

Optimum Repair Planning of Existing Bridges by Genetic Algorithm Based on Inspection Data

メタデータ	言語: jpn 出版者: 公開日: 2019-02-14 キーワード (Ja): キーワード (En): 作成者: Chikata, Yasuo メールアドレス: 所属:
URL	https://doi.org/10.24517/00053204

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 International License.



点検データに基づく遺伝的アルゴリズム を援用した既存橋梁の最適補修計画立案

(課題番号 09650521)

平成9年度～平成11年度科学研究補助金
基盤研究 (C) 研究成果報告書

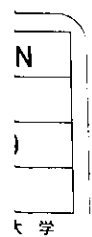
平成12年3月

金沢大学附属図書館



8000-89069-0

研究代表者 近田 康夫
(金沢大学大学院自然科学研究科 助教授)



KAKEN
1999
78

点検データに基づく遺伝的アルゴリズム
を援用した既存橋梁の最適補修計画立案

(課題番号 09650521)

平成9年度～平成11年度 科学研究補助金
基盤研究(C) 研究成果報告書

平成12年3月

研究代表者 近田康夫
(金沢大学大学院自然科学研究科 助教授)

著 者 寄贈

はしがき

本研究は、既存橋梁の補修計画立案を支援する場合を想定し、既存の橋梁点検データに基づいて補修対象部位の組合せを選択する、ナップサック問題と見なしたアプローチを試みたものである。

これまで、多くの研究が橋梁点検台帳に基づいて行ってきた、橋梁損傷の要因分析あるいは因果関係の解明といった後ろ向きの研究から、点検台帳のデータに基づいて補修計画を立案支援する前向きの研究へと方向転換を図った。

本研究は、

- 組合せ最適化問題として定式化した、橋梁補修代替案探索問題を遺伝的アルゴリズムを用いて解くこと
- 橋梁点検台帳自体をデジタル化(パーソナルコンピュータ上でのデータベース化)することにより、補修代替案作成支援プログラムとの連携を図りやすくすること。さらには、データベース管理システムをGIS(地理情報システム)と連動させて、作成された補修代替案のより広い視野(視点)からの検討を可能にすること。

の2つの命題から構成されている。

本研究を遂行するにあたり、金沢大学大学院自然科学研究科環境基盤工学専攻 西雄一君、谷口健君、廣瀬智士君、中山善隆君をはじめとする学生諸君の献身的な協力があつたことを付記し、深甚の謝意を表します。

研究組織

研究代表者：近田 康夫(金沢大学自然科学研究科(工学部兼務) 助教授)

研究経費

平成9年度	1,600 千円
平成10年度	1,000 千円
平成11年度	300 千円
合計	2,900 千円

研究発表

(1) 学会誌など

1. 近田康夫, 廣瀬智士, 城戸隆良: SOM/LVQ による橋梁点検データの見直し, 構造工学論文集, Vol.44A, pp.593-600, 1998.3.
2. 近田康夫, 福山貴久, GA による補修項目の関連を考慮した橋梁補修計画, 構造物の診断に関するシンポジウム論文集 (土木学会), pp.59-64, 1998.7.
3. 近田康夫, 辻岡信也, 城戸隆良: NN による橋梁損傷度判定の精度向上に関する研究, 鋼構造工学年次論文報告集 (日本鋼構造協会), Vol.6, pp.451-456, 1998.11.
4. 近田康夫, 清水宏隆, 城戸隆良: ネットワーク信頼性を指標とした GA 援用橋梁補修計画土木情報処理システム論文集 (土木学会), Vol.8, pp.199-206, 1999.10.

(2) 口頭発表

1. Yasuo CHIKATA, Takayoshi KIDO : Application of Hyper-Text Expression to Bridge Damage Disgnosis, 7th Int.Conf. on Computing in Civil and Building Eng., Voll. p.565-570, 1997.8.(Seul).
2. 辻岡信也, 近田康夫: 自己組織化特徴マップによる橋梁点検データの検証, 土木学会年次学術講演会講演概要集, I-A, pp.390-391, 1997.
3. 西雄一, 近田康夫, 城戸隆良: GA を援用した橋梁補修計画のための橋梁評価方法に関する研究, 土木学会第 53 回年次学術講演会講演概要集, I-A264, pp.528-529, 1998.10.6.
4. 谷口健, 近田康夫, 城戸隆良: 橋梁データ管理システムの構築に関する研究, 土木学会第 53 回年次学術講演会講演概要集, I-A268, pp.536-537, 1998.10.6.
5. 清水宏孝, 近田康夫, 城戸隆良: ウィルス進化型 GA を援用した橋梁補修計画に関する研究, 土木学会第 54 回年次学術講演会講演概要集, I-A121, pp.242-243, 1999.9.24.
6. Yasuo CHIKATA and Satoshi HIROSE : Examination of Bridge Inspection Data by SOM/LVQ, Proc. of First China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems(CJK-OSM 1), Oct.31-Nov.4, 1999(Xi'an China).

目次

第1章	はじめに	1
1.1	研究の目的と概要	1
第2章	GAを援用した点検データに基づく橋梁補修計画立案支援	3
2.1	橋梁点検台帳とデータベース化	3
2.2	橋梁の評価	4
2.2.1	ニューラルネットワークの感度解析による分析	6
2.2.2	線形判別式	9
2.3	補修費用	10
2.3.1	橋梁各部位の補修費用	10
2.3.2	補修足場の費用	13
2.4	橋梁補修計画案の作成	16
2.4.1	補修部位選択へのナップサック問題の適用	16
2.4.2	スケジューリング	18
2.5	遺伝的アルゴリズムの概念	20
2.6	遺伝的アルゴリズムの処理過程	20
2.6.1	コーディングと初期世代の生成	23
2.6.2	適応度	24
2.6.3	淘汰	24
2.6.4	交叉	26
2.6.5	突然変異	27
2.6.6	収束条件	27
2.7	遺伝的アルゴリズムの特徴と弱点	28
2.8	補修部位の選択	28
2.8.1	GAへの適用	28
2.8.2	投資効率を考慮した関数による解析	32
2.8.3	関数の無次元化	40
2.8.4	2種類の関数の比較	43

2.8.5	足場の考慮	47
2.9	スケジューリング	55
2.9.1	GA への適用	55
2.9.2	スケジューリング問題の適用	57
2.9.3	解析結果の比較	60
2.10	結論	67
	参考文献	72
	第3章 橋梁データ管理支援	74
3.1	管理支援システムの提案	74
3.1.1	概念	74
3.1.2	システム設計	74
3.1.3	目的の明確化	75
3.1.4	環境の予測	77
3.1.5	現状の把握	78
3.1.6	データベースの運用	79
3.1.7	維持管理システム構築の目的	80
3.1.8	既存の構築例	81
3.2	橋梁データ管理システム	83
3.2.1	システムの概要	83
3.2.2	システムの説明	85
3.3	ネットワーク環境を考慮したシステム	95
3.3.1	本システムの目的	95
3.3.2	システムの構成	96
3.3.3	IIS について	97
3.3.4	システムの説明	100
3.3.5	考察	103
3.4	GIS との連携	104
3.4.1	導入	104
3.4.2	本システムとの連携	104
3.4.3	地図上でのデータ表現	107
3.4.4	地図上でのデータ管理	110
3.4.5	考察	113
3.5	結論	114
3.5.1	本システムの有効性	114

3.5.2	今後の課題	114
参考文献		116
第4章	橋梁管理データベースと補修計画立案支援の連携	117
4.1	リンクアプリケーションの作成	117
4.1.1	アプリケーションの設計	117
4.1.2	プログラムの概要	120
4.2	橋梁補修計画作成システム	130
4.2.1	処理の流れ	130
4.2.2	実行処理例	138
4.3	結論	142
参考文献		143
第5章	橋梁点検データの見直し	145
5.1	ニューラルネットワーク	145
5.1.1	ニューラルネットワークの概要	145
5.1.2	SOM(Self-Organizing Map)	150
5.1.3	LVQ(Learning Vector Quantization)	154
5.2	入力データの洗練化	158
5.2.1	概要	158
5.2.2	鋼橋	159
5.2.3	コンクリート橋	170
5.2.4	考察	181
参考文献		185

第1章 はじめに

戦後の経済成長期から本格的に始まった全国的な道路交通網の整備事業の実施に伴い、多くの橋梁が建設された。これらの橋梁の中には、橋梁が設計・架設された時期には予測し得なかった交通量の急激な増加、および車両の大型化などの環境の変化により、著しい損傷を受けている例も少なくない。

それら損傷・老朽化が進行した橋梁は、事故や交通遮断の原因となり多大な社会的・経済的影響を与えることは容易に理解できる。そこで、新規橋梁の建設と同等あるいはそれ以上に既存橋梁の維持管理対策が重要課題として認識されることになる。

橋梁の維持管理を人間の健康管理と相似させて捉え、定期点検による損傷の早期発見・早期補修による維持管理が有効であると考えられている。その考え方に沿って、建設省をはじめとして、公団や地方自治体でも橋梁の定期点検データの蓄積とデータベース化が進んでおり、蓄積されたデータ量は少なくない。しかし、そのデータは、損傷要因の分析や、橋梁点検・診断を行った専門家の判定論理を逆解析することに利用されることが多く、補修計画へ反映する試みはほとんど見られなかった。

点検データはいわゆる専門家の外観調査によるもので、特に損傷の著しいものは架け替えや大規模補修・補強が行われるが、これらは別枠の予算要求を経て実施されることが多い。しかし、中程度以下の損傷に関しては、多数の橋梁を管理する場合、限られた予算内でどの橋梁のどの部位を優先的に補修すべきかという問いに答えられる有効な手段は、その組み合わせ数の膨大さ故に未だ構築されていない。

本研究では、自治体や公団のように多くの橋梁を管理するときに、その管理のためのプラットフォーム(作業環境)と補修計画立案への意志決定支援を目的としている。

1.1 研究の目的と概要

橋梁維持管理に使われる予算には上限があるため全ての要補修箇所を補修できるわけではない。その場合、「どの橋のどの部位を補修すれば良いか」といった、投資効率の良い、あるいは費用対効果の高い橋梁補修計画が求められる。そこで、データベース管理システムの構築とは別の研究では、限られた予算の中で社会的満足性を最大に満たす補修対象橋梁群と補修部位の組み合わせを決定し補修計画の代替案を作成する橋梁維持管理支援シ

システムの構築を試みる。検討する補修部位の組合せは膨大なものとなるため、遺伝的アルゴリズムを適用した解探索を試み、実用的な付加条件の実装についても検討する。

また、補修計画立案のためのデータを管理するために、データベース管理システムの構築を試み、橋梁台帳、橋梁点検台帳を橋梁データ、材料集計表、橋梁診断、震災点検、ASR 調査、塩害調査、補修歴(上部工・下部工)、の8つのデータ群に分割し、それぞれに対するデータベースファイルを設計する。またこれらのデータベースファイルの操作を行うためにデータベース管理システムの基礎となる閲覧、新規登録、修正、削除ができるようにプログラムを作成する。

これら2つのシステムは、プロトタイプが既に構築されており、成果を上げている。しかし、橋梁補修計画の代替案作成において遺伝的アルゴリズムを適用する際に、必要なデータを橋梁台帳・橋梁点検台帳のデータが格納されているデータベースファイルからデータを探し、入力データとして整える作業が必要となってくる。この作業は、データを探し出すのと書き出すのに多くの手間と時間を必要とする。またデータが更新、変更された時にも同様の作業が必要となってくる。

そこで、橋梁維持管理システムの補修計画の策定を行うために、本研究では、データベースに新たに遺伝的アルゴリズムに用いるデータの項目群を設け、プログラムソフトを用いてデータベースからのデータの供給と、遺伝的アルゴリズムを用いた組み合わせ計算を一貫して行える、リンクアプリケーションの構築を行う。

