器，部品などが，規定の期間中，規定の機能を遂行する確率と規定されている．このように信頼性は定性的な表現であるのに対し，信頼度は定量的な尺度といえる．

8.1.1 信頼度関数と故障率

N 個の同一特性をもつ要素よりなるシステムを考える．使用を開始して t 時刻および $t + \Delta t$ 時刻での正常な（故障していない）要素の数をそれぞれ $n(t)$ ， $n(t + \Delta t)$ とすると単位時間の故障頻度は次式となる．

$$\frac{n(t) - n(t + \Delta t)}{N} \frac{1}{\Delta t}$$

従って Δt 時間を零に近づけた極限值をもって故障密度関数（failure density function） $f(t)$ と定義する．すなわち

$$f(t) = \lim_{\Delta t \rightarrow 0} \frac{n(t) - n(t + \Delta t)}{N} \frac{1}{\Delta t} = -\frac{1}{N} \frac{dn}{dt} \quad (8.1)$$

故障密度関数の時間 t の積分値 $F(t)$ を累積故障分布関数（cumulative density function）または不信頼度関数という．

図 8.1: 信頼度関数と故障率の例

$$F(t) = \int_0^t f(t)dt \quad (8.2)$$

$$F(\infty) = \int_0^{\infty} f(t)dt = 1 \quad (8.3)$$

信頼度関数 (reliability function) は次のとおり定義する .

$$R(t) = 1 - F(t) = 1 - \int_0^t f(t)dt = \int_t^{\infty} f(t)dt \quad (8.4)$$

故障率 (failure rate) $\lambda(t)$ は単位時間に故障を起こす確率を表し , 上記で定めた関数を用いてつぎの通り定義する .

$$\lambda(t) = \frac{f(t)}{R(t)} = -\frac{1}{R(t)} \frac{dR(t)}{dt} = \frac{1}{1 - F(t)} \frac{dF(t)}{dt} \quad (8.5)$$

故障率は瞬間故障率 と言われることもある .

従って信頼度と故障率とはつぎの関係が成り立つ .

$$R(t) = \exp\left(-\int_0^t \lambda(t)dt\right) \quad (8.6)$$

上に定義した関数の一例を図 8.1 に示す .

8.1.2 信頼性の指標

信頼性に関するいくつかの重要な指標を挙げる .

1) M T B F (Mean Time Between Filure): 平均故障間隔

修理可能な設備において故障発生からつぎの故障発生までの時間の平均値であり , つぎのとおり定義される .

$$MTBF = \int_0^{\infty} t \cdot f(t)dt \quad (8.7)$$

故障率が時間に関して一定 λ の場合は

$$R(t) = \exp(-\lambda t) \quad (8.8)$$

$$f(t) = \frac{dF}{dt} = -\frac{dR}{dt} = \lambda \exp(-\lambda t) \quad (8.9)$$

$$\text{MTBF} = \int_0^{\infty} t \lambda \exp(-\lambda t) dt = \frac{1}{\lambda} \quad (8.10)$$

となる。

2) M T T F (Mean Time To Failure): 故障までの平均時間

故障したら修理せず廃却または新品に交換するユニットに対し使用開始から故障までの時間の平均値をいう。

M T B Fと同じく故障率一定の場合には次式で表される。

$$\text{MTTF} = \frac{1}{\lambda} \quad (8.11)$$

本指標は電子部品のように修理できないものに適用する。

3) M T T R (Mean Time To Repair): 平均修復時間

設備が修復のため休止している時間の平均値を示す。修理して再使用可能な設備に適用する。

M T B Fの時と同じく、次式により修理率 μ を定義できる。

$$\mu = \frac{1}{\text{MTTR}} \quad (8.12)$$

4) M U T (Mean Up Time): 平均動作可能時間

動作可能な時間の平均値。

5) M D T (Mean Down Time): 平均動作不可能時間

動作不可能な時間の平均値。これは修復のための休止時間のほかに待時間なども含まれる。

6) アベイラビリティ (Availability)

修理可能なものに対し用い、特定の瞬間に機能を維持している確率を示す。

アベイラビリティ (A) は次式でもとめる。

$$A = \frac{\text{MUT}}{\text{MUT} + \text{MDT}} \quad (8.13)$$

または

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \quad (8.14)$$

で求めることも多い。

設備の稼働率を高めるにはアベイラビリティを高くする必要があり、これには

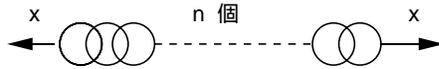


図 8.2: チェーンモデル

- MTBF または MUT を高める．すなわち構成要素の信頼性を高めるか，システムの冗長化等を行う．
- MTTR または MDT を低める．すなわち故障発見および修復時間を短縮する．

のいずれか，あるいは両者の併用により実現できる．

例題 8.1

ある電子装置の MTBF が 10,000 時間，MTTR が 30 時間のとき，以下に答えよ．

- 1) 故障率
- 2) アベイラビリティ
- 3) アベイラビリティを 0.999 以上にするには MTTR をいくら以下にせねばならないか．

[解答]

$$1) \lambda = \frac{1}{\text{MTBF}} = 1 \times 10^{-4} (1 / \text{時間})$$

$$2) A = 10,000 / (10,000 + 30) = 0.997$$

3) $\text{MTTR} = (\frac{1}{A} - 1)\text{MTBF}$ であるので， $A \geq 0.999$ のためには

$$\text{MTTR} \leq (\frac{1}{0.999} - 1) \times 10,000 = 10.0 (\text{時間})$$

故障モデル

8.2.1 故障のモデルとワイブル分布

ワイブル分布 (Weibull distribution) はワイブル (W. Weibull) が金属材料の疲労寿命の研究に使用したものであり，1つの数式で各種の分布を表すことができ便利である．ここではこの考えに沿った故障モデルとワイブル分布について説明する．

まず，故障のモデルとして図 8.2 に示すチェーンモデルを考える． n 個のチェーンにストレスとして力 x を加えた時，もし 1 個でもチェーンが切れると全体が切れチェーンの用をなさない．これはいずれかが故障すると全体が故障するという故障の実際に対応している．このモデルを基にワイブルはストレスとして時間を考え，つぎの累積故障分布関数 $F(t)$ の式を導いた．

$$F(t) = 1 - \exp\left\{-\left(\frac{t - \gamma}{\eta}\right)^m\right\} \quad (8.15)$$

図 8.3: 故障率曲線

これがワイブル分布関数である．ここで

m : 形状母数

η : 尺度母数

と言われる．

ワイブル分布の性質について簡単に述べる．

故障密度関数 $f(t)$ および故障率 $\lambda(t)$ は

$$f(t) = \frac{dF(t)}{dt} = \frac{m}{\eta} \left(\frac{t-\gamma}{\eta}\right)^{m-1} \exp\left\{-\left(\frac{t-\gamma}{\eta}\right)^m\right\} \quad (8.16)$$

$$\lambda(t) = \frac{f(t)}{1-F(t)} = \frac{m}{\eta} \left(\frac{t-\gamma}{\eta}\right)^{m-1} \quad (8.17)$$

ここで

$$\eta = 1, \quad \gamma = 0$$

とすると故障率 $\lambda(t)$ は次式となる．

$$\lambda(t) = mt^{m-1} \quad (8.18)$$

$\lambda(t)$ の形状母数 m による変化の一例を図 8.3 に示す．

このように m の値により種々の故障パターンを表すことができ、つぎのように分類できる．

$m < 1$ 故障率減少形

$m = 1$ 故障率一定形

$m > 1$ 故障率増加形

図 8.4: ワイブル確率紙

つぎに尺度母数の意味につき補足する．式 (8.15) にて $t - \gamma = \eta$ とすると

$$F(\eta) = 1 - e^{-1} = 0.63$$

すなわち 63% が故障し，37% が生き残る時間を示していることになる．

つぎに実データに基づき形状母数および尺度母数の推定法について述べる．式 (8.15) において $\gamma = 0$ と置くと

$$1 - F(t) = \exp\left\{-\left(\frac{t}{\eta}\right)^m\right\}$$

したがって，この逆数の 2 回の自然対数を取ると

$$\ln \ln \frac{1}{1 - F(T)} = m \ln t - m \ln \eta$$

となる．このように横軸を $\ln t$ ，縦軸に $\ln \ln \frac{1}{1 - F(T)}$ で表わしたグラフをワイブル確率紙と呼んでいる．図 8.4 にこれを示す [20]．この用紙の上に実際の故障データより求めた t と $F(t)$ を記して行き，図 8.5 に示すように近似直線を引くと，直線の勾配より形状母数 m が求められる．また縦軸 ($\ln t = 0$) との交点より $m \ln \eta$ が得られるので，先に求めた m を代入して η が求められる．

なお， $F(t)$ の求め方はつぎのようにすればよい．例えば，試験した部品数が最初に N 個あり， t 時間後までに $i(t)$ 個が故障したとすると $F(t) = \frac{i(t)}{N}$ とする．

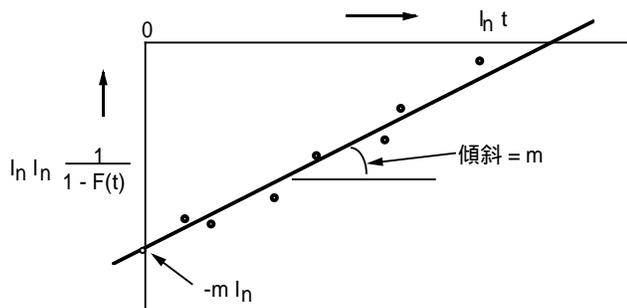


図 8.5: ワイブル分布の母数の推定法

8.2.2 その他の分布

ワイブル分布のほかにつぎの分布もあり目的に応じて用いられている。

1. 二項分布：不良率（または故障の確率） p ，良品率 $q = 1 - p$ をもつ大きなロットからランダムに抽出したサンプルの不良率（故障の確率）として定式化される分布。
2. ポアソン（Poisson）分布：二項分布において不良個数の期待値を一定にして $p \rightarrow 0$ にした極限の分布であり， p の小さな二項分布はポアソン分布で近似できる。
3. 正規分布：平均値 μ ，標準偏差 σ を母数とし故障率がつぎの式で表される分布。

$$f(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \quad (8.19)$$

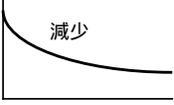
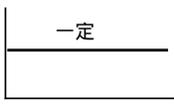
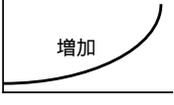
4. 指数分布：故障率一定の分布であり，ワイブル分布において形状母数1の特別な場合に相当する。

8.2.3 故障パターンと対策

故障の発生状況はその対象により変わってくる。軸受の場合と電子部品とでは当然，様子が異なるが大局的にみると，故障の表れ方，すなわち故障率の時間に対する変化が，ワイブル分布にて述べたように，つぎの3種のパターンに分類できる。

1. 減少形 DFR（Decreasing Failure Rate）：時間とともに故障率が減少するタイプで，電子部品などの初期故障がこれに相当する。
2. 一定形 CFR（Constant Failure Rate）：故障率が一定のタイプであり偶発故障とも呼ばれている。電子部品などの故障の多くはこのタイプを示している。

表 8.1: 故障パターンとその方策

パターン	故障率 ($f(t)$)	ワイブル分布の形状母数	有効な手段
減少形 D F R		$m < 1$ 初期故障	製造上の改善 (デバッグなど) 調整, 試運転の改善
一定形 C F R		$m = 1$ 偶発故障	事後保全 コンディション モニタリング
増加形 I F R		$m > 1$ 磨耗故障	定期的な取換え コンディション モニタリング

3. 増加形 I F R (Increasing Failure Rate): 時間とともに故障率が増加を示すもので磨耗故障とも言われ, ベアリングなどでその寿命が近づきつつある状態で現れる.

これらの故障に対する方策はそれぞれのパターンに応じたものを用いる必要がある. その状況を表 8.1 に示している. 表中に示すように設備診断技術による故障の防止は I F R タイプのものに最も有効であり, D F R タイプではデバッグなど製造上の対策の方が効果的である.

普通, 殆どのものはこれら D F R, C F R, I F R をそのライフサイクル中に図 8.6 のように示す. この曲線はその形の西洋風呂との類似からバスタブ曲線 (bath tub curve) と言われている.

システムの信頼度

8.3.1 直列系, 並列系の信頼度

システムの信頼度はシステムを構成する要素の信頼度とその要素の組合せ方とにより定まる. ここではその代表として直列系と並列系の両者についてその信頼度を考える.

直列系

まず直列系では図 8.7 に示すように, 各要素が直列に接続されているので, すべての要素の機能が正常であって, はじめて全体システムは正常に機能する. 従って n 個の要素よりなるシステムを考え, 各々の信頼度を $R_i(t), i = 1, \dots, n$ とすると, 全体システムの信頼度

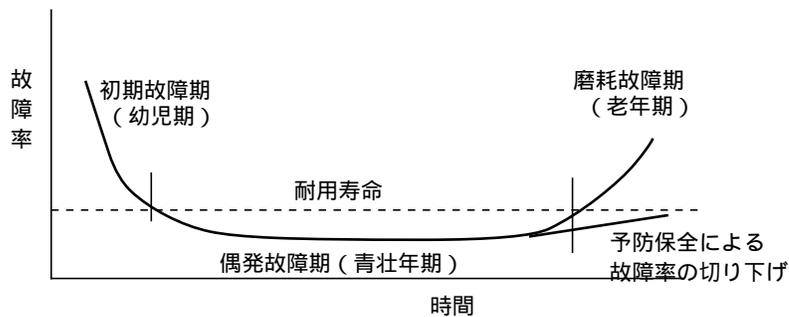


図 8.6: バスタブ曲線

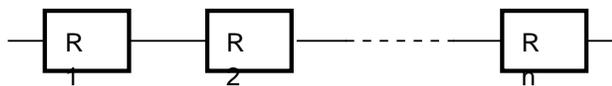


図 8.7: 直列系

$R_s(t)$ は次式で与えられる .

$$R_s(t) = R_1(t) \cdot R_2(t) \cdots R_n(t) = \prod_{i=1}^n R_i(t) \quad (8.20)$$

したがって , 故障率で表すと ,

$$R_s(t) = \exp\left(-\sum_{i=1}^n \int_0^t \lambda_i(t) dt\right) \quad (8.21)$$

となる . さらに故障率が一定の指数分布の故障の場合は

$$R_s(t) = \exp\left(-\sum_{i=1}^n \lambda_i t\right) \quad (8.22)$$

となり , 全体システムの故障率はつぎのとおりになる .

$$\lambda_s = \sum_{i=1}^n \lambda_i \quad (8.23)$$

また MTBF (または MTTF) で表わすとつぎのようになる .

$$\frac{1}{\text{MTBF}} = \sum_{i=1}^n \frac{1}{\text{MTBF}_i} \quad (8.24)$$

並列系

一方 , 並列系 (図 8.8) の場合は並列に接続されている全ての要素が故障して初めて全体

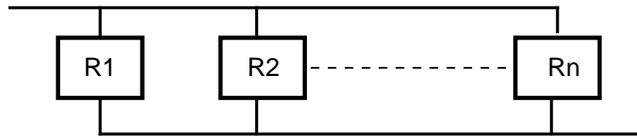


図 8.8: 並列系

システムが故障となる．従って各要素の不信頼度関数を $F(t), i = 1, \dots, n$ とすると，全体システムの不信頼度関数 $F_p(t)$ は

$$F_p(t) = F_1(t) \cdot F_2(t) \cdots F_n(t) = \prod_{i=1}^n F_i(t) \quad (8.25)$$

したがって，信頼度に直すと

$$R_p(t) = 1 - F_p(t) = 1 - \prod_{i=1}^n (1 - R_i(t)) \quad (8.26)$$

となる．

ここで特に 2 個の並列系の場合を考えると [20]

$$R_p(t) = 1 - (1 - R_1(t))(1 - R_2(t)) = R_1(t) + R_2(t) - R_1(t)R_2(t) \quad (8.27)$$

両辺を微分すると，式 (8.4) の関係より

$$\begin{aligned} f_p(t) &= f_1(t) + f_2(t) - f_1(t)R_2(t) - R_1(t)f_2(t) \\ &= F_2(t)f_1(t) + F_1(t)f_2(t) \end{aligned} \quad (8.28)$$

ここで $F_*(t), f_*(t)$ はそれぞれ累積故障分布関数，故障密度関数 ($f_*(t) = dF_*(t)/dt$) を表す．ここでいずれも故障率が一定の指数分布の故障とすると，

$$F_1(t) = 1 - R_1(t) = 1 - \exp(-\lambda_1 t)$$

$$F_2(t) = 1 - R_2(t) = 1 - \exp(-\lambda_2 t)$$

であるので，これを上式に代入すると， $MTBF_p$ (または $MTTF_p$) はつぎのようになる．

$$\begin{aligned} MTBF_p &= \int_0^{\infty} t(F_2(t)f_1(t) + F_1(t)f_2(t))dt \\ &= \frac{1}{\lambda_1} + \frac{1}{\lambda_2} - \frac{1}{\lambda_1 + \lambda_2} \\ &= MTBF_1 + MTBF_2 - \frac{MTBF_1 * MTBF_2}{MTBF_1 + MTBF_2} \end{aligned} \quad (8.29)$$

特に $\lambda_1 = \lambda_2 = \lambda$ の時は並列化により次のように $MTBF$ が 50% 増加することが分かる．

$$\text{MTBF}_p = \frac{3}{2\lambda} = \frac{3}{2}\text{MTBF} \quad (8.30)$$

以上ではMTBFについて述べたが、MTTFについても同様に成り立つ。

例題 8.2

ある電子装置があり、4個のパワートランジスタ、10個のダイオード、5個のコンデンサおよび20個の抵抗により構成されている。各 부품のMTTFがパワートランジスタが 10^5 時間、ダイオードが 10^6 時間、コンデンサが 10^5 時間、抵抗が 10^6 時間であるとしたとき、この電子装置のMTTFはいくらになるか。

[解答] この電子装置ではいずれの部品が故障しても使えないので部品は直列系を構成していることになる。したがって、式(8.24)を適用して

$$\frac{1}{\text{MTTF}} = \frac{4}{10^5} + \frac{10}{10^6} + \frac{5}{10^5} + \frac{20}{10^6} = \frac{12}{10^5}$$

すなわち

$$\text{MTTF} = 8,333(\text{時間})$$

となる。

例題 8.3

MTBFが10,000時間の電子装置2台を並列に使用した場合にMTBFはいくらになるか。

[解答]

$$\text{MTBF}_p = 2 \times 10,000 - \frac{10,000 \times 10,000}{2 \times 10,000} = 15,000$$

8.3.2 マルコフモデルによる冗長系の解析

冗長系のような状態変化を伴うシステムの信頼性の解析はマルコフモデルによる解析が有効である。今、対象とするシステムが単純なマルコフ過程に従うとする。図8.9に示すマルコフモデルを考えると、つぎの関係が成立する。

$$P_j(t + \Delta t) = p_{ij}P_i(t) + p_{jj}P_j(t) \quad (8.31)$$

ここで S_i, S_j は状態を示し、 $P_i(t), P_j(t + \Delta t)$ はそれぞれ時刻 t に状態 S_i 、次の時刻 $t + \Delta t$ に S_j にある確率を示している。 p_{ij} は S_i から S_j に移る確率を示し遷移確率という。したがって p_{jj} は S_j の状態に留まる遷移確率を示している。

いま、一般的に $n + 1$ 個の状態 S_0, \dots, S_n が時刻 t とつぎの時刻 $t + \Delta t$ で取りうる確率をそれぞれ $P_0(t), \dots, P_n(t)$ および $P_0(t + \Delta t), \dots, P_n(t + \Delta t)$ とすれば、つぎの式で表せる。

$$\begin{bmatrix} P_0(t + \Delta t) \\ \vdots \\ P_n(t + \Delta t) \end{bmatrix} = \begin{bmatrix} p_{00} & \cdots & p_{0n} \\ \vdots & & \vdots \\ p_{n0} & \cdots & p_{nn} \end{bmatrix}^T \cdot \begin{bmatrix} P_0(t) \\ \vdots \\ P_n(t) \end{bmatrix} \quad (8.32)$$

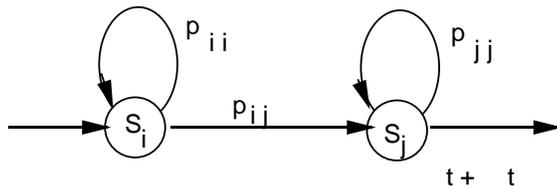


図 8.9: 状態遷移グラフ

ここで行列

$$p = \begin{bmatrix} p_{00} & \cdots & p_{0n} \\ \vdots & & \vdots \\ p_{n0} & \cdots & p_{nn} \end{bmatrix} \quad (8.33)$$

を遷移行列 (transition matrix) という . ここで

$$\sum_{j=0}^n p_{ij} = 1, i = 0, \dots, n \quad (8.34)$$

が成立する .