一方、橋梁維持管理におけるデータ管理と意思決定を支援するもう一つの柱として、データベース化された橋梁データとそれを視覚的に表示、分析するGIS(地理情報システム)との連携も今後の発展が望めるものの一つである。本研究でも構築した橋梁データの維持管理システムとGISソフトウェアとの連携を試み、その可能性を探ることとする。

他方、橋梁点検データベースのデータに基づいて補修代替案や安全性評価の基準が模索されることを考えると、データベース内のデータ自身の信頼性を確保しなければならない。そこで、点検データベース内のデータの見直しを、ニューラルネットワークの1種類であるSOM/LVQを用いて行う。

すなわち、

- データベースマネジメントシステムの構築
- 補修代替案システムの構築
- 両システムのリンクアプリケーションの開発
- データベース内のデータの点検

が本研究の主たる内容となる。これらの内、補修代替案作成が第一義の目的であり、そのための周辺環境整備がその他の内容となっている。

第2章 GAを援用した点検データに基づく 橋梁補修計画立案支援

2.1 橋梁点検台帳とデータベース化

本研究においては、I県が昭和57年度から昭和63年度の期間に行われた、定期点検を基に作成された橋梁点検台帳²⁾の点検データを用いた。

定期点検は、橋面上・橋下から橋梁の細部について点検するものであり、長期点検計画に基づいて所定区域にわたって足場などを用いて、目視・たたき点検などを行う点検である。点検項目は、橋面・地覆・高欄の破損等の有無、床版コンクリートのひび割れの有無、排水施設の状況、塗装の状況、橋の振動、発振音の状況、照明施設の状況、下部構造部の沈下・傾斜・移動、基礎の洗掘、河床の変動などである。点検項目と点検結果の例を表-2.1に示す。

表- 2.1 橋梁点検台帳

橋梁名	橋面舗装	地覆高欄	床版	床組構	主構	支承	伸縮継手	排水装置	塗装	洗掘変動	躯体変動	安定構造	安定材質	耐震性	損傷度評価
A橋	○	○	△	△	△	○	△	△	○	○	○	○	○	○	B
B橋	○	△	○	○	○	○	○	○	-	○	○	△	○	△	D
C橋	○	○	×	×	×	×	×	×	-	△	○	△	○	△	A
・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
Z橋	○	○	○	△	○	○	○	○	○	○	○	○	○	○	C

点検項目は14項目あり点検項目の評価基準として、次の3段階評価が設定されている。

○: ほとんど損傷していない。危険性はない。

△: 損傷している。危険性はやや高い。

×: 損傷が激しい。危険性が高い。

また、各点検項目の具体的な判断基準を表-2.2に示す。各橋梁の損傷度評価は、各点検項目の点検結果に基づいて専門家が次の4段階で行う。

A: 架け替え、あるいは全面的な補強を必要とする。

B: 早急に補修を必要とする損傷がある。

C: 軽微な補修を必要とする損傷がある。

D: 健全であり、目立った損傷は見られない。

本研究ではこの点検データを基に、各点検項目の点検結果を○・△・×の順にそれぞれ1・2・3と数値化し、そして各橋梁の損傷度評価をA・B・C・Dの順にそれぞれ1・2・3・4と数値化しデータベース化したものを、橋梁補修計画案の作成に用いる。

2.2 橋梁の評価

これまで、さまざまな手法を用いて橋梁の評価が行われてきた。まず、数量化理論第Ⅱ類を用いた橋梁の評価³⁾が行われたが、得られたカテゴリースコアは損傷が軽微なほど評価値が増加するとは限らず、補修することにより、橋梁の評価が下がるというような逆転現象が起こる場合があった。次に、ファジー理論の λ 積分⁴⁾・ファジー理論のminmax法⁵⁾を用いた橋梁評価が行われた。これにより、橋梁各部位の重要度係数が求められ線形判別式(2.2)のように設定することにより、数量化理論第Ⅱ類の分析結果に基づく場合の逆転現象が起こらなくなったが、どちらの重要度係数も一定の値に偏っており必ずしも妥当な橋梁評価を行っているとは言い難い結果であった。なお、線形判別式(2.2)は後述する。

そしてニューラルネットワークの感度解析⁶⁾が行われた。この解析によって得られた重要度係数は一定の値に偏ることなく、また専門家による損傷度評価との誤差も少ない結果であった。

本研究においては4手法のうち、最もよい結果の選られたニューラルネットワークの感度解析結果を用いて橋梁の評価を行った。

表- 2.2 各点検項目の判断基準

	要因	評価	判断基準
上	舗装	△	縦方向のひび割れ, あるいは数箇所段差が見られる。
		×	縦方向, 亀甲状のひび割れがある。 あるいは, 実際に走行してみて, 明らかに支障があると思われる。
	地覆高欄	△	地覆に劣化が見られ, ひび割れが発生している。また, 高欄が事故などで一部破損している。
		×	地覆の劣化が激しく, 欠け落ちが見られる。高欄の腐食が激しく, 耐荷力に不安がある。
	床版	△	亀甲状のひび割れがある。何ヶ所も石灰分の遊離が見られる。
		×	ひび割れの間隔が狭く, 欠け落ちが見られる。 あるいは, 施工不良により, 骨材の分離, 鉄筋の露出が激しい。
	床組構・主構	△	コンクリート構造の場合, ひび割れ, 剥離が見られる。 鋼構造の場合, 全体にわたって腐食している。
		×	コンクリート構造の場合, ひび割れに間隔がある。 鋼構造の場合, 腐食が激しく, 耐荷力に不安がある。また, 鋼材がひび割れを起こしている。
	支承	△	錆, 腐食が激しい。あるいは, 変形, 破損が見られる。
		×	上記の症状が助長し, 支承が可動しなくなっている。
	伸縮継手	△	後打材やシール部に欠陥が見られる。あるいは, 目的周辺が陥没している。
		×	盲目地になっている。あるいは, 水漏れが激しく周辺部を腐食している。また, 実際に車で走行してみて, 著しく衝撃・騒音がある。
排水装置	△	土砂の堆積, 排水管のずれなどにより, 排水が不完全である。	
	×	路上に水が溜まる箇所がある。	
塗装	△	部分的にはがれ, 発錆が見られる。	
	×	全体的に劣化し, 発錆が著しい。	
耐震性	△	支承縁端距離が不足している。あるいは, 支承の機能が著しく低下している。	
	×	橋梁が古いため, 移動制限装置が施されていない。耐震構造的に極めて不安定である。	
下部工	洗掘変動	△	洗掘が見られる。
		×	洗掘が進み, 基礎の安定性を欠くと思われる。
	躯体変動	△	躯体の沈下, 移動, 傾斜が見られる。
		×	症状が進み, 上部工にまで影響がみられる。
	安定性(構造)	△	ひび割れ, 鉄筋露出, 腐食による劣化等構造的な破損が見られる。
		×	ひび割れの間隔が大きい。あるいは, 一部が欠けている。
安定性(材質)	△	構造的な破損と合わせ, 極めて不安定である。	
	×	破損, 移動が激しく, 極めて不安定である。	

2.2.1 ニューラルネットワークの感度解析による分析

ニューラルネットワークの概要

ニューラルネットワークは、脳における神経細胞の信号伝達系をモデルとした情報処理法であり、学習によって自らのパターン認識能力を高めることができるのが特徴である。

従来のコンピューターでは、よくデバックされたプログラムが必要であるように、ニューラルネットワークでは十分に吟味された適切な量の学習用データが必要である。今回検討に用いた階層型ニューラルネットワークは、ニューロン(神経細胞)を数学的にモデル化したセルを多数結合させたものである。セルには、入力層と出力層があり、その間にはいくつかのニューロンが設けられており、それらをまとめて中間層と呼んでいる。各層のニューロンは、シナプス結合と呼ばれる結合で結ばれており、ニューロンの刺激を、入力層から出力層へ向かって一方通行に順次伝えるようになっている。また刺激の強さは、ニューロンやシナプスを伝わる間に、しきい値や重み付けにより変化する。

各セルからの出力値は、前述のとおり入力値に与えられる重みとしきい値によって設定されるが、その応答特性を表す代表モデルとして、式(2.1)で定義されるシグモイド関数があげられる。

$$f(u_i) = \frac{1}{1 + \exp(-u_i/T)} \quad (2.1)$$

ただし, $u_i = \sum_{i=1}^n w_i \cdot x_i - \theta$

- $f(u_i)$: ユニット i の出力値
- u_i : ユニット i への入力値
- n : ユニット i へ信号を出力するユニット (入力ユニット) 数
- x_i : 入力側ユニットからユニット i への出力値
- w_i : 入力側ユニットからユニット i への重み付け係数
- θ : ユニット i のしきい値
- T : 温度

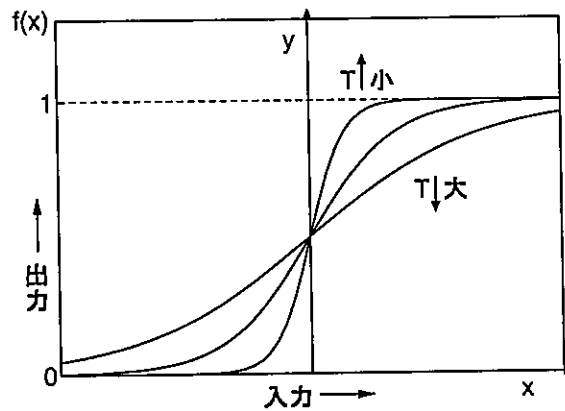


図- 2.1 シグモイド関数

ここで、式(2.1)と同じ形式で、 T の部分温度とする式が熱力学で使われているため、便宜上温度と呼んでいる。温度 T が大きくなるほど、S字型カーブは滑らかになり、逆に小さくなるほど、カーブは急勾配となる。ニューラルネットワークの学習速度や認識結果は、この温度 T の値により変化する。問題によっても、変化の傾向は異なるが、一般に小さくすれば学習時間は短くなる反面、未学習パターンに対する認識精度が低下する傾向がある⁷⁾。

このようにしてできたネットワークを用いて、未知のデータのパターン認識などを行う。一般的な特徴として、特に曖昧さを含むデータや、原因と結果の因果関係が明確な形で定義できない場合の情報処理に有効であることが知られている。具体的な応用例としてはパターン認識や制御、各種診断問題、予測・予知、最適化問題、信号処理などがあげられる。

感度解析

橋梁点検台帳²⁾のデータを用いて、ニューラルネットワークの感度解析を行った。使用したデータは206橋分であり、内訳はコンクリート橋(CO橋)146橋、鋼橋(ST橋)60橋となっている。解析を行うにあたって、橋梁の種類により橋梁各部位の重要度係数にも差が現れると推測されたことから、今回は大きく2つの種類(コンクリート橋・鋼橋)の橋に分別し、それぞれ解析を行った。

コンクリート橋146橋の点検データのうち136橋分のデータと、鋼橋60橋の点検データのうち54橋分のデータを教師データとし、残りのデータを検証用データとして感度解析を行い、各点検項目の重要度係数(重み付け係数)を求め、橋梁各部位の重要度として用いる。重要度係数を表-2.3と図-2.2に示す。

表- 2.3 感度解析による各点検項目の重要度係数

	橋面舗装	地覆高欄	床版	床組構	主構	支承	伸縮継手	排水装置	塗装	洗掘変動	躯体変動	安定構造	安定材質	耐震性
CO橋	0.17	0.23	0.30	0.13	0.08	0.13	0.23	0.08	—	0.18	0.12	0.36	0.32	0.17
ST橋	0.17	0.21	0.22	0.13	0.08	0.13	0.27	0.07	0.08	0.07	0.59	0.36	0.18	0.14