遷移確率 p_{ij} は故障率一定 λ (指数分布) の確率過程では

$$p_{ij} \approx \lambda \Delta t \quad (8.35)$$

と表すことができる . 何故なら

$$\frac{f(t)}{R(t)} = \lambda$$

従って

$$R(t) = \exp(-\lambda t), f(t) = \lambda \exp(-\lambda t) \quad (8.36)$$

$$p_{ij}(t + \Delta t | t) = \frac{1}{R(t)} \int_t^{t+\Delta t} f(t) dt = 1 - \exp(-\lambda \Delta t) \approx \lambda \Delta t \quad (8.37)$$

冗長系の解析ではこのマルコフモデルによる解析が有効である . その一例をつぎに示す . 冗長系にはつぎの 2 種があるので , それぞれについて解析してみる . なお , ここでは 2 台の冗長系について考える .

1. 並列冗長系 : 常時 , 2 台が使用状態にあり , いずれが故障してもそのまま運転を継続し , その間に故障した 1 台を修理する . 修理が終わると , 再び並列運転に入れる方式である . この方式の場合 , 2 台とも故障しない限り , 運転は中断なく続けられるため , システムに対して影響を与えない .

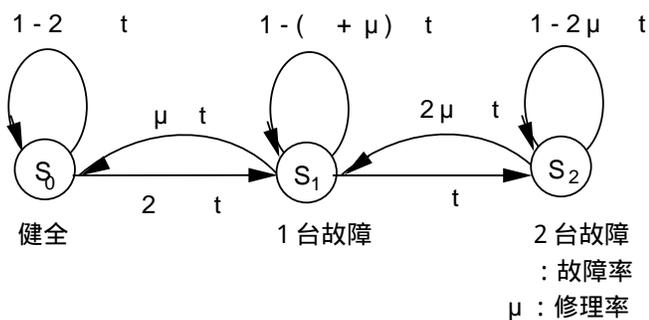


図 8.10: 並列冗長系の状態遷移グラフ

2. 待機冗長系：常時は1台のみを使用し，他は待機状態にある．使用中のものが故障すると，他の待機中のものに切り替えて，再び運転を続ける．この方式の場合，故障時に待機中のものと切り替えるため，一時的に運転を中断せねばならない．また，待機中のものは時々，状態が正常であることの確認が必要となるが，使用状態にないため，劣化は起こらない利点がある．

1) 並列冗長系

ここでは2台の並列冗長系のアベイラビリティを求める．図8.10が以下で解析する冗長系の状態遷移グラフである．ここで S_0 , S_1 , S_2 はそれぞれ健全，1台故障，2台故障の状態を表している． λ , μ はそれぞれ故障率，修理率を表す．

S_0 , S_1 , S_2 の状態にある確率を $P_0(t)$, $P_1(t)$, $P_2(t)$ とし，かつ

$$P(t) = [P_0(t), P_1(t), P_2(t)]^T$$

とすると

$$P(t + \Delta t) = \begin{bmatrix} (1 - 2\lambda\Delta t) & \mu\Delta t & 0 \\ 2\lambda\Delta t & (1 - (\lambda + \mu)\Delta t) & 2\mu\Delta t \\ 0 & \lambda\Delta t & (1 - 2\mu\Delta t) \end{bmatrix} P(t) \quad (8.38)$$

従って

$$\frac{dP}{dt} = \lim_{\Delta t \rightarrow 0} \frac{P(t + \Delta t) - P(t)}{\Delta t} = p^T P(t) \quad (8.39)$$

ただし

$$p^T = \begin{bmatrix} -2\lambda & \mu & 0 \\ 2\lambda & -(\lambda + \mu) & 2\mu \\ 0 & \lambda & -2\mu \end{bmatrix} \quad (8.40)$$

$P(t)$ の初期条件を $P(0)$ として上式のラプラス変換を行うと

$$sP^*(s) - P(0) = p^T P^*(s)$$

したがって,

$$P^*(s) = (sI - p^T)^{-1}P(0) \quad (8.41)$$

また $P(t)$, $P^*(s)$ の各要素を

$$P(t) = [P_0(t), P_1(t), P_2(t)]^T$$

$$P^*(s) = [P_0^*(s), P_1^*(s), P_2^*(s)]^T$$

と表わす.

ここでアベイラビリティを求める. 初期条件を $P(0) = (1, 0, 0)^T$ とし, 1台故障までは正常に動作するので, アベイラビリティはつぎのようになる.

$$A(t) = P_0(t) + P_1(t) \quad (8.42)$$

したがって, アベイラビリティのラプラス変換は

$$A^*(s) = P_0^*(s) + P_1^*(s) \quad (8.43)$$

$t \rightarrow \infty$ でのアベイラビリティの定常値はラプラス変換の最終値定理

$$A = \lim_{s \rightarrow 0} sA^*(s)$$

を用いて

$$A = 1 - \frac{\lambda^2}{(\mu + \lambda)^2} \quad (8.44)$$

また, MTBFとMTTRを用いてアベイラビリティを表現すると

$$A = 1 - \frac{\text{MTTR}^2}{(\text{MTBF} + \text{MTTR})^2} \quad (8.45)$$

2) 待機冗長系

並列冗長系の場合と同様に2台の待機冗長系のアベイラビリティを求める. 図8.11にその状態遷移グラフを示す. S_0, S_1, S_3 はそれぞれ健全(1台使用中, 他の1台は待機中), 1台故障(他の1台は修理中), 2台とも故障の各状態を表している. λ, μ は前と同じく故障率, 修理率を表す. 並列冗長系との違いは健全状態から1台故障状態への遷移確率が半分(1台分)になっていることである. したがって, 遷移確率はつぎのようになる.

$$p^T = \begin{bmatrix} -\lambda & \mu & 0 \\ \lambda & -(\lambda + \mu) & 2\mu \\ 0 & \lambda & -2\mu \end{bmatrix} \quad (8.46)$$

表 8.2: F T A と F M E A / C A

F T A	F M E A / C A
システム → 部品	部品 → システム
Top Down (結果から原因を)	Bottom Up (原因から結果を)
分析的	総合的
論理図・相互関係が分かる	表による解析・相互関係がやや不明確
多重故障の分析も可能	単一故障の上位への波及
安全性の解析	信頼性の解析

$$A_1 = 1 - \frac{10^2}{(1,000 + 10)^2} = 0.999901$$

待機冗長系では

$$A_2 = 1 - \frac{10^2}{(1,000 + 10)^2 + 1,000^2} = 0.999950$$

システム信頼性解析手法

信頼性、安全性の解析手法として代表的なものに F M E A / C A (Failure Mode Effect Analysis / Criticality Analysis, 故障モード影響解析) と F T A (Fault Tree Analysis, 事故樹解析) があり、いずれも広く用いられ信頼性、安全性の向上に大きく貢献している。F M E A / C A と F T A は表 8.2 に示すように互いに逆の関係にある。F T A は個々の要素から見た全体への影響の解析であり、主として設計段階の信頼性解析に用いられている。これに反し F M E A / C A はシステム全体の事故に着目し、これに関わる要因を分析するもので、安全性の解析である。両者の出発点は異なるが目的とするところは同じである。F T A では要素間の関係を解析し、論理的であることから安全面で多く用いられている。F T A のようなグラフに基づく解析法としてはこのほか次のものも用いられている。

1. H A (Hazard Analysis) : 方法は F T A とほぼ同じであるが、F T A の事故の代わりに ハザードすなわち人間に与える傷害、死亡に重点を置いている点異なる。
2. E T A (Event Tree Analysis) : 事故では初期の軽微な事故が起因となり、つぎつぎに発展して重大な事故に発展する場合が殆どである。それを定量的に分析するため発展段階の各事象(成功か失敗かなど)を図 8.12a) に示す木グラフ表現で表して事故発生確率、損失期待値などを求める解析法である。

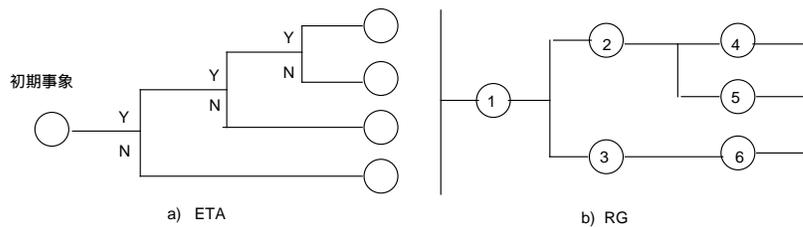


図 8.12: E T AとR Gのグラフ表現

3. R G (Reliability Graph) : F T (事故樹) が事故に着目したグラフ表現であるのに対し R Gは健全側に着目したもので互いに双対の関係にある (図 8.12b) .

8.4.1 故障モード影響解析 (F M E A / C A)