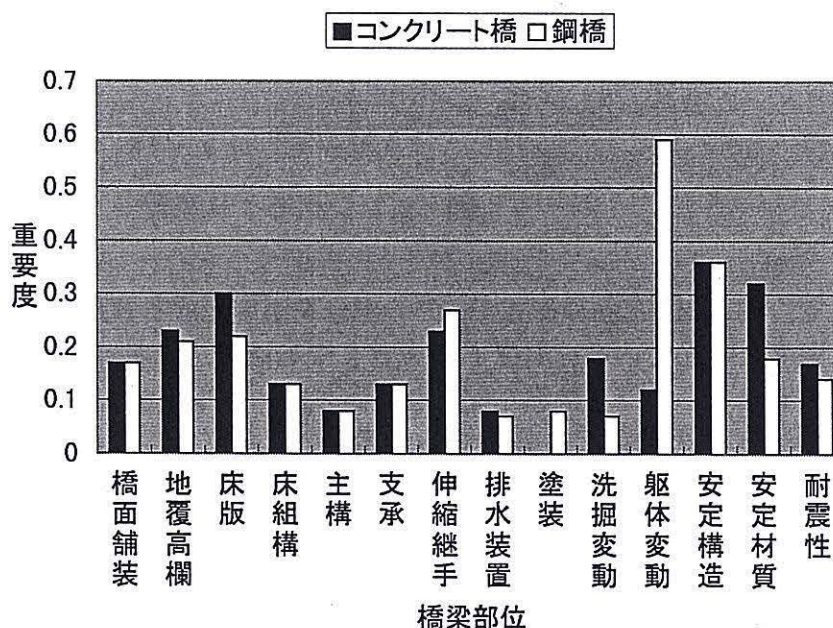


図- 2.2 橋梁各部位の重要度係数

ただし、橋梁点検台帳²⁾のRC橋において、塗装の項目のデータは存在しないため、塗装を抜いた13項目を用いて解析を行った。

結果を比較すると、コンクリート橋と鋼橋の重要度係数に違いがあるのが分かる。コンクリート橋においては安定構造・安定材質・床版の重要度が高く、それに対して鋼橋は躯体変動が大きな値を示しておりそれに安定構造・伸縮継手が続いている。

2.2.2 線形判別式

ニューラルネットワークの感度解析により、求めた重要度係数を用いて橋梁の評価を行うにあたって、線形判別式を利用する。線形判別式を式(2.2)と設定し、これから各橋梁の評価値を求める。

$$Y = \sum_{j=1}^R \sum_{k=1}^{c_j} w_j b_{jk} \delta_{jk} \quad (2.2)$$

ただし、

$$\delta_{jk} = \begin{cases} 1 \dots (\text{ある個体が要因の } j \text{ アイテム } k \text{ カテゴリーに反応したとき}) \\ 0 \dots (\text{その他のとき}) \end{cases}$$

$$\begin{cases} b_{j1} = 300 \\ b_{j2} = 200 \\ b_{j3} = 100 \end{cases} \quad (2.3)$$

ここで、 R はアイテム数(点検項目数14)、 c_j は j アイテムのカテゴリー数(1~3)、 w_j は j アイテム(各点検項目)の重要度係数である。

式(2.3)のように各要因(点検項目)の点検結果1, 2, 3に対し、順に300, 200, 100の点数を与えることによって、式(2.2)は、単調増加関数となる。

2.3 補修費用

2.3.1 橋梁各部位の補修費用

橋梁の各部位の補修内容と補修費用に関しては、専門家の協力を得て設定を行った。本来であるならば、橋梁の種類・損傷度の程度などにより、補修方法・補修費用が異なってくると思われるが、本研究においては表-2.4にあるように一様な設定をした。

橋梁点検台帳には点検データとともに、橋梁の基本的なデータとして橋梁の種類・橋梁面積・橋長が含まれている。この4つのデータより主桁・伸縮継手・橋台・杓・移動制限装置の数と、橋梁の塗装面積を求めて補修費用・工期の算出に用いた。それぞれの算出方法は以下に示すとおりである。なお、橋梁点検台帳に加え、橋梁データ管理支援システム⁸⁾のデータを参考に、43橋中42橋を単純橋とし、そして1橋を連続橋(ゲルバー式鋼桁橋)として算出した。

- 主桁 3m を主桁間隔の最大値として幅員より算出 (幅員÷3 ただし余りはくりあげ)
- 伸縮継手 支間数より算出 (支間数+1)
- 橋脚数 支間数より算出 (支間数-1)
- 橋台数 2
- 杓・移動制限装置 支間数と橋桁数から算出
 - 単純橋—主桁数×((橋脚数×2)+橋台数)
 - 連続橋—主桁数×(橋脚数+橋台数)

なお、塗装面積の算出については文献⁹⁾を参考にして行った。まず、橋梁の種類と支間長を基に、グラフ⁹⁾を参考に単位鋼重(kgf/m²)を求める。次に、支間長より単位塗装面(m²/tf)をグラフ⁹⁾を参考に求め、最終的に総鋼重と単位塗装面積を掛け合わせ橋梁の塗装面積とする。

この方法によって求めた各橋梁の諸データを表-2.5に示す。

表- 2.4 各部位の補修内容とその費用

部位	補修内容	補修費用(千円)
橋面 舗装	オーバーレイ	3 (単位面積当たり)
地覆 高欄	旧地覆撤去 地覆・高欄設置	80 (単位長さ当たり)
床版	旧床版撤去 新床版建設	100 (単位面積当たり)
床組構	増設桁工 (図-2.3)	850 (単位重量当たり)
支承	杓交換(50 tf)	500 (1箇所当たり)
伸縮 継手	ジョイント交換 (ゴム, 鋼, アルミ)	150 (単位長さ当たり)
排水 装置	塩化ビニル管交換	5 (単位長さ当たり)
塗装	ケレン, 下塗り, 中塗り, 上塗り	4 (単位面積当たり)
安定性 (材質)	ASR 補修	29 (単位面積当たり)
耐震性	移動制限装置設置	500 (1基当たり)

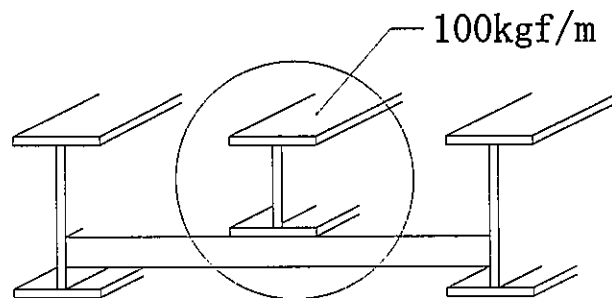


図- 2.3 増設桁工の重量

表- 2.5 橋梁データ

橋梁番号	橋梁面積	橋長	塗装面積	支間	継手	橋脚	沓
ST01	1311	218.5	8883	5	6	6	24
ST02	150.1	21.8	396.2	3	4	4	24
ST03	368	66.9	2314.4	1	2	2	6
ST04	465.5	133	2078.5	4	5	5	24
ST05	329	47	1421.3	1	2	2	8
ST06	313.5	57	526.7	3	4	4	18
ST07	546	91	1550.3	3	4	4	18
ST08	141.4	20.2	371.2	1	2	2	8
ST09	92.4	22	329.9	1	2	2	6
ST10	3866.5	407	21621.3	7	8	8	70
ST11	186	30	558	1	2	2	8
ST12	62	12.4	327.4	1	2	2	6
ST13	93.5	17	225	1	2	2	6
ST14	182.7	40.6	931.8	1	2	2	6
ST15	105.4	17	254.5	1	2	2	8
ST16	120	20	304	1	2	2	6
ST17	58.3	6.2	153.9	1	2	2	10
RC18	244.8	27.2	0	3	4	4	24
RC19	556	40	0	2	3	3	24
RC20	136	17	0	3	4	4	24
RC21	34.8	6.1	0	1	2	2	6
RC22	3105	172.5	0	9	10	10	126
RC23	105.3	12.1	0	1	2	2	8
RC24	102.5	20.5	0	2	3	3	12
RC25	118.6	20.8	0	2	3	3	12
RC26	86	19.1	0	2	3	3	12
RC27	127.5	17	0	1	2	2	8
RC28	153	17	0	2	3	3	16
RC29	67.5	15	0	1	2	2	6
RC30	234	39	0	2	3	3	12
RC31	322.4	49.6	0	2	3	3	16
RC32	120	20	0	1	2	2	6
RC33	799.3	66.6	0	2	3	2	20
RC34	70	14	0	1	2	2	6
RC35	78	12	0	1	2	2	8
RC36	55.7	10.5	0	1	2	2	6
RC37	45	10	0	1	2	2	6
RC38	100.8	14.4	0	1	2	2	8
RC39	50	10	0	1	2	2	6
RC30	85.4	11.7	0	1	2	2	8
RC41	97.5	15	0	1	2	2	8
RC42	114	19	0	1	2	2	6
RC43	91.2	8	0	1	2	2	10

面積・塗装面積の単位は m^2 ，橋長は m

2.3.2 補修足場の費用

足場は、作業者が地上または床上から手の届かない高所の施工をするために、組み立てる仮設構造物(施工設備)で、作業床や作業通路およびこれらを指示するものをさす。

足場は作業用の資材置場、作業者のための作業場所、施工管理者が工事を管理したり管理者が管理するための場所であり、また、管理者・施工管理者・作業者が、その場所に行くための通路としての機能も要求される。

足場は多種多様であるので、構造物の種類・構造・高さ・足場上の作業内容・作業員の数・足場上で使用する材料の重量・大きさ・数量・施工の順序・階段の位置・敷地境界・隣接構造物との関係・道路・材料の搬出入路・揚重方法・材料置場との関係などを考慮して使用する必要がある。

以下に、橋梁補修において用いられる主な足場の種類を示す。

- 枠組み足場図-2.4

足場の構成部材はあらかじめ工場で生産しているユニット構造であるため、組み立て・解体が容易で、なおかつ部材の強度上の信頼性も高いことから安全性に優れ、一般構造物の足場として現在最も多く使用されている。

- 張出し足場(ブラケット)図-2.5

本足場を下から建てると工期に影響したり、隣接して既設構造物があったりして地上から足場が組めない場合に、構造物の途中の外壁に水平の梁(張出し材)を取り付け、その上に本足場(枠組み足場、単管足場)を組み立てるものである。

- つり足場図-2.6

橋梁上部工の架設時などで、桁下空間に制限がある場合に用いられる。つり枠足場は既製のつり枠を鉄骨梁などに取り付け、それに足場板あるいは布板を設置した足場である¹⁰⁾。つり棚足場は、鉄骨梁などからつり下げたチェーンに鋼管を格子状に組み、その上に足場板を敷いた足場である。

本研究では橋梁各部位を補修するにあたり、専門家のアドバイスに基づいて表-2.6のように設定した。なお、足場の種類において、同じ番号の部位は足場を共有できるものとして補修計画を行う。