F M E A / C Aは F M E Aと F M E C Aの意であり、両者の違いはつぎの通りであり、普通は両者をつにして F M E A / C Aまたは F M E A / F M E C Aと呼ばれることが多い。いずれもシステムを構成する個々の要素の故障のシステム全体への影響を分析したものである。

- F M E A : システムの設計評価のための解析手法で、システムに潜在する故障の影響を理論的に解明し、信頼性向上のための指針を与える系統的な手法であり、一定の様式に基づき解析がなされる。
- F M E C A : 各構成要素の事故確率を考え事故の重大性の定量的な解析を一定の公式に基づいて行う手法である。

1) F M E Aの手順

つぎの順序で解析を進める。

1. システムを定義する。解析するシステムを規定する。構成要素からなる論理ブロック線図が有効である。例えば図 8.13 に示すようなブロック線図を作り、構成要素とそれらの相互関係を示して置くとよい。
2. F M E Aを行う基礎となる条件を明確にする。
 - システムの目的 (1 次 , 2 次 , …)
 - 環境や運転などからもたらされる制約
 - システム要素の故障の中身

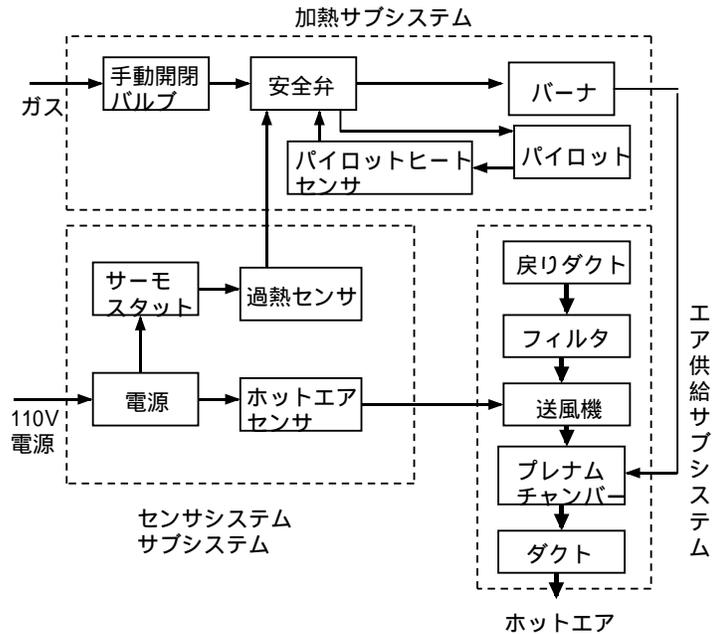


図 8.13: 加熱システムのブロック線図

3. システムを構成するハードウェアや機能などの説明を行う。

ブロック線図と記述法との併用による。ブロック図では直列・並列関係，冗長関係を明示する。

4. 各要素故障のモードとその影響を分析し，図 8.14 に示すような一定の様式の表により示す。

表の項目としては例えば次のものがある。これらすべてを入れる必要はなく，対象のシステムや装置の性格など考えて取捨選択すればよい。

- 部品番号：部品の識別のための番号，ブロックダイアグラムとの対応番号など。
- 部品名（または要素名）：ブロックダイアグラムに記載した名称。
- 機能：その部品の働きを記す。
- 故障モード：故障開，故障閉，破壊，亀裂，汚損，漏れ，接触不良，動作不良，など。
- 故障原因
- 故障の影響：部分的な影響と全体システムへの影響とに分けて記すのがよい。影響の重大性なども記す。

部品番号	部品名	機能	故障モード	故障原因	故障の影響		故障検出法	対策
					部分的	最終システム		
SS-2	ホットエアセンサ	空気温度の検出	動作不良	回路の故障	空気温度制御が不能となる	空気温度の異常上昇	空気温度の検出	取り替え
SS-3	過熱センサ	燃焼ガス温度の検出	低温で動作	センサ部分の故障	ガスの燃焼がすぐ止まる	空気の温度が上がらない	空気の温度とガスの燃焼状況	取り替え

図 8.14: 加熱システムの FMEA

- 故障検出法：どのような方法があるかを記す。
- 対策：故障した場合の対策を記す。
- 補償手段：故障時のバックアップなど補償手段が講じられているか否かを記す。
- 運転時期：連続運転か断続運転かなどを含め運転方法，使用方法を記す。これにより対策が違ってくる。
- 修復時間：故障時の修復時間の推定値を記す。

こうして作成した表は上長が承認する手続きを明確にして置く。このように関係者が文書により確認するのが大切である。

2) FMECA

FMEAにCA (Criticality Analysis) を加えたもので、各部品ごとにつぎの解析を行ない、先に示したFMEAの表の最後に下記の欄を追加して記載する。

1. 故障発生確率：環境条件，使用状態でのストレス，運転時間などを考慮して発生確率を計算する。
2. 致命度ランク：故障発生確率，システムあるいは人間に与える影響の程度を考慮して，4ランク（軽微，非致命的，致命的，壊滅的）位に区分けした番号で示す。

8.4.2 事故樹解析 (FTA)

FTA はシステムとして望ましくない事象すなわち故障を定義し，これを出発点に，故障を起こす原因をトップダウン的にたどることにより原因を明らかにするとともに安全性を定性的並びに定量的に評価する技法である。対象としてはプラント，システム，または装置など種々あるが一般に大規模のものに適用される場合が多く，したがって，FTAの実施の際は関係の技術者がチームを編成したり，会議などにより検討するなどが行なわれる。その手順としては図8.15に示す順序で行なわれる。

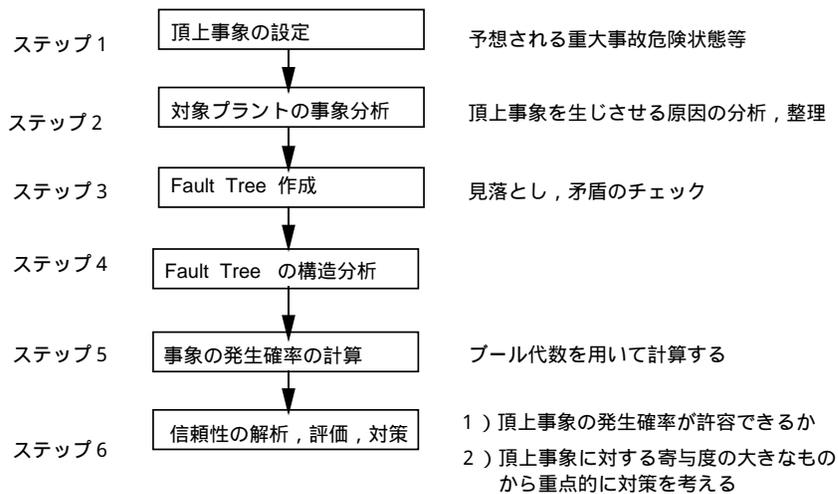


図 8.15: F T A の手順

まず，ステップ 1 では頂上事象を設定する．頂上事象とは FTA を行なう対象のプラント，システム，装置などの故障の中で最も重大なものを指し，通常はシステム全体の故障を選ぶ．

ステップ 2 では頂上事象から出発し，それぞれの原因になる事象を順番に挙げて，かつ事象相互の関係すなわち AND の関係，OR の関係など論理的な関係も明らかにして行く．この際，どこまで細分して行くかも決めて置く．このように分析を行なう最小の単位となる事象を基本事象と呼んでいる．

ステップ 3 ではこうして集めた各事象の因果関係をグラフの木で表わす．これをフォールトツリー (FT, Fault Tree) と呼んでいる．その作成には図 8.16 に示す記号を用いる．こうして作られたフォールトツリーの簡単な一例を図 8.17 に示す．こうして作られたフォールトツリーは関係者間で検討し，見落とし，矛盾などが無いかなどの多面的な調査を行ない，精度の高いものにする努力が必要である．

ステップ 4 ではこうして作られたフォールトツリーに対して構造的な分析を行なう．例えば，後で述べるカットセットやパスセットを求めて定性的な安全性を調べたり，二重系などのバックアップシステムや手動システムの要否などの検討も行なう．

ステップ 5 では定量的な分析を行なう．基本事象の故障データすなわち故障確率のデータを集め，それと各事象間の因果関係や論理関係に基づいて，ブール代数を利用して順次上位の事象の発生確率を求め，頂上事象の確率に到達するまで計算を進める．

ステップ 6 ではこうして得られたフォールトツリーや各事象の発生確率を見ながら総合的な検討を行なう．例えば

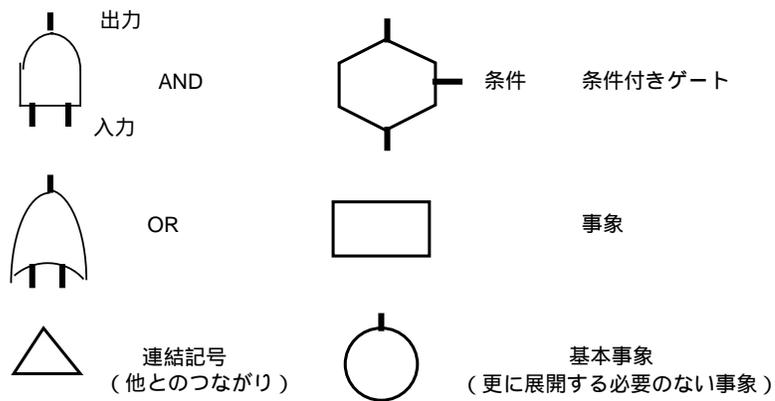


図 8.16: フォールトツリーに使用する記号

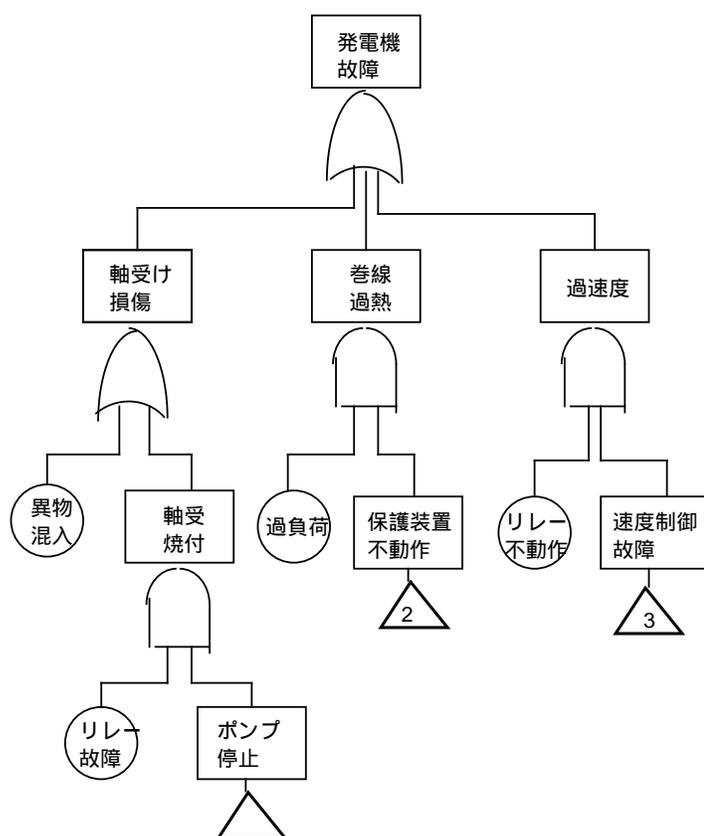


図 8.17: フォールトツリーの一例

1. 頂上事象（すなわち全体の故障）の発生確率が許容できるか．許容できれば終了であるが，許容できない場合は以下に述べる方法で検討を行なう．
2. 頂上事象の発生確率に寄与する程度の高い事象から重点的に対策を考える．対策としては，基本事象については故障確率の小さいものすなわち信頼性の高いものに変えるか，複数使用して並列冗長システムを構成して信頼性を高めるなどを行なう．また中間事象については信頼性を高める方策としては論理関係を利用して，新しく要素を追加したり，冗長系にしたりする．あるいはシステムの構成を基本的に変えてしまうなども行なう．

以上に述べたように F T A による解析手順の主なものを整理すると，次のとおりである．

1. フォールトツリーの作成（作図）
2. 最小カットセット，パスセットの算定
3. 故障発生確率，MTBF，MTTF，MTTR，アベイラビリティの計算

8.4.3 最小カットセットと最小パスセット

F T A において重要な役目をするこの二つについて説明する．

1. 最小カットセットの定義

- 安全が損なわれる側から見た場合，そこを切断すればつながらなくなる，ぎりぎりの切断の集合．
- F T A においては「この集合に含まれる全ての基本事象が生起することによって頂上事象を発生させる（すなわち故障する）に必要にして十分な基本事象の最小の集合」を示す．

2. 最小パスセットの定義

- システムが無故障であるために最小限つながっていなければならない道の組．
- F T A においては「この集合に含まれる基本事象が生起しなければ頂上事象が発生しない（すなわち故障しない）ことを保証する基本事象の最小の集合」を示す．

図8.18により一例を述べるとつぎのとおり．

最小カットセット： $(F_1)(F_2, F_3)$

最小パスセット： $(F_1, F_2)(F_1, F_3)$

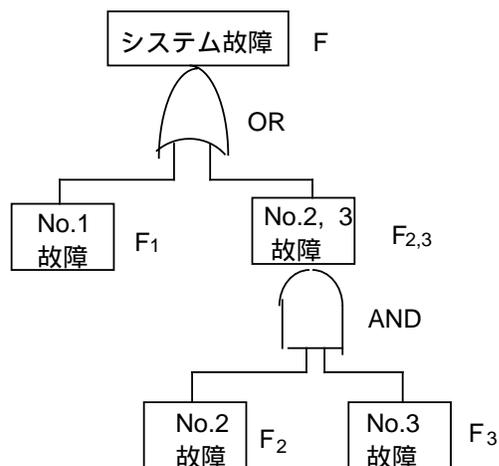


図 8.18: 最小パスセット, 最小パスセットの一例

システムの安全という観点からは以下のことが望ましい。

1. 最小カットセットについては
 - セットの数が少ないこと。
 - セット内の要素が多いこと。
2. 最小パスセットについては
 - セットの数が多きこと。
 - セット内の要素が少ないこと。

8.4.4 システム信頼度の計算

F T Aにて最小カットセット, 最小パスセットが得られると, いずれかを用いてシステム信頼度, 故障確率(頂上事象の)がつぎのとおり計算できる。

1) 最小カットセット法

システム信頼度 R および故障確率 F は次式により計算できる。

$$\begin{aligned}
 F &= Pr\left\{\bigcup_{i=1}^n C_i\right\} \\
 &= \sum_{i=1}^n Pr\{C_i\} - \sum_{i=1}^n \sum_{j>i}^n Pr\{C_i \cap C_j\} + \cdots + (-1)^{n-1} Pr\left\{\bigcap_{i=1}^n C_i\right\} \quad (8.49)
 \end{aligned}$$

ここで

n = 最小カットセットの数

C_i = i 番目の最小カットセット

$Pr\{C_i \cap C_j\}$ = C_i, C_j の同時発生確率

基本事象が互いに独立であるとする

$$F = \sum_{i=1}^n \prod_{l \in C_i} f_l - \sum_{i=1}^n \sum_{j>i}^n \prod_{l \in C_i \cup C_j} f_l + \cdots + (-1)^{n-1} \prod_{l \in \cup_{i=1}^n C_i} f_l \quad (8.50)$$

$$R = 1 - F \quad (8.51)$$

ここで

\prod = 積記号

f_l = 基本事象の故障確率

2) 最小パスセット法

$$\begin{aligned} R &= Pr\left\{\bigcup_{i=1}^n T_i\right\} \\ &= \sum_{i=1}^m Pr\{T_i\} - \sum_{i=1}^m \sum_{i>j}^m Pr\{T_i \cap T_j\} + \cdots + (-1)^m Pr\left\{\bigcap_{i=1}^m T_i\right\} \\ &= \sum_{i=1}^m \prod_{l \in T_i} r_l - \sum_{i=1}^m \sum_{j>i}^m \prod_{l \in T_i \cup T_j} r_l + \cdots + (-1)^{m-1} \prod_{l \in \cup_{i=1}^m T_i} r_l \end{aligned} \quad (8.52)$$

ここで

m = 最小パスセットの数

$T = i$ 番目の最小パスセット

$Pr\{T_i \cap T_j\}$ = T_i, T_j の同時発生確率

何れの方法を使用するかは両セットの数により決まるが、各々の数を NOC , NOP とすると $NOC < (NOP)^2$ の時は最小カットセットを用いた方がよいとされている。

3) 近似法

上記厳密式によりすべてを計算するのは計算時間を要することから、近似法が用いられている。例えば

$F_k = F$ の式の第 k 項までを採用

とすると

$$F \leq F_1 = \sum_{i=1}^n Pr\{C_i\} \quad (8.53)$$

$$F \geq F_2 = \sum_{i=1}^n Pr\{C_i - \sum \sum Pr\{C_i \cap C_j\}\} \quad (8.54)$$

⋮

したがって、誤差 $\Delta F = |F - F_k|$ が所定の値以下になった時点で止めるか、あるいは、あらかじめ k を定めて近似する方法がある。

例題 8.6

図8.18に示すシステムについて、それぞれの基本事象の故障確率を f_1, f_2, f_3 とした時の頂上事象の故障確率を求めよ。

[解答] 最小カットセット法と最小パスセット法との二つの方法で求めてみる。

1. 最小カットセット法：最小カットセットは $(F_1)(F_2, F_3)$ の2つであるので、故障確率はつぎのとおりになる。ここで F_1, F_2 はそれぞれ第1近似値、第2近似値を示す。

$$F_1 = f_1 + f_2 f_3$$

$$F_2 = f_1 + f_2 f_3 - f_1 f_2 f_3$$

2. 最小パスセット法：最小パスセットは $(F_1, F_2)(F_1, F_3)$ であり、各基本事象の信頼度はそれぞれ $r_1 = 1 - f_1, r_2 = 1 - f_2, r_3 = 1 - f_3$ であるので、頂上事象の信頼度および故障確率はつぎのようになる。

$$\begin{aligned} R &= r_1 r_2 + r_1 r_3 - r_1 r_2 r_3 \\ &= (1 - f_1)(1 - f_2) + (1 - f_1)(1 - f_3) - (1 - f_1)(1 - f_2)(1 - f_3) \\ &= 1 - (f_1 + f_2 f_3 - f_1 f_2 f_3) \end{aligned}$$

$$F = 1 - R$$

上記より分かるように最小カットセットより求める方が簡単である。

8.4.5 解析プログラム

F T Aによるシステム信頼性の解析には多量の計算を必要とするため、各種の計算機プログラムが開発されており、最近の大規模システムの計画では欠かせないものになっている。これらプログラムの基本的な考えは図8.19に示すようにシステム構造をフォールト

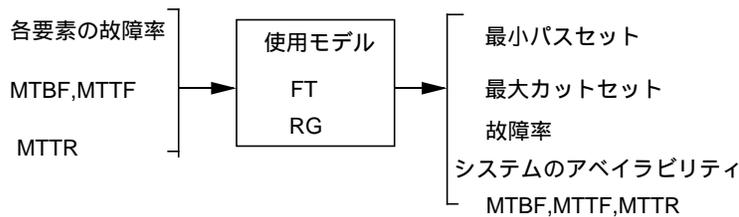


図 8.19: 信頼性解析プログラム

ツリ (F T) または信頼性グラフ (R G) など で表し , 各要素の故障率などのデータを入れてシステム信頼性を表す数値を出力するようになっている .