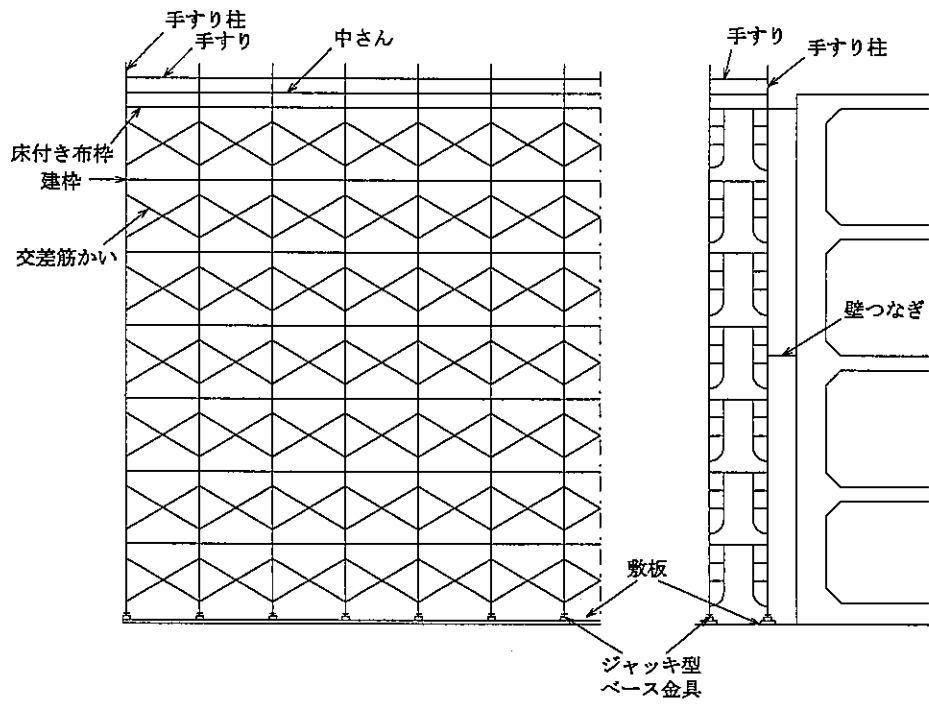


図- 2.4 枠組み足場

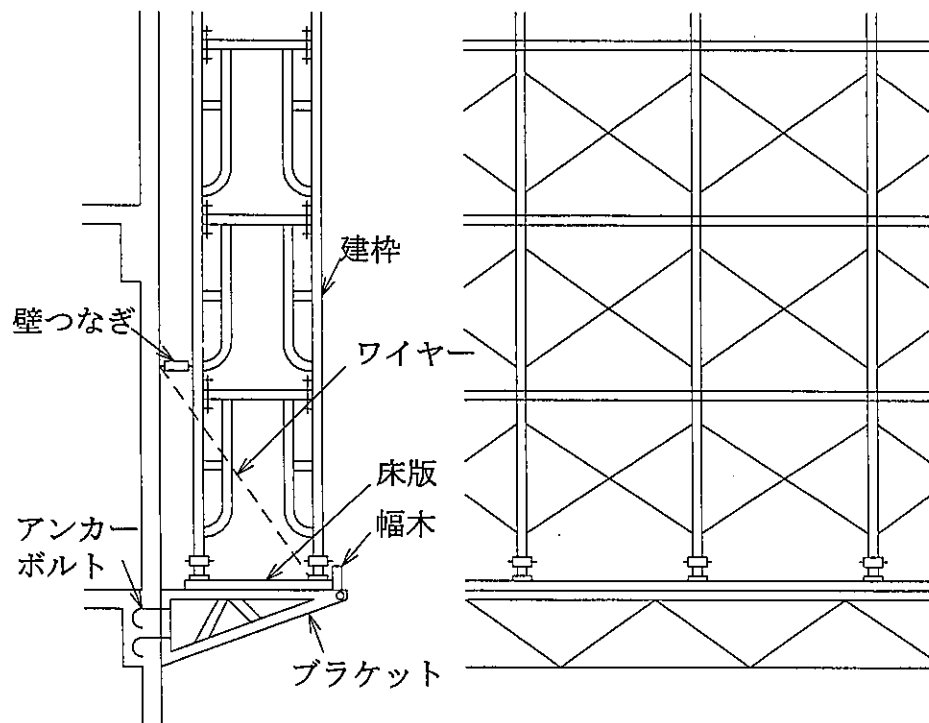


図- 2.5 張出し足場

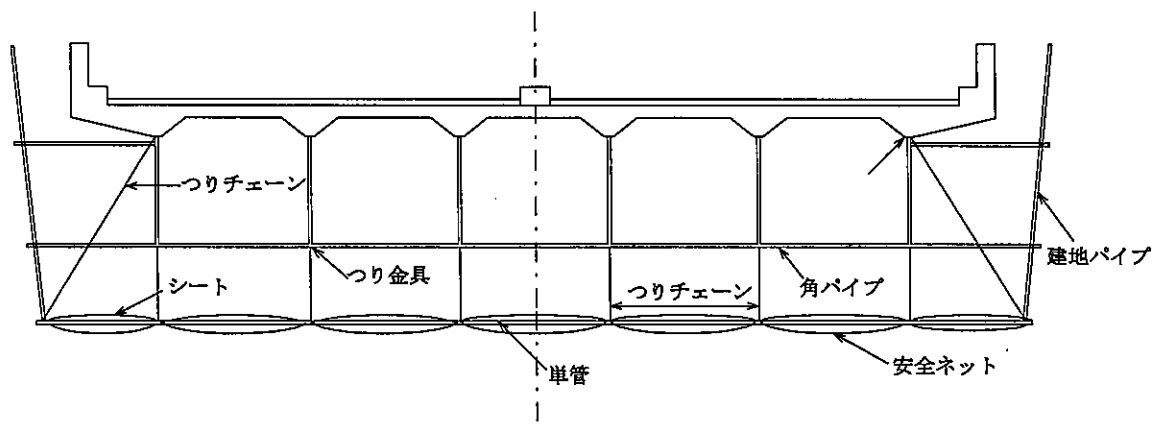


図- 2.6 つり足場

表- 2.6 各部位の足場の種類と費用

部位	足場の種類	費用(千円)
橋面 舗装	必要なし	
地覆 高欄	①つり足場	10 (単位長さ当たり)
床版	②つり足場	5 (単位面積当たり)
床組構	②つり足場	5 (単位面積当たり)
支承	③張り出し足場	3 (単位面積当たり)
伸縮 継手	必要なし	
排水 装置	②つり足場	5 (単位面積当たり)
塗装	②つり足場	2.5 (単位面積当たり)
安定性 (材質)	④枠組み足場	3 (単位面積当たり)
耐震性	③張り出し足場	3 (単位面積当たり)

2.4 橋梁補修計画案の作成

橋梁補修計画案の作成は、これまで述べてきた橋梁点検データ・橋梁各部位の重要度係数・補修費用を用いて行う。

橋梁補修計画案の対象とした橋梁は、橋梁点検台帳²⁾のデータ 206 橋のうち、損傷が著しいと橋梁自体の架け替えが必要となってしまう主構・洗掘変動・躯体変動・安定構造(下部工)の4つの点検項目が1(損傷なし)であった43橋である。この4つの項目を除いたのは、本研究は橋梁の補修計画を対象にしており、橋梁架け替えは考慮していないためである。

対象となる43橋の損傷した部位の中から、まず一定の予算内で最も効果をあげるような補修部位の組み合わせを求め、選択された補修部位を基に限られた期日でより多くの部位を補修できるように、スケジューリングを行い、これを補修計画案とする。

2.4.1 補修部位選択へのナップサック問題の適用

本研究では補修部位の選択を、ナップサック問題として扱った。ナップサック問題とは、式(2.4)、式(2.5)に表されるように制約式が1つだけの、形式的には最も単純な構造をもつ整数計画問題であるが、現実的応用例が豊富であり、かつ、しばしば複雑な問題を解く過程でその部分問題として現れてくることがある³⁾。

目的関数

$$z = \sum_{j=1}^n c_j x_j \rightarrow \max \quad (2.4)$$

制約条件

$$\sum_{j=1}^n a_j x_j \leq b \quad (2.5)$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, n$$

具体的な例として、一定の容量を持った袋(ナップサック)に複数個の荷物の中から幾つかを任意に選んで詰め込んだとき、袋の容量に最大限に積み込める荷物の組合せを求める問題を設定し、これを数学的に定義してみる。

最初に、袋の容量を V と定義する。次に、 n 個の荷物を集合 A で表し、その中の荷物の容量をその要素 a で表す。つまり、 n 個の要素からなる

$$A = \{a_1, a_2, \dots, a_n\} \quad (2.6)$$

という正の整数の集合を定義する。この中で、袋の容量 V 以内に収まる a の組合せの最大値を求めるのである。

これを計算すると以下のようなになる。まず集合 A から任意に m 個の要素を取り出し、その部分集合 P_m を作る。この部分集合の取り得る組み合わせの数は明らかに ${}_n C_m$ 個ある。その中で i 番目の部分集合を P_{mi} とする。この問題で求めるのは、 P_{mi} の部分集合の和を S_{mi} としたとき、

$$S_{mi} \leq V \quad (2.7)$$

の中で最大値を取る組み合わせの数は、それを X とすると、

$$X = {}_n C_1 + {}_n C_2 + \dots + {}_n C_n \quad (2.8)$$

$$= 2^n \quad (2.9)$$

となる¹¹⁾。

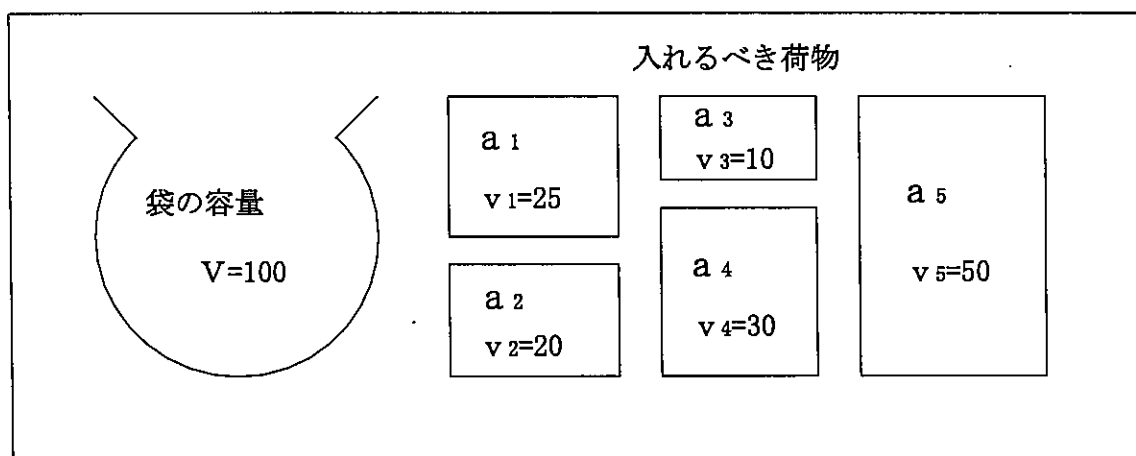


図- 2.7 ナップサック問題

本研究において、このナップサック問題を適用すると一定の予算(袋の容量)で最大限の効果をあげるように橋梁の補修部位(荷物)を選択していくことになる。ただし、補修部位の費用(荷物の容量)に加えてそれぞれの重要度(価値)を考慮しなければならないのである。

ナップサック問題は組み合わせ数が多いため、コンピューターを用いて解いたとしても非常に多くの時間を必要とする。先ほどの例において、 $n = 100$ とすると組合せの数は $X = 2^{100} \approx 1.3 \times 10^{20}$ 通りになり、1通りの計算に1秒かかるかすると、全ての組合せを計算し終わるまでに約 1.5×10^{15} 日という膨大な時間が必要となってくる。

そこで、本研究においては組み合わせ最適化問題に適した遺伝的アルゴリズムを適用することによって解探索時間の短縮を図る。

2.4.2 スケジューリング

ナップサック問題の適用によって採用された補修部位を基に、橋梁補修のスケジューリングを行う。ナップサック問題においては、予算という制約条件が存在したが、スケジューリングにおいては、限られた日数(1年であれば365日)でしか補修は行えないという制約条件が存在する。つまり、限られた予算・限られた期間で最も効果をあげる補修計画案の作成を行うわけである。本研究においては、補修工期を表-2.7のように設定し、橋梁各部位の補修工期の算出を行う。なお、1日当たりの工事実施時間を8時間とする。

橋梁を補修するにあたって、例えば舗装と伸縮継手の補修を行う場合、伸縮継手を先に補修しなければならない、など守らなければならない補修順序が存在し、補修計画案を作成するにあたり、その順序関係を守ることが要求される。本研究においては、図-2.8のように補修順序のアロー・ダイアグラムを設定し、どの橋梁においてもこの補修順序を守らなければならないものとして補修計画案の作成を行う。

遺伝的アルゴリズムは組み合わせ最適化問題のみならずスケジューリング問題にも適用することが可能であるので、本研究においても遺伝的アルゴリズムを適用する。

最後に、本研究で使用した43橋の点検データを表-2.8に示す。

表- 2.7 各橋梁部位の補修工期

部位	補修工期(単位:日)
橋面 舗装	0.05 (単位面積当たり)
地覆 高欄	0.5 (単位長さ当た)
床版	1 (単位面積当たり)
床組構	5 (単位重量当たり)
支承	3 (一個所当たり)
伸縮 継手	0.5 (単位長さ当たり)
排水 装置	0.05 (単位長さ当たり)
塗装	0.02 (単位面積当たり)
安定性 (材質)	0.5 (単位面積当たり)
耐震性	2 (一基当たり)

ただし、1日の補修実施時間を8時間とする

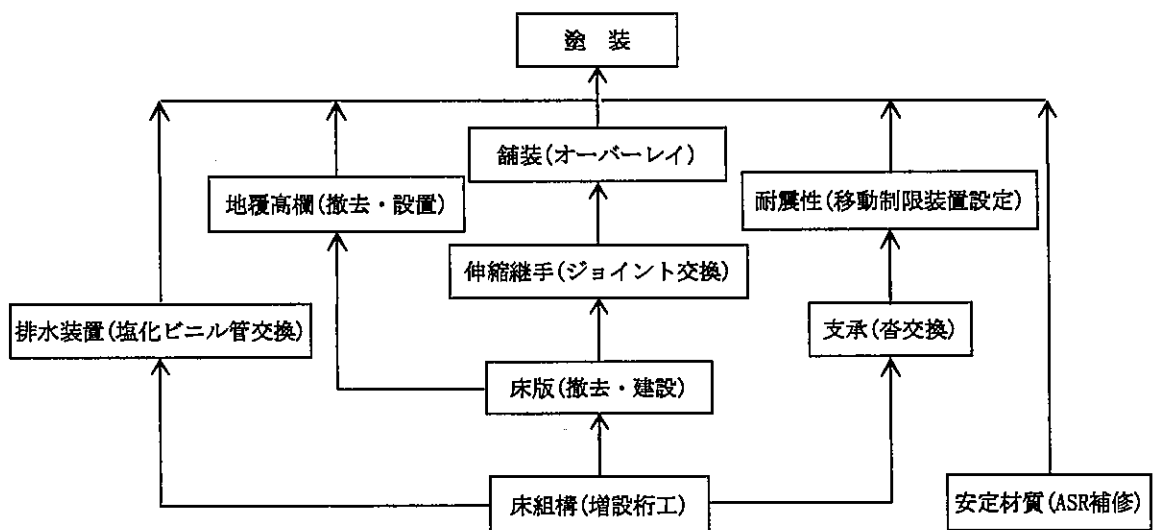


図- 2.8 補修順序のアロー・ダイアグラム

2.5 遺伝的アルゴリズムの概念

遺伝的アルゴリズム (Genetic Algorithm, 以下 GA と記す) は, Holland らによって 1960 年代に提唱された計算理論である. この計算理論は最近になって最適化問題に適した方法論として, 急速に注目を集めている.

現実世界には, 最適解はわからないが, 解の評価は可能であるという問題が多数存在する. 実際, 最適解のわかっている問題は, すでに解けている問題か, 解法がわかっている問題であり, 多くの場合 GA を使う必要はない. 現実世界で解きたい問題は, 基本的に最適解が知られてなく解法も確立していない問題か, 最適解の求め方が非常に非効率で実用にならないなどという問題がある. 特に, 大規模で複雑な技術・工学の問題に対しては, 往々にしてアルゴリズムが明確でないので, プログラム化できない. プログラム化できたとしても, 膨大な時間と手間がかかる (組み合わせ爆発) という難点が生じる. 一方, 実際上の問題に対しては, 厳密な解は必ずしも必要でなく, 「良い近似解」が得られれば十分である. したがって, 解の評価ができればよいという GA は, 現実問題に応用するのに適した手法である.

GA は, その名が示すように, 生物の遺伝と進化を模倣するモデルとして考えられた計算理論である. 簡単にいえば, 解くべき問題の解を遺伝子列 (染色体) として表現するものとし, それらを任意に複数個設定することにより集団を形成した上で, 「より優れた親 (解) より多くの子孫を残すことができる. また, 子どもは親である 2 つの解の性質を個々に受け継ぐため, 場合によっては, 親よりも優れた解を得ることができる (当然, 逆の場合も有り得る).」という状況下で, 数十世代~数百世代にわたって世代交替を繰り返すことによって, 確率的に優れた解を求めようとするものである¹²⁾.

本システムにおいても, 第 2 章末で述べたようにナップサック問題の適用により膨大な組合せの中から最適解を求めなければならない. また, GA はスケジューリング問題にも適用することが可能である. そこで, 本システムに GA を適用した.

2.6 遺伝的アルゴリズムの処理過程

GA の基本的な流れをまとめると, 図-2.9 のようになる.

まず, 遺伝子をもつ仮想的な生物 (個体) を複数個発生させる. さらに, 各個体の適応度をそれぞれの環境に対して計算し, 低い適応度を持つ個体を淘汰して消滅させ, 高い適応度を持つ個体を増殖させ, 親の形質を継承した遺伝子をもつ子孫の個体を生成する世代交替シミュレーションを繰り返す. この際, 実際の生物の生殖においても生ずる, 遺伝子の交叉, および突然変異と呼ばれる操作を行う. そして, 最終的に, 最も適応度の高い個体を求める.

これより, GA の基本的な流れに従って説明を行う.

表- 2.8 対象データ

橋梁名	路線名	橋面舗装	地覆高欄	床版	床組構	主構	支承	伸縮継手	排水装置	塗装	洗掘変動	躯体変動	安定構造	安定材質	耐震性	損傷評価	面積	橋長	支間
ST01	主 Route01	1	1	3	2	1	1	2	1	1	1	1	1	1	1	2	1289	219	5
ST02	主 Route02	1	2	3	1	1	2	2	3	3	1	1	1	1	2	2	117	22	3
ST03	一 Route03	3	2	1	1	1	1	2	1	1	1	1	1	2	2	2	368	67	1
ST04	一 Route04	2	2	2	1	1	1	2	1	1	1	1	1	1	2	2	576	133	4
ST05	国 Route05	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	329	47	1
ST06	主 Route06	1	1	2	1	1	1	2	1	1	1	1	1	1	2	2	313	57	3
ST07	主 Route07	1	1	2	1	1	1	1	1	1	1	1	1	2	1	1	191	91	3
ST08	国 Route05	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	141	20	1
ST09	主 Route08	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	92	22	1
ST10	主 Route09	1	1	2	1	1	1	1	1	2	1	1	1	1	1	3	4270	305	7
ST11	主 Route10	2	1	1	1	1	2	1	1	1	1	1	1	1	1	3	165	30	1
ST12	一 Route11	2	1	2	2	1	2	2	1	2	1	1	1	1	2	2	57	12	1
ST13	主 Route12	2	2	1	1	1	1	2	1	2	1	1	1	1	1	3	93	17	1
ST14	一 Route13	3	2	2	1	1	1	2	2	2	1	1	1	1	2	3	182	41	1
ST15	一 Route14	2	1	2	1	1	1	2	1	1	1	1	1	1	1	3	105	17	1
ST16	一 Route15	3	2	1	1	1	1	1	1	1	1	1	1	1	1	3	120	20	1
ST17	一 Route16	2	1	1	1	1	2	2	1	1	1	1	1	1	2	3	45	6	1
RC18	主 Route17	2	1	1	1	1	1	2	2	0	1	1	1	1	1	3	163	27	3
RC19	国 Route18	1	1	1	1	1	3	1	1	0	1	1	1	1	3	1	340	34	2
RC20	一 Route19	2	2	1	1	1	1	3	3	0	1	1	1	1	1	3	126	16	3
RC21	国 Route20	1	1	1	1	1	1	2	1	0	1	1	1	1	1	3	40	6	1
RC22	主 Route01	1	1	2	1	1	1	1	1	0	1	1	1	2	2	2	3107	173	9
RC23	一 Route21	1	1	1	2	1	1	1	1	0	1	1	1	1	1	2	82	13	1
RC24	主 Route22	1	3	1	1	1	1	2	1	0	1	1	1	1	1	3	103	27	2
RC25	主 Route23	2	2	1	1	1	1	2	1	0	1	1	1	1	1	3	119	21	2
RC26	国 Route24	2	2	1	1	1	2	2	1	0	1	1	1	1	2	2	86	19	2
RC27	主 Route25	3	1	1	2	1	1	1	1	0	1	1	1	1	1	2	102	17	1
RC28	国 Route05	2	1	1	2	1	2	1	1	0	1	1	1	1	2	2	170	17	2
RC29	主 Route26	1	1	1	1	1	2	1	1	0	1	1	1	1	2	2	68	15	1
RC30	国 Route05	1	1	1	1	1	2	1	1	0	1	1	1	1	2	2	234	39	2
RC31	主 Route27	1	2	1	1	1	1	1	1	0	1	1	1	1	2	2	273	50	2
RC32	主 Route28	1	1	1	1	1	1	1	1	0	1	1	1	1	1	3	120	20	1
RC33	国 Route05	1	2	1	1	1	1	2	1	0	1	1	1	1	1	3	466	67	2
RC34	主 Route25	2	1	1	2	1	1	2	1	0	1	1	1	1	2	2	70	14	1
RC35	一 Route29	1	1	1	1	1	2	3	1	0	1	1	1	1	2	3	90	12	1
RC36	一 Route30	2	2	1	1	1	2	3	1	0	1	1	1	1	2	3	79	5	1
RC37	主 Route31	3	3	1	1	1	2	3	1	0	1	1	1	2	2	3	45	10	1
RC38	一 Route32	1	1	1	2	1	1	2	1	0	1	1	1	1	2	3	64	15	1
RC39	一 Route33	2	1	1	2	1	2	3	1	0	1	1	1	1	2	3	45	10	1
RC30	国 Route05	1	1	1	2	1	2	2	2	0	1	1	1	1	1	3	85	12	1
RC41	主 Route34	1	1	1	1	1	2	3	1	0	1	1	1	1	2	2	98	15	1
RC42	国 Route05	2	1	1	1	1	1	2	1	0	1	1	1	1	1	3	118	20	1
RC43	一 Route35	3	1	1	1	1	1	1	1	0	1	1	1	1	1	3	56	8	1