このような信頼性解析には大別して定性的解析と定量的解析がある . 前者は主としてシステムの構造解析であり , 後者は構造並びに個々の要素の信頼度データを基にした故障率などの数値解析である . これら二者により総合的に信頼性を評価し , 対策を検討する . 図 8.20 に解析の手順を示す .

1. 定性的解析 : システム信頼性に関する構造解析である . 従来から行われている影響モード解析 (F M E A) , あるいは各要素間の論理構造を基にして信頼性の総合表示としてのフォールトツリあるいは信頼性グラフを作成し , 最小カットセットあるいは最小パスセットを計算することが中心になる .
2. 定量的解析 : 前述の定性的解析の結果と要素の信頼度データを基にしてシステム全体の M T B F , M T T R , アベイラビリティ , 故障率などを求める解析である , これらの値がシステムの目標値と合致すれば解析は完了であるが , 合致しない場合には次に述べる方法により解析をさらに進めることになる .
 - (a) システム再検討 : システムの構造を見直し , 例えば冗長系 (二重系 , バックアップシステム) の導入あるいはその強化を行う .
 - (b) : 要素の高信頼化 : 要素の信頼度をさらに高いものにする (スクリーニングの強化など)

このような検討結果を再び入力データとして与え , 再計算し総合的評価を行うという経過を経て満足した値が得られるまで解析を続ける .

解析プログラムはすでに多数作られている . 海外で公表されているもののいくつかの例を表 8.3 に示す .

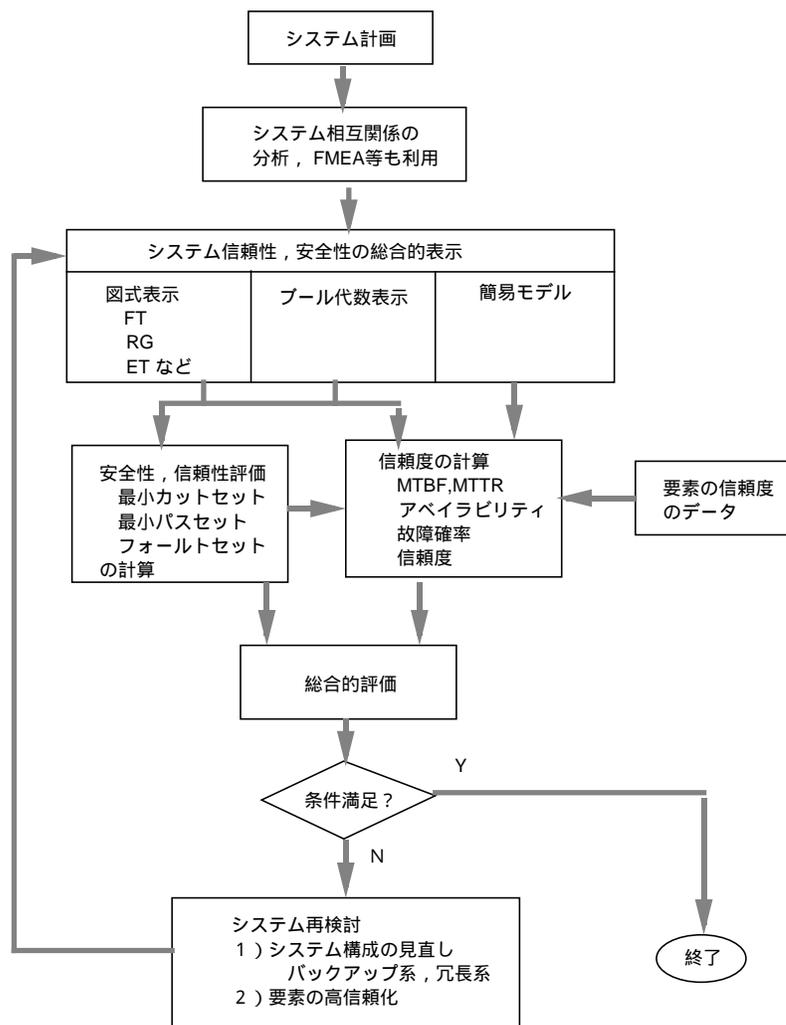


図 8.20: システム信頼性解析ルーチン

8.4.6 解析時に考慮すべき事項

ここではシステム信頼性の解析の際に特に注意すべき事項について述べる。

1) 共通原因故障

電源の故障, 地震など環境に関する事故原因では, その影響が複数の要素に共通に及ぶものが多い。これらを共通原因故障 (common mode failure) と呼び, 解析時に特別の扱いを要する。これらにより発生する基本事象間では独立性が保たれないので, 独立でない

表 8.3: 信頼性解析プログラムの例

No.	名称	作成者	出力	入力	備考
1	RGAYB	I.R.Batts M.Yamazaki (1976)	最小カットセット 最小パスセット システム信頼度	信頼度グラフ 要素の信頼度	RGベース
2	RELICS	G.D.M.Person (1977)	最小カットセット 最小パスセット アベイラビリティ MTTF, MTTR 重要度	信頼度グラフ 要素のMTTF, MTTR	RGベース
3	FTAB	D.B.Brown (1976)	発生確率 コスト 効果	フォールトツリー 要素の発生確率 コスト	FTベース
4	MOCUS	J.B.Fussel et al.	最小カットセット 最小パスセット 発生確率	フォールトツリー 要素の発生確率	FTベース
5	WAMCUT	F.L.Leverenz H.Kirch (1978)	最小カットセット アンアベイラビリティ	フォールトツリー	電力関係
6	GO Methodology	R.L.Williams W.V.Gately (1978)	フォールトセット 発生確率	GO Chart	電力関係

注： FT: フォールトツリー，RG:信頼度グラフ

ものとして数式化せねばならない。また冗長システムは共通原因故障に対しては有効に機能しないので、特に注意が必要である。

2) 無修理系対修理系

電子部品のように故障すれば廃棄する無修理系と一般の機械のように修理して使用する修理系とでは評価指標が変わってくる。無修理系では故障率やMTTFなどを用い、修理系では故障率、MTBFのほかアベイラビリティなどを使用する。

3) 誤動作故障と不動作故障

通常の機器については、正常か故障だけを考えればよいが、センサーや保護システムの信頼性の解析では故障を不動作故障と誤動作故障とに分けて扱う必要がある。不動作故障は通常の故障に相当するが、誤動作故障は他からの指令無しに誤信号を伝える故障であり、両者の与える影響が異なる。このため不動作故障と誤動作故障の両方のフォールトツリーを作成し、各々の解析からシステム故障に対する影響を比較検討せねばならない。

4) 冗長系

信頼性向上策は要素の信頼性向上のほかに、システム構成の面で配慮し、例えば冗長システムを採用するなど大切である。冗長システムには各種あり、その選定に当たっては無修理系であるか修理系であるか、さらに誤動作と不動作の確率などを考えて、より信頼度の高い方式を選択すべきである。一例として代表的な冗長系である”3 or”系と”2 out of 3”系について修理系の場合のアベイラビリティで比較すると、不動作故障のみを考えると前者の”3 or”系の方が高いが、誤動作故障を考えると誤動作故障と不動作故障の確率の比で変わり、多くの場合後者の”2 out of 3”系の方が高い[5]。

5) 自動監視保全の効果

システム信頼度向上の今一つの方策として、常時監視または自動点検によりアベイラビリティを上げる方法がある。例えば、制御装置のプリント基板に故障表示装置を装備させれば、故障時に直ち発見できるので修理時間を短縮されてアベイラビリティが改善される。

演習問題

- 8.1 故障率が 5.0×10^{-6} 時間⁻¹ の素子がある。MTTF は幾らか。
- 8.2 MTBF が 5,000 時間の電子装置がある。アベイラビリティを 0.999 以上にするには MTTR をいくら以上にせねばならないか。
- 8.3 信頼度 R の装置を 2 台並列に使用した場合の信頼度はいくらになるか。
- 8.4 故障率が 1.0×10^{-5} 時間⁻¹ の装置 1 台を単独および 2 台並列に使用した場合の各々に対して、1 年間および 2 年間の動作時間に対する信頼度を求めよ。
- 8.5 図 8.21 に示すフォールトツリーの最小カットセットを求め、頂上事象の故障率を求める式を誘導せよ。
- 8.6 単車の故障について F T A と F M E A を行え。

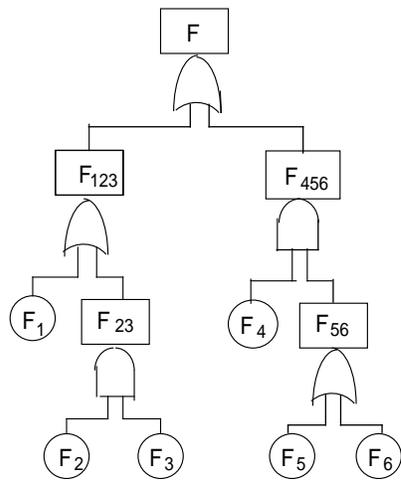


図 8.21: フォールトツリー

参考文献

- [1] 浅居喜代治ほか：現代システム工学の基礎，オーム社，1979
- [2] 麻生英樹：ニューラルネットワーク情報処理，産業図書，1988
- [3] 市川悖信：多目的決定の理論と方法，計測自動制御学会，1980
- [4] 井上紘一ほか：FTAの基礎理論，安全工学，Vol.17,No.3,150-157, 1978
- [5] 小野俊彦ほか：システム信頼性解析技法，東芝レビュー，Vol.37,No.7,601-604,1982
- [6] 小野俊彦：棒材の切断における最適組合せへのニューラルネットワークの応用，電気学会論文誌D, 113巻, 12号, 1371-1377, 1993
- [7] 川喜田二郎：発想法，中央公論社，1967
- [8] 北野宏明編：遺伝的アルゴリズム，産業図書，1993
- [9] 北野宏明編著：遺伝的アルゴリズム2，産業図書，1995
- [10] 小玉陽一：BADSIICによるシステム・ダイナミックス，共立出版，1980
- [11] 近藤次郎：システム工学，改訂増補，丸善，1981
- [12] 榎木義一ほか：システム工学の基礎，日新出版,1971
- [13] 塩見弘：信頼性工学入門，丸善，1972
- [14] 菅野文友：信頼性工学入門，コロナ社，1980
- [15] 寺野寿郎：システム工学入門，共立出版，1985
- [16] 中村嘉平ほか：システム工学通論，朝倉書店，1987
- [17] 鍋島一郎：動的計画法，森北出版，1968
- [18] 奈良宏一，佐藤泰司：システム工学の数理手法，コロナ社，1996