面積の単位はm²、橋長はm
 国：国道，一：一般県道，主：主要地方道

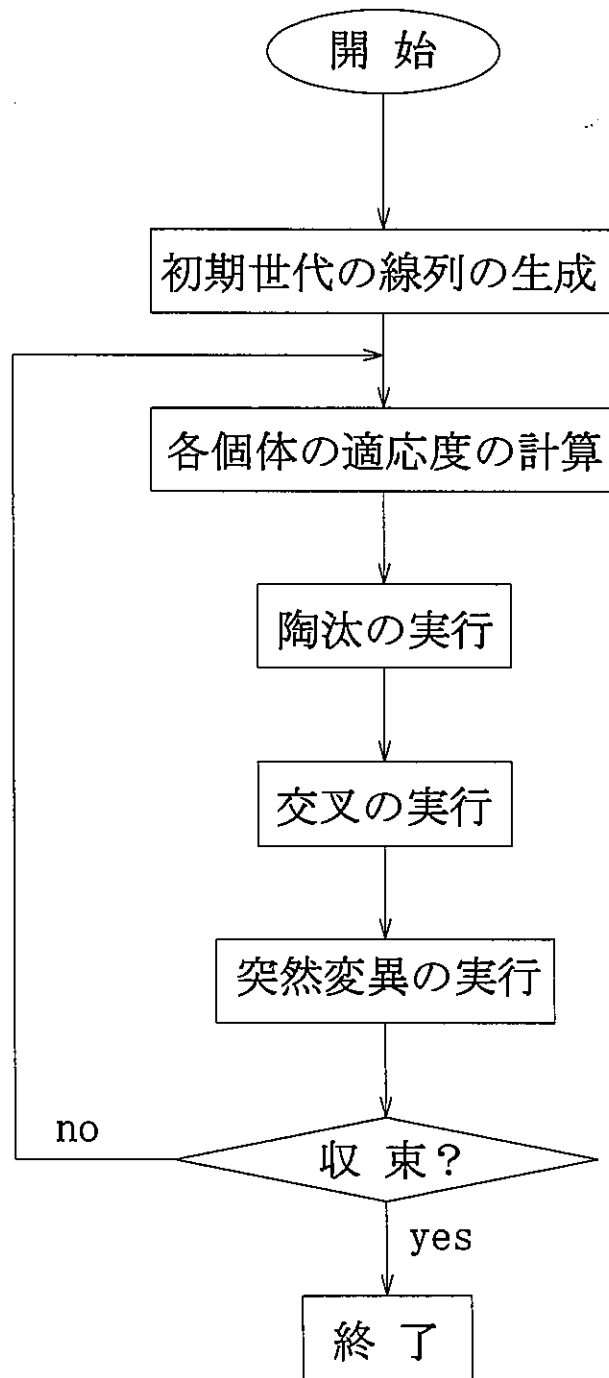


図- 2.9 遺伝的アルゴリズムの基本的な流れ

2.6.1 コーディングと初期世代の生成

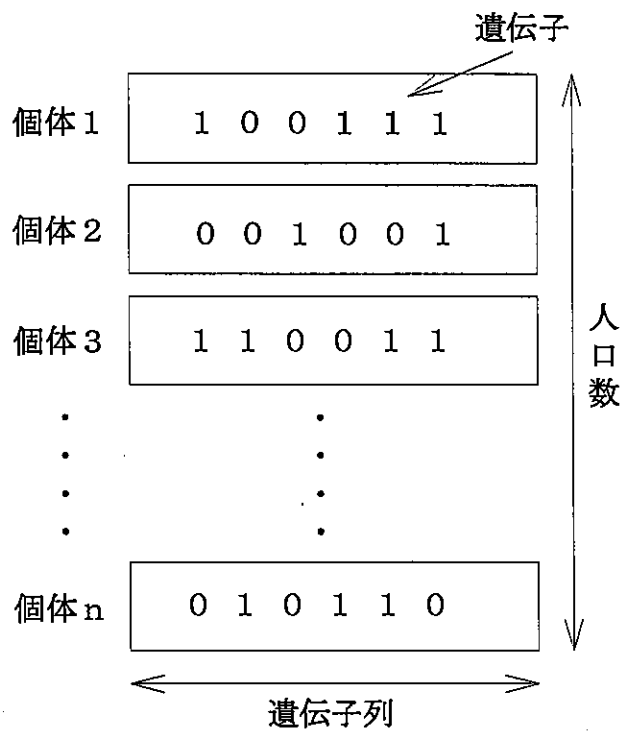


図- 2.10 初期世代の生成

実際の問題に GA を適用するには、まず数種類の記号 (一般的には、0 と 1) を用意し、解候補をこの記号の線列にコーディング (コード化) しなければならない (図-2.10)。この記号は「遺伝子」と呼ばれ、生物界では DNA の 4 種類の塩基に当たると考えてよい。

GA では、個体が常にある程度の数以上並存していないと、遺伝と進化は起こらない。各個体は基本的には同じ遺伝子型を持つが、その細部の違いにより多数の個性を持つ個体が出現する。多数の集団を発生させることにより、進化のスピードを向上させることになる。

初期世代の生成のさせ方は、特に定まっているわけではないが、乱数を用いてランダムに生成する方法が一般的である。

2.6.2 適応度

最適化の目的関数値がより高い変数ほど、それに対応する個体 (遺伝子列) の適応度が大きくなるように適応度を定義 (設計) する。その際、目的関数値と適応度の間に適切な非線形関係を持たせるなどの工夫を施す¹³⁾。

2.6.3 淘汰

GA における世代交替による自然淘汰において、集団に含まれる個体をより優れたものにするためには、優れたもののみを選択し、その子孫を残していけばよいことになる。しかし、一方では、優れたもののみを選択してしまうと集団の中の多様性が急速になくなり、GA の多点探索のメリットが失われることになり、あまり好ましくない局所解に収束してしまうことになる。

そのような状況を避け、集団の多様性を保ちつつ、徐々に準最適な解が求められるように、以下の方法が考えられている。

- ルーレット方式
- エリート保存方式
- ランキング方式
- 期待値による方法

これらの中で、淘汰によく用いられるルーレット方式とエリート保存方式について以下に詳しく述べる。

ルーレット方式

いまの世代の N 個の個体 $A_1 \sim A_n$ から、重複を許して N 個の個体をランダムに選択することによって、次の世代の N 個の個体を決定する。ここで、ある個体 A_i が、次の世代の個体として選択される確率 $P(A_i)$ を次式によって決定する。

$$P(a_i) = \frac{f(A_i)}{\sum_{i=1}^N f(A_i)} \times N \quad (2.10)$$

ここで、右辺の分子は個体 A_i の適応度、分母は現在の集団の平均適応度である。すなわち、各個体の次世代における生存の可能性は、自分の適応度に比例する。このため、適応度が高い個体ほど、次世代の個体として選ばれる確率が大きい。

ルーレット方式は、図-2.11 のように考えると理解しやすい。図では、個体 A_1 から A_N に対応する扇型の角度 θ_1 から θ_N は、それぞれ $f(A_1)$ から $f(A_N)$ に比例するようにつくられている。ルーレット方式は、このルーレットを N 回だけ回し、それぞれの回でルーレットの矢印にきた個体を重複を許して次の世代の個体として選択することと等価である¹⁴⁾。

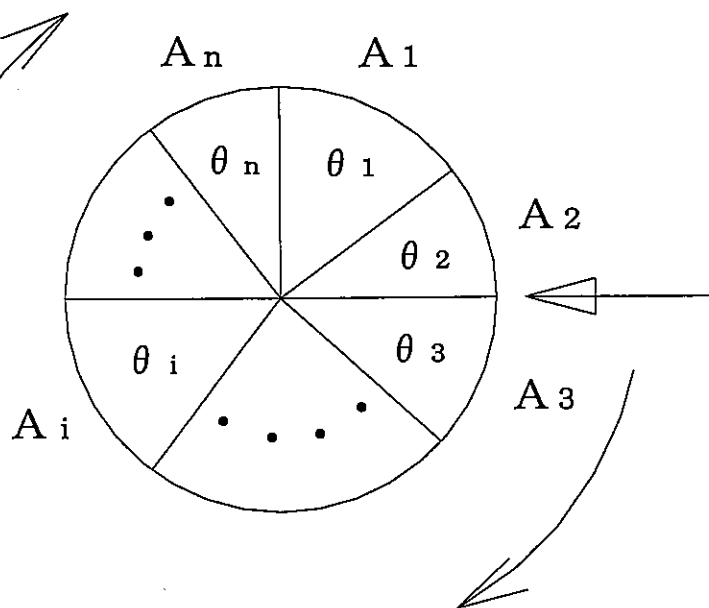


図- 2.11 ルーレット方式

エリート保存方式

エリート保存戦略は、集団中で最も適応度の高い個体をそのまま次世代に残す方法である。

この方法を採用すると、その時点で最も適応度の高い個体が交叉や突然変異で破壊されないという利点がある。ただし、エリート個体の遺伝子が集団中に急速に広がる可能性が高いため、局所解に陥る危険もある¹⁵⁾。

2.6.4 交叉

交叉は、二つの親の遺伝子列を組み替えて、この遺伝子を作る操作である。これは、部分的な解を含んだ線列同士を組み合わせることができるというGAの最も特徴的なものである。

1点交叉

最も単純な方法は、交叉する位置を1つ決めて、その前と後でどちらの親の遺伝子型を受け継ぐかを変える方法である。これを単純交叉、または1点交叉と呼ぶ。図-2.12では、3番目と4番目の遺伝子間が交叉位置となっており、個体Aの染色体の先頭から3番目までと個体Bの4番目から最後までまでの遺伝子が結合され、新しい個体の遺伝子となる。個体Bも同様な結合を行い、2つの個体から次世代へ新たな染色体を持った2つの個体が生まれるわけである¹⁶⁾。

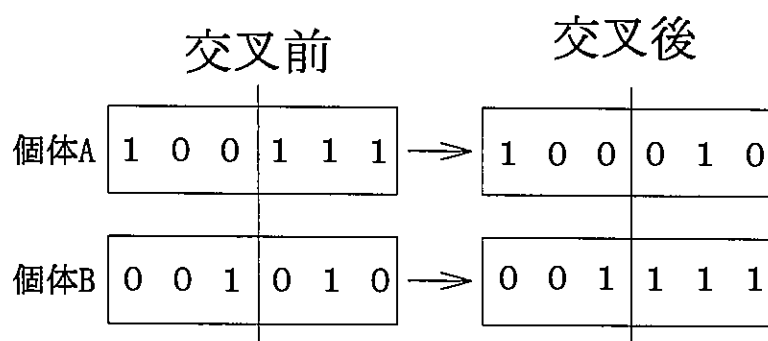


図- 2.12 1点交叉

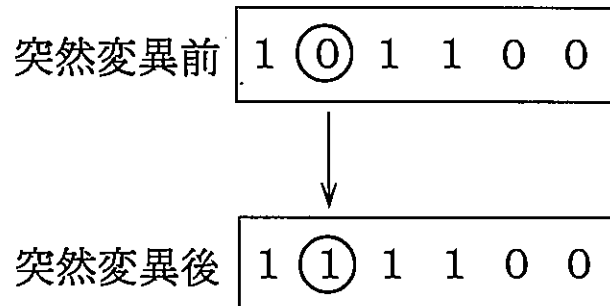


図- 2.13 突然変異

2.6.5 突然変異

突然変異は、遺伝子を一定の確率で変化させる操作である (図-2.13 では、2 番目の遺伝子が 0 から 1 に変化している)。この突然変異の操作により、交叉だけでは生じない遺伝子をもつ個体が生成されることになる。これは、局所解に陥った場合、そこから脱出する働きを持つ。

突然変異は、あまり大きな生起確率に設定するとランダム・サーチと化してしまうが、ある程度の変異は必要である。突然変異がない場合は、初期の遺伝子の組合せ以外の空間を探索することはできず、求める解の質にも限界が出てくる。一般的に、突然変異は固定された確率で各遺伝子が変わるように設定するが、突然変異発生確率を動的に変化させる手法もある¹⁶⁾。

2.6.6 収束条件

GA は適応度の計算、淘汰、交叉、突然変異という操作をある収束条件に当てはまるまで繰り返す。以下に主な収束条件をあげる。

1. ある世代まで繰り返したとき。
2. 最も良い適応度が数十～数百世代に渡って変化しないとき。
3. 適応度がある値になったとき。
4. 集団中に同じ遺伝子列 (個体) が一定以上存在するとき。

複数の条件を組み合わせて用いる場合もある。

2.7 遺伝的アルゴリズムの特徴と弱点

GAの研究は、未だ比較的浅い歴史しか持っていない。それだけに、そのアルゴリズムとしての性質や振る舞いの特徴についても未だ不明の点が多い。ここでは、GAの特徴ならびに弱点を整理してみる¹⁶⁾。