- [19] 成田成之助：システム工学の手法，コロナ社，1970
- [20] 野中保雄，島岡惇：冗長系，日科技連，1990
- [21] 馬場ほか：数理計画法入門，共立出版，1989
- [22] 原田耕介ほか：信頼性工学，養賢堂，1977
- [23] 松永省吾：システム工学入門，東京電機大学出版局，1987
- [24] 松原正一：プロセスシステム工学，朝倉書店，1970
- [25] 三浦武雄ほか：現代システム工学概論，オーム社，1977
- [26] 村山雄二郎，寺野寿郎：あいまい論理を用いた異常診断，システムと制御，24，11，719-725 (1980)
- [27] メドウズ，D.H.，メドウズ，D.L：成長の限界，ダイヤモンド社，1972
- [28] 森雅夫ほか：オペレーションズリサーチ，朝倉書店，1991
- [29] Bertsekas, Dimitri P. : Dynamic Programming: Deterministic and Stochastic models, Prentice-Hall, 1987
- [30] Dasgupta, D. and Michalewicz, Z.(Eds.) : Evolutionary Algorithms in Engineering Applications, Springer, 1997
- [31] Henley, E.J. et al. : Riliability Engineering and Risk Assessment, Prentice-Hall, 1981
- [32] Rosses, C. and Anton, H. : Applications of Linear Algebra, John Willey & Sons, 1977

演習問題略解

1章

- 1.1 略
- 1.2 略
- 1.3 略
- 1.4 略

2章

2.1 略

2.2 有効グラフを図9.1に示す．また A および $T = A + I$ は

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

可到達行列 R を求めると $T^2 = T^3 = R$ となり，

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3 $x(0) = [0, 1, 0, 0, 0]$ において各時刻での各位置にいる確率を求めると表9.1のとおりになる．この表と本文の表との比較により，初期状態により落ちつく状態が変わってくる事が分かる．これは溝に落ちると戻れないとの非可逆な部分があるからである．

2.4

- 1) よろめき歩きの状態遷移図を図9.2に示す．
- 2) 遷移確率はつぎのとおり，

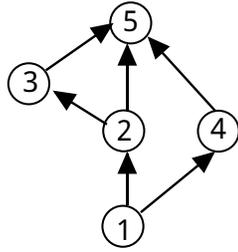


図 9.1: 有向グラフ

表 9.1: ランダウオークの経過

時間-k (分)	$x_1(k)$	$x_2(k)$	$x_3(k)$	$x_4(k)$	$x_5(k)$
0	0.0	1.0	0.0	0.0	0.0
1	0.5	0.2	0.3	0.0	0.0
2	0.6	0.19	0.12	0.09	0.0
3	0.695	0.098	0.126	0.054	0.027
4	0.744	0.0826	0.0816	0.0486	0.0432
5	0.785	0.0573	0.0654	0.0342	0.0578
10	0.874	0.0137	0.0150	0.0082	0.0895
20	0.899	0.0007	0.0008	0.0004	0.0987

$$p = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.2 & 0.4 & 0.4 \\ 0 & 0.6 & 0.4 \end{bmatrix}$$

3) 初期状態を $x(0) = [0, 1, 0]$ として, 1, 2, 3, 10 分後に各位置にいる確率を求めると表 9.2 のとおりになる. なお, 参考までに定常的にどの状態に落ちつくかを見てみる. 定常状態を $x^* = [x_1^*, x_2^*, x_3^*]$ とすると, つぎの関係が成り立つ.

$$[x_1^*, x_2^*, x_3^*] \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.2 & 0.4 & 0.4 \\ 0 & 0.6 & 0.4 \end{bmatrix} = [x_1^*, x_2^*, x_3^*]$$

さらにいずれかの位置にいるので,

$$x_1^* + x_2^* + x_3^* = 1$$

この一次式を解くと定常値としてつぎが得られる.

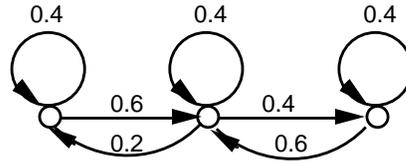


図 9.2: よろめきの状態遷移グラフ

表 9.2: ランダウオークの経過

時間-k (分)	$x_1(k)$	$x_2(k)$	$x_3(k)$
0	0.0	1.0	0.0
1	0.2	0.4	0.4
2	0.16	0.52	0.32
3	0.168	0.496	0.336
10	0.167	0.500	0.333

$$x^* = [0.167, 0.5, 0.333]$$

なお，前の問題では非可逆な部分があったので，この方法は適用できなかった．

3 章

- 3.1 略
- 3.2 略
- 3.3 略
- 3.4 略

4 章

4.1

- 1) 図的解法を図 9.3 に示す．
- 2) まず標準形式に変換して

$$\text{制約条件式 } 4x_1 + x_2 + x_3 = 20$$

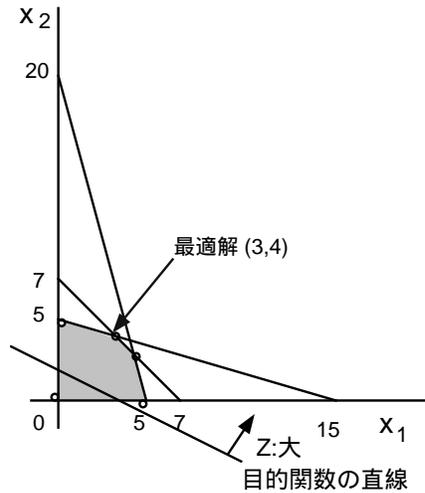


図 9.3: 制約条件の範囲と最適解

$$x_1 + 3x_2 + x_4 = 15$$

$$x_1 + x_2 + x_5 = 7$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

$$\text{目的関数} \quad -7x_1 - 12x_2 + 0 \times x_3 + 0 \times x_4 + 0 \times x_5 + z = 0$$

これを基に表 9.3 に示すように、初期テーブルを作りピボット操作を行なうと解が得られる。以上より

$$x_1 = 3, \quad x_2 = 4, \quad z = 69$$

5 章

5.1 本文の例を参照

6 章

6.1 まず勾配を求めると

$$\nabla f(x) = \text{grad}f(x) = [2(x_1 - 5), 4(x_2 - 4)]^T$$

1) 最適勾配法

a) 第 1 ステップ 最初は原点 $x_0 = [0, 0]^T$ より出発して最適点の最初の推定値 $x^1 = [x_1^1, x_2^1]^T$

表 9.3: シンプレックステーブル

サイクル	基底	x_1	x_2	x_3	x_4	x_5	z	b	b/a
1	x_3	4	1	1	0	0	0	20	$20/1 = 20$
	x_4	1	3	0	1	0	0	15	$15/3 = 5$ \Leftarrow
	x_5	1	1	0	0	1	0	7	$7/1 = 7$
	z	-7	-12	0	0	0	1	0	
			↑						
2	x_3	11/3	0	1	-1/3	0	0	15	$15 \times 3/11 = 4.1$
	x_2	1/3	1	0	1/3	0	0	5	$5 \times 3 = 15$
	x_5	2/3	0	0	-1/3	1	0	2	$2 \times 3/2 = 3$ \Leftarrow
	z	-3	0	0	4	0	1	60	
			↑						
3	x_3	0	0	1	3/2	4.5	0	4	
	x_2	0	1	0	5/6	-0.5	0	4	
	x_1	1	0	0	-1/2	3/2	0	3	
	z	0	0	0	5/2	4.5	1	69	

を求める .

$$x_0 = [0, 0]^T$$

$$\nabla f(x) \equiv \text{grad} f_0(x) = [-10, -16]^T$$

$$\begin{aligned} \max_{\Delta t} f(x(t + \Delta t)) &= \max_{\Delta t} f(-\Delta t[-10 - 16]^T) \\ &= \max_{\Delta t} [(10\Delta t - 5)^2 + 2(16\Delta t - 4)^2] \end{aligned}$$

上式の括弧の中の式を Δt で微分して零と置いて , Δt および x^1 を求める .

$$\Delta t_1 = 0.291$$

$$x_1^1 = 10 * 0.291 = 2.91, \quad x_2^1 = 16 * 0.291 = 4.65$$

b) 第 2 ステップ つぎに $x_1 = [2.91, 4.65]^T$ より出発してつぎの最適点の推定値を求める .

$$\nabla f(x) = \text{grad} f_1(x) = [2(2.91 - 5), 4(4.65 - 4)]^T = [-4.18, 2.60]^T$$

$$\begin{aligned}\max_{\Delta t} f(x(t + \Delta t)) &= \max_{\Delta t} f([2.91, 4.65]^T - \Delta t[-4.18, 2.6]^T) \\ &= \max_{\Delta t} [(4.18\Delta t - 2.09)^2 + 2(2.6\Delta t - 0.65)^2]\end{aligned}$$

括弧の中を微分して零と置き Δt_2 および x を求めると,

$$\Delta t_2 = 0.391$$

$$x_2^1 = 2.91 + 4.18 \times 0.391 = 4.54, \quad x_2^2 = 4.65 - 2.60 \times 0.391 = 3.64$$

まだ最小値 (5, 4) ではないので, この方法を繰り返すことになる.