特徴

1. GAは、逐次近似あるいは逐次改良によって、解を探索するアルゴリズムである。解候補(個体)の集合を扱うので多点探索法の一つである。また、適応度の値のみを用いるので、直接探索法とすることができる。
2. 交叉は、「良い解候補」には「良い部分解」が含まれ、それらを組み合わせることによって「より良い解候補」が得られることを期待するものである(Building Block 仮説)。
3. 淘汰、交叉などの操作は確率的な要素を含むものであるから、GAは確率探索法の要素を持っている。確率を決めるパラメータを適切に選ぶことによって、局所解を回避できる可能性がある。

弱点

1. 有限の世代数で必ずしも最適解が得られるとは限らない。
2. 局所解が多数存在するような問題、特に最適解が狭い領域に孤立して存在するような問題では、GAによる探索は困難である。このような問題に関しては、適応度の定義(設計)に注意を要する。
3. 局所解の初期収束の可能性があり最適解までたどり着けないことがある。

2.8 補修部位の選択

2.8.1 GAへの適用

点検データのコーディング

本システムでは、補修部位の選択において、ナップサック問題として定式化し、GAを援用するにあたって図-2.14のように線列のコーディングを行った。コーディングの手順は、以下のとおりである。

1. 点検データ(A)より点検結果が2(Δ)あるいは3(\times)である補修の必要な項目を抜き出し、要補修データ(A)'を作成する。これは、点検結果が1(\circ)であった補修する必

要のない項目にも遺伝子を与え線列を長くすると、組み合わせ数が大幅に増え、GAの解探索の効率に大きな障害を与えるためである。

2. 要補修データ (A)' で選ばれた項目の数だけ、0か1の乱数を発生させ線列を生成する。なお、0の遺伝子は補修を行わないことを表し、1の遺伝子は補修することを表している。
3. 与えられた遺伝子を基に、補修後の橋梁部位の評価 (B) を作成する。床版・支承・排水装置・塗装・安定材質には1の遺伝子を与えられているので、補修が行われる。なお、中途半端な補修は行わないと設定し、補修を行った項目はすべて損傷度を1(○)とする。

図-2.14には、1橋分の線列しかないが、実際には43橋分のデータを遺伝子化して1つの個体として扱う。

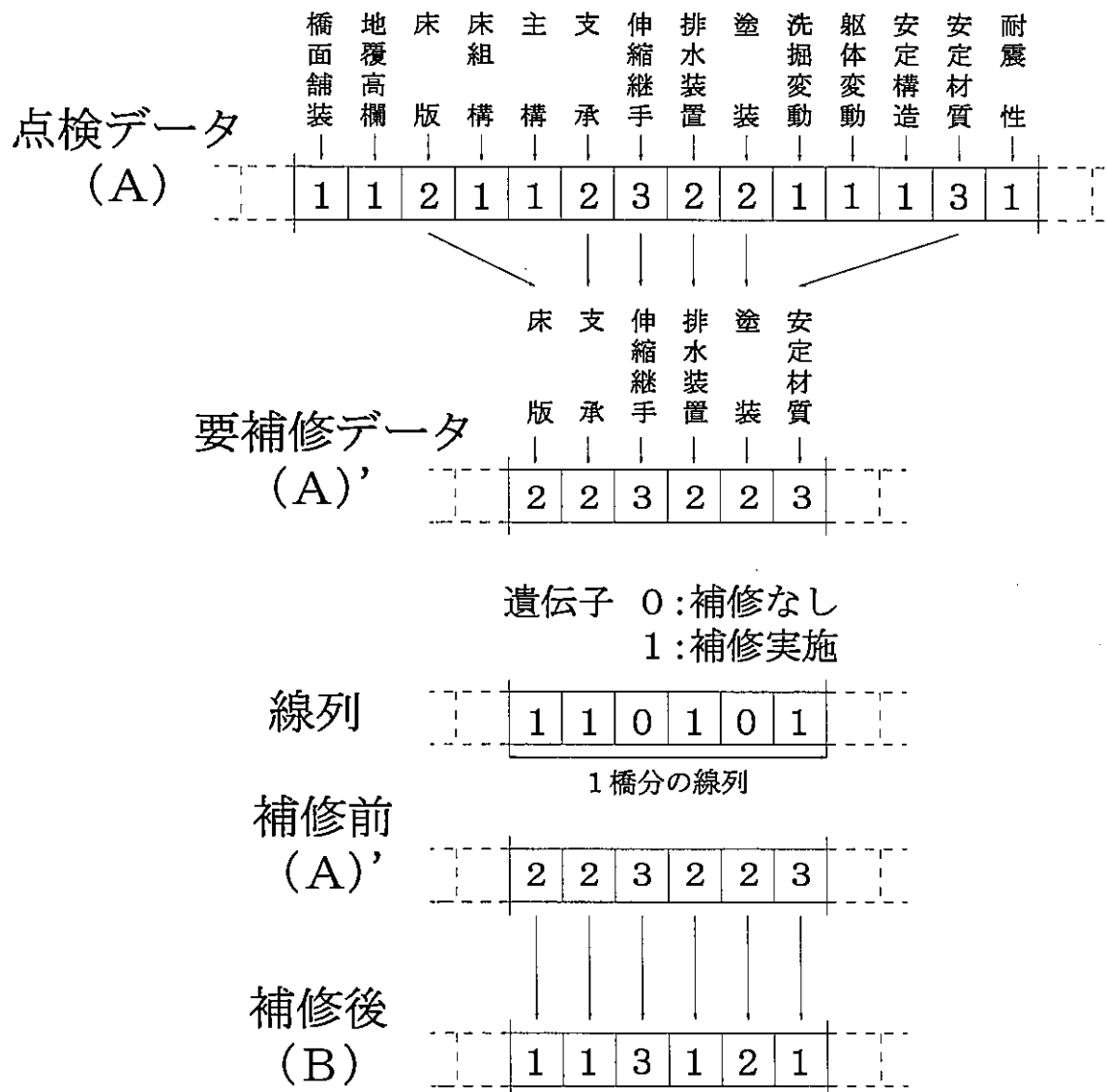


図- 2.14 補修橋梁の遺伝子フォーマット

淘汰方法

木下⁴⁾の研究において橋梁維持管理支援システムに、淘汰方法にエリートルールとルーレットルールを用いた解析と結果の比較が行われている。人口数を 20 から、300 までの間のいくつかの値に変えて 2 つの淘汰方法で解析を行った結果、すべての人口数においてルーレットルールよりもエリートルールを用いた方が、適応度の高い橋梁補修計画案が作成されている。ルーレットルールの場合は、人口数が大きくなるに従って次世代に残す個体がランダムに選択することと等価になり、実際に人口数が高いほど適応度の低い橋梁補修計画案が作成されている。この結果を受けて、淘汰方法にはエリートルールを採用する。

ペナルティ関数

GA では制約条件を含む問題をペナルティ関数法を用いて無制約の問題に変換する必要がある。そのため、本システムでは、目的関数 F を制約条件 g と合わせて式 (2.11) のように定式化した。 r はペナルティ係数と呼ばれるもので、右辺第 1 項の目的関数に対する第 2 項のペナルティ項の効き具合を調節するためのパラメータである。

$$\Psi = \begin{cases} F & (g \geq 0) \\ F - r(-g) & (g < 0) \end{cases} \quad (2.11)$$

制約条件が満たされない ($g < 0$) 場合、目的関数にペナルティ項が加えられることによって制約条件があるのと同様の効果をもたらされる。制約条件 g は後程式 (2.13) で与えられており、予算より補修計画案の補修費用を引いた値であり、補修費用が予算をオーバーした場合にペナルティとして目的関数値が減少するように設定している。なお、 $r = 50$ とする。

収束条件

補修部位選択においては、GA の収束条件を以下のように設定し収束条件を満たした場合 GA の計算を終了とし、収束世代において最も高い適応度をもつ個体を基に橋梁補修計画案の作成を行う。

- 集団中の適応度最大の個体が 1000 世代更新されなかった場合。
- 集団中の個体が、すべて同一になった場合。
- 世代交代が 5000 世代に到達した場合。

表- 2.9 補修部位選択における GA の諸値

集団数	: 1
人口数	: 100
エリート個体数	: 1
交叉	: 一点交叉
交叉率	: 0.6
突然変異率	: 0.05

GA の諸値

以上に述べてきた GA の設定に加え，GA の諸値を表-2.9 に示す。

2.8.2 投資効率を考慮した関数による解析

GA においては各個体に適応度を与え，より高い適応度の個体を見つけ出すために計算を行う。各個体の適応度を計算するために，本研究では2種類の目的関数の設定を行い解析を進めていく。まず，これまでの橋梁維持管理支援システムにおいて最も高い効果をあげた式 (2.12) に示した，投資効率を考慮した関数による解析を行う。

この関数は，余った予算で橋梁の評価値増加量を割ることにより，費用に対する評価値の増加量，つまり投資効率を考慮した関数である。分母は最小に，分子は最大になるように作用させることにより目的関数値を最大化させ，より効果をあげるような橋梁補修代替案の作成を目指す。

投資効率を考慮した目的関数

$$F_a = \frac{\sum_{i=1}^n (e_{i2} - e_{i1})}{\sqrt{g} + \delta} \Rightarrow \max \quad (2.12)$$

制約条件

$$g = E - C \geq 0 \quad (2.13)$$

- n : 橋梁数
- e_{i2} : 補修後の i 番目の橋梁の評価値
- e_{i1} : 補修前の i 番目の橋梁の評価値
- E : 予算
- C : 補修費用合計
- δ : 補正值

この関数においては δ の値が、目的関数値に大きな影響を与えるものと思われる。 δ の値を 10^0 から 10^8 まで変化させ、これを予算 5 千万円、1 億円、2 億円においてそれぞれ解析を行い、最適な δ の値の探索と橋梁補修部位選択における δ の与える影響を調べる。

評価値増加量と補修費用の比較

予算 5 千万円の場合の結果を図-2.15 に、予算 1 億円の場合を図-2.16 に、そして予算 2 億円の場合の解析結果を図-2.17 に示す。これらのグラフは、それぞれ 10 回ずつ解析を行い、選択された補修部位の補修費用と評価値の増加量の平均値を示している。

解析結果を検討すると、全体的な傾向としては評価値増加量は δ の値が増加するに従って増加しており、補修費用は δ の増加に反して減少する傾向となっている。つまり、 δ が小さければ予算を使いきるような橋梁補修計画案に、そして δ が大きければ、より評価値増加量がより大きくなるような橋梁補修計画案が作成されていると言える。評価値増加量は δ の値の増加とともに収束へと向かって行くが、補修費用は δ が大きくなり 10^4 か 10^5 を越えると補修費用が増加することがあることが分かる。これは、 δ の増加に伴って δ の値に対して g のレンジが小さくなり、 g の与える影響が少なくなったためではないかと思われる。