1) 共役勾配法

a) 第 1 ステップ 最適勾配法と同じなので, そのまま流用する.

b) 第 2 ステップ まず係数 β_1 を計算する. それに使用するデータはすでに最適勾配法において求めているので, それを流用する.

$$\beta_1 = \frac{|\nabla f(x_1)|^2}{|\nabla f(x_0)|^2} = \frac{4.18^2 + 2.6^2}{10^2 + 16^2} = 0.0681$$

$$p_1 = [-4.18, 2.6]^T + [-10, -16]^T \times 0.0681 = [-4.86, 1.511]^T$$

$$\max_{\Delta t} f(x(t + \Delta t)) = \max_{\Delta t} [(4.86\Delta t - 2.09)^2 + 2(1.511\Delta t - 0.65)^2]$$

括弧の中を微分して零と置くことにより,

$$\Delta t_2 = 0.430$$

$$x_2^1 = 2.91 + 4.86 \times 0.430 = 4.99, \quad x_2^2 = 4.65 - 1.511 \times 0.430 = 4.00$$

となり, 最適値が得られた.

以上のように共役勾配法の方が収束性がよい. この状態は図 9.4 に示す.

7 章

7.1 略

8 章

$$8.1 \quad \text{MTTF} = 1/\lambda = 1/5.0 \times 10^{-6} = 2.0 \times 10^5$$

$$8.2 \quad A = 5,000/(5,000 + \text{MTTR}) \geq 0.99 \text{ であるから } \text{MTTR} \leq 5 \text{ (時間)}$$

$$8.3 \quad R_p = 1 - (1 - R)^2 = 2R - R^2$$

8.4

[単独の場合] $R(t) = \exp(-\lambda t)$ 1 年間は $t = 24 \times 365$ (時間) であるので, 1 年間および 2 年間の信頼度 $R(1), R(2)$ はつぎのとおりになる.

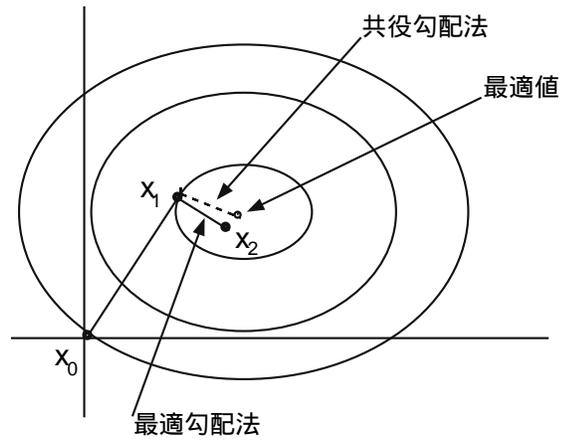


図 9.4: 最適解への接近の状況

$$R(1) = \exp(-10^{-5} \times 24 \times 365) = \exp(-0.0876) = 0.916$$

$$R(2) = \exp(-0.0876 \times 2) = 0.839$$

[2 台並列の場合] $R_p(t) = 2R(t) - R^2(t)$ であるので ,

$$R_p(1) = 2 \times 0.916 - 0.916^2 = 0.993$$

$$R_p(2) = 2 \times 0.839 - 0.839^2 = 0.974$$

8.5 最小カットセットはつぎのとおり

$$(F_1)(F_2, F_3)(F_4, F_5)(F_4, F_6)$$

また基本事象の発生確率はそれぞれの確率を f_* で表わすと

$$\begin{aligned} f &= f_1 + f_2 f_3 + f_4 f_5 + f_4 f_6 \\ &\quad - (f_1 f_2 f_3 + f_1 f_4 f_5 + f_1 f_4 f_6 + f_2 f_3 f_4 f_5 + f_2 f_3 f_4 f_6 + f_4 f_5 f_6) \\ &\quad + (f_1 f_2 f_3 f_4 f_5 + f_1 f_4 f_5 f_6 + f_2 f_3 f_4 f_5 f_6) \\ &\quad - f_1 f_2 f_3 f_4 f_5 f_6 \end{aligned}$$

8.6 略

索引

[アルファベット]

CFR, 107

CPM, 99

DFR, 107

DP, 65

DYNAMO, 25

ETA, 116

FMEA, 116, 117

FMEA/CA, 116, 117

FMECA, 116, 117

FTA, 116, 119

GA, 86

HA, 116

IFR, 108

ISM, 37

LP, 46

MDT, 103

MTBF, 102

MTTF, 103

MTTR, 103

MUT, 103

NN, 82

PERT, 93

RG, 117

[あ]

ISM手法, 37

アベイラビリティ, 103

アローダイヤグラム, 93

[い]

一対比較, 36

遺伝的アルゴリズム, 86

[か]

階層構造, 3

確定的システム, 4

確率的システム, 4, 16

片方向連結, 21

カットセット, 122

可到達, 21

可到達行列, 22, 37

ガントチャート, 97

[き]

基底解, 51

基底変数, 51

基本許容解, 51

基本事象, 120

共役勾配法, 80

共役方向, 80

共通原因故障, 127

強連結, 21

強連結グラフ, 22

[く]

グラフモデル, 20

クリティカルパス, 97

クリティカルパス法, 99

[け]

形状母数, 105

KJ法, 33

[こ]

交叉, 87

勾配法, 79

故障診断, 28

故障のモデル, 104

故障パターン, 105

故障密度関数, 101

故障モード影響解析, 116, 117

故障率, 102

誤動作故障, 128

[さ]

最急降下法, 80

最小パスセット, 122

最早開始時刻, 96

最早結合点時刻, 94

最早終了時刻, 96

最大傾斜法, 79

最短経路問題, 67

最遅結合点時刻, 95

最遅終了時刻, 96

最適勾配法, 80

最適性の原理, 65

最小カットセット, 122

差分方程式, 16

[し]

シグナルフロー線図, 17

資源配分問題, 69

事故樹解析, 116, 119

指数分布, 107

システム, 1

システム工学, 5

システム信頼度, 123

システムダイナミックス, 25

システムの構造, 2

システムの実現手段, 6

システムの分類, 3

自動監視保全, 129

シナリオライティング, 34

時不変システム, 4

時変システム, 4

シミュレーション, 29

シミュレーテッド・アニーリング, 86

尺度母数, 105

弱連結, 21

周波数応答モデル, 12

自由余裕時間, 96

修理系, 128

修理率, 103

瞬間故障率, 102

状態推移行列, 14

状態遷移グラフ, 23

状態変数モデル, 13

状態方程式, 13

冗長系, 111, 129

人為変数, 58

シンプルGA, 86

シンプレックス法, 47

信頼性, 101

信頼度関数, 102

[す]

- スラック変数, 48
- [せ]
- 正規分布, 107
- 静的システム, 4
- 制約条件付き最適化, 90
- セールスマン巡回問題, 84
- 遷移確率, 23, 111
- 遷移行列, 24, 112
- 線形計画法, 46
- 線形システム, 4
- 選好曲線, 40
- 選択, 87
- 全余裕時間, 96
- [そ]
- 倉庫運営問題, 72
- 相互結合形 NN, 82
- 双対問題, 63
- [た]
- ダイグラフ, 21
- 多段決定問題, 65
- タブロー法, 55
- 多目的最適化, 88
- 単純マルコフ, 23
- [ち]
- チェーンモデル, 104
- 頂上事象, 120
- 直列系, 108
- [て]
- テクノロジーアセスメント, 41
- デシジョンツリー, 40
- デルファイ法, 36
- テロテクノロジー, 42
- 伝達関数行列, 12
- 伝達関数モデル, 12
- [と]
- 動的計画法, 65
- 動的システム, 4
- 突然変異, 87
- [に]
- 二項分布, 107
- 2段階法, 57
- 入出力モデル, 14
- ニュートン法, 82
- ニューラルネットワーク, 82
- [ぬ]
- ネットワークエネルギー, 83
- [の]
- ノルム, 90
- [は]
- バスセット, 122
- バスタブ曲線, 108
- パルス伝達関数, 16
- パレート最適解, 88
- [ひ]
- 非基底変数, 51
- 非線形システム, 4
- 微分方程式モデル, 12
- 費用勾配, 99
- [ふ]
- ファジー集合, 27
- ファジーモデル, 27
- フォールトツリー, 120
- 不信頼度関数, 101

不動作故障, 128

部分システム, 3

ブレーストーミング, 33

フローダイヤグラム, 25

ブロック線図, 17

フローモデル, 25

分散構造, 3

[へ]

平均故障間隔, 102

平均修復時間, 103

平均動作可能時間, 103

平均動作不可能時間, 103

並列系, 108

[ほ]

ポアソン分布, 107

ホップフィールドネットワーク, 82

[ま]

マトリックス法, 38

マルコフ過程, 23

マルコフモデル, 23, 111

[む]

無修理系, 128

[め]

メンバーシップ関数, 27

[も]

モデリング, 11

[や]

山くずし, 97

山積み図, 97

山登り法, 79

[ゆ]

有向グラフ, 21

[ら]

ライフサイクルコスト, 42

ラグランジュ乗数法, 91

ランダムウォーク, 23

[り]

離散系, 15

離散事象システム, 4

離散システム, 4

隣接行列, 21

[る]

累積故障分布関数, 101

ルーレット選択方式, 87

[れ]

連続システム, 4

[ろ]

ローカルミニマム, 86

ロジスティックス, 42

[わ]

ワイブル分布, 104

システム工学

平成10年4月1日 初版発行
著者 小野俊彦
発行所 福岡工業大学 情報工学部
情報工学科 小野研究室
発売所 紀伊国屋書店福工大ブックセンター
印刷所 コロニー印刷