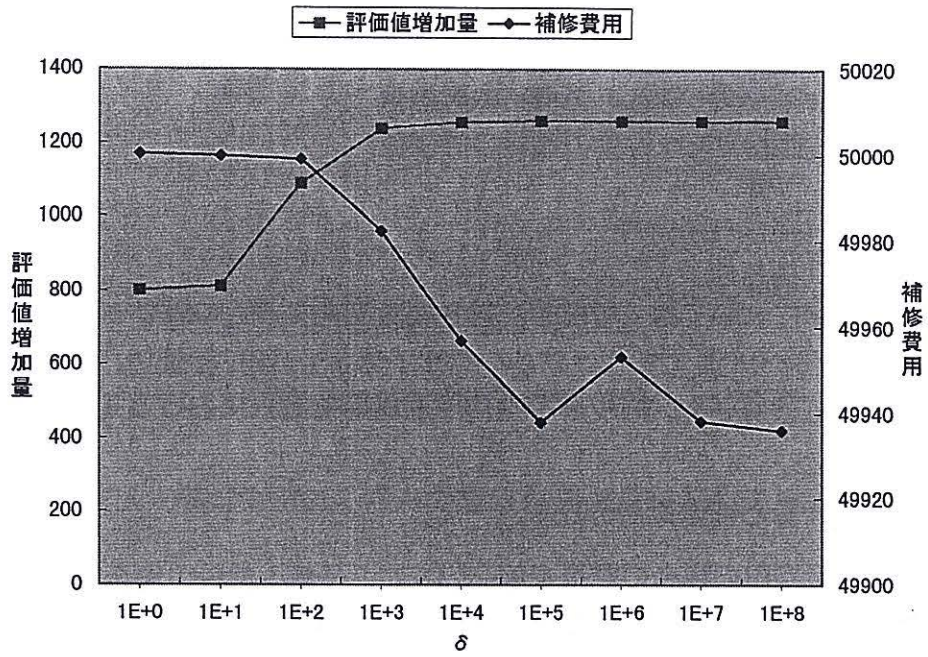


図- 2.15 予算5千万円における δ の検討

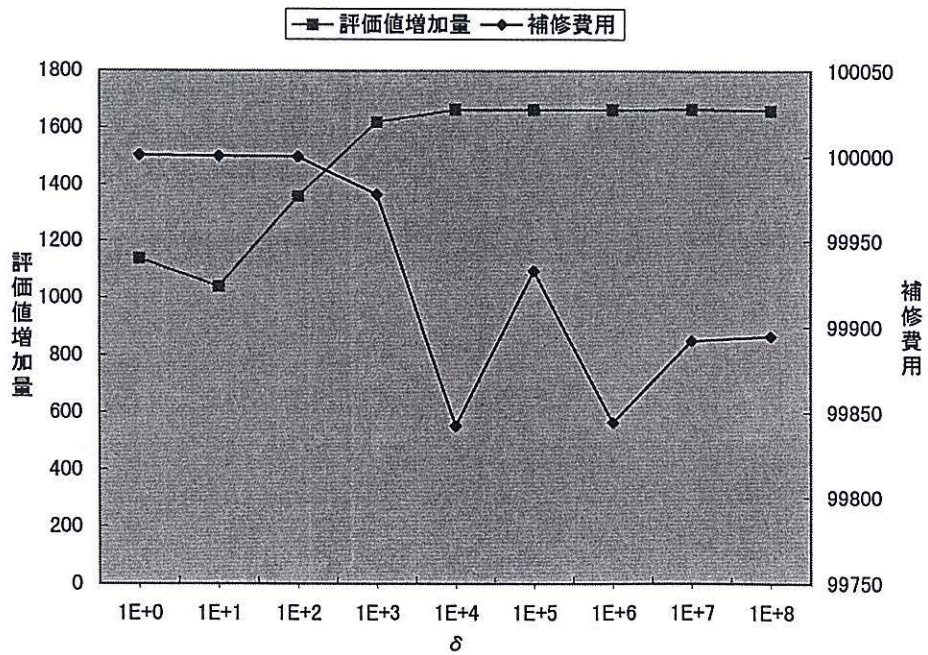


図- 2.16 予算1億円における δ の検討

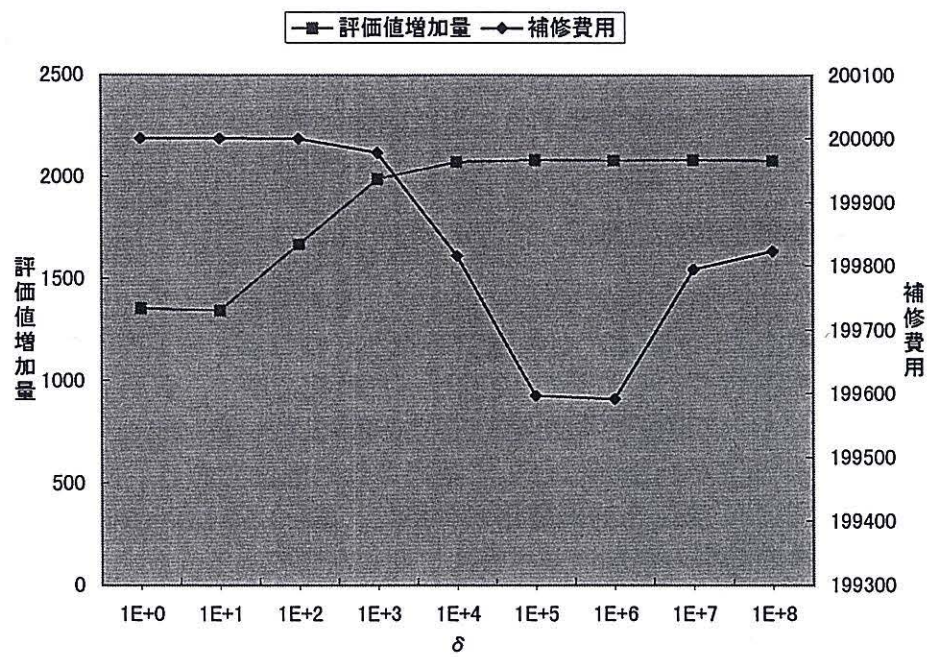


図- 2.17 予算2億円におけるδの検討

表- 2.10 δ の最適値

予算	最適な δ の値
5 千万円	10^5
1 億円	10^4
2 億円	10^6

補修部位を選択するにあたって、同じ評価値増加量を得ることができるならば、なるべく補修予算が少なく済む橋梁補修計画案の方が好ましいと考え、解析結果を基に表-2.10のようにそれぞれの予算における δ の最適値を決定する。

この結果、予算と最適であると思われる δ の値との相関がわかりにくいことが分かる。予算から最適な δ の値を推測することは困難に思われる。

選択された補修部位の比較

次に、 δ の値が補修部位の選択にどのように影響を与えているかを調べるために、予算 5 千万円において、 $\delta = 10^0$ と最適であると思われる $\delta = 10^5$ において選択された橋梁補修部位の比較を行う。それぞれ 10 回行った解析の中で、最も評価値増加量の高い結果となった解析において選択された補修部位を表-2.12・表-2.13・表-2.14 に示す。”○”が記されている部位が補修部位として選択された部位であり、表-2.12 は 43 橋の中で補修が必要となる 134 部位を補修費用の昇順に、表-2.13 は評価値増加量の降順に、そして表-2.14 は投資効率の降順に並べ替えて記した表である。なお、投資効率とは各部位の評価値増加量を補修費用で割った値である。

また表-2.11 に、それぞれの δ における評価値増加量・補修費用・選択補修部位数を示す。

表- 2.11 補修部位選択の諸値

δ	10^0	10^5
評価値増加量	918	1266
補修費用 (千円)	50000	49904
選択補修部位数	37	54

表- 2.12 予算5千万円の補修費用昇順におけるδの違いによる比較

橋種	橋梁番号	部位	補修費用 単位:円	評価値増加量	δ		橋種	橋梁番号	部位	補修費用 単位:円	評価値増加量	δ	
					10 ⁰	10 ⁵						10 ⁰	10 ⁵
CO	40	排水装置	117000	8	○	○	ST	12	耐震性	3000000	14		
CO	37	橋面舗装	135000	34	○	○	ST	14	耐震性	3000000	14		
CO	39	橋面舗装	150000	17	○	○	CO	29	支承	3000000	13		
CO	36	橋面舗装	166950	17	○	○	CO	29	耐震性	3000000	17	○	
CO	20	排水装置	170000	16	○	○	CO	34	耐震性	3000000	17	○	
ST	17	橋面舗装	174840	17	○	○	CO	36	支承	3000000	13		
ST	12	橋面舗装	186000	17	○	○	CO	36	耐震性	3000000	17		
CO	34	橋面舗装	210000	17	○	○	CO	37	支承	3000000	13		
ST	2	排水装置	217500	14	○	○	CO	37	耐震性	3000000	17		
CO	26	橋面舗装	257850	17	○	○	CO	39	支承	3000000	13		
CO	18	排水装置	272000	8	○	○	CO	39	耐震性	3000000	17		
CO	43	橋面舗装	273600	34	○	○	CO	26	地覆高欄	3056000	23		
ST	13	橋面舗装	280500	17	○	○	ST	16	地覆高欄	3200000	21		
ST	15	橋面舗装	316200	17	○	○	CO	24	地覆高欄	3280000	46	○	○
CO	42	橋面舗装	342000	17	○	○	ST	6	伸縮装置	3300000	27		
CO	25	橋面舗装	355680	17	○	○	CO	25	地覆高欄	3328000	23		
ST	16	橋面舗装	360000	34	○	○	ST	2	地覆高欄	3480000	21		
CO	27	橋面舗装	382500	34	○	○	ST	14	塗装	3727080	8		
ST	14	排水装置	406000	7	○	○	ST	5	耐震性	4000000	14		
CO	20	橋面舗装	408000	17	○	○	ST	11	支承	4000000	13		
ST	12	床組構	425000	13	○	○	CO	35	支承	4000000	13		
CO	34	床組構	425000	13	○	○	CO	35	耐震性	4000000	17		
CO	39	床組構	425000	13	○	○	CO	38	耐震性	4000000	17		
CO	28	橋面舗装	459000	17	○	○	CO	40	支承	4000000	13		
ST	1	床組構	510000	13	○	○	CO	41	支承	4000000	13		
ST	14	橋面舗装	548100	34	○	○	CO	41	耐震性	4000000	17		
ST	11	橋面舗装	558000	17	○	○	ST	2	伸縮装置	4140000	27	○	○
CO	38	床組構	595000	13	○	○	CO	20	伸縮装置	4800000	46		
CO	40	床組構	620500	13	○	○	ST	17	支承	5000000	13		
CO	27	床組構	637500	13	○	○	ST	17	耐震性	5000000	14		
CO	18	橋面舗装	734400	17	○	○	ST	1	伸縮装置	5400000	27		
CO	23	床組構	739500	13	○	○	CO	18	伸縮装置	5400000	23		
CO	28	床組構	765000	13	○	○	CO	33	伸縮装置	5400000	23		
ST	13	塗装	900022.2	8	○	○	CO	26	支承	6000000	13		
ST	3	橋面舗装	1103850	34	○	○	CO	26	耐震性	6000000	17		
CO	37	安定材質	1305000	32	○	○	CO	30	支承	6000000	13		
ST	12	塗装	1309440	8	○	○	CO	30	耐震性	6000000	17		
ST	14	伸縮装置	1350000	27	○	○	ST	12	床版	6200000	22		
CO	37	伸縮装置	1350000	46	○	○	ST	14	地覆高欄	6486000	21		
ST	4	橋面舗装	1396500	17	○	○	CO	30	安定材質	6786000	23		
ST	8	塗装	1484700	8	○	○	CO	31	地覆高欄	7936000	23		
ST	12	伸縮装置	1500000	27	○	○	CO	28	支承	8000000	13		
CO	34	伸縮装置	1500000	23	○	○	CO	28	耐震性	8000000	17		
CO	39	伸縮装置	1500000	46	○	○	CO	31	耐震性	8000000	17		
ST	2	塗装	1584792	16	○	○	ST	6	耐震性	9000000	14		
CO	36	伸縮装置	1690000	46	○	○	ST	5	安定材質	9541000	18		
CO	37	地覆高欄	1600000	46	○	○	ST	15	床版	10540000	22		
ST	3	伸縮装置	1650000	27	○	○	CO	33	地覆高欄	10657600	23		
ST	13	伸縮装置	1650000	27	○	○	ST	3	安定材質	10670550	18		
CO	36	地覆高欄	1680000	23	○	○	ST	3	地覆高欄	10704000	21		
CO	21	伸縮装置	1710000	23	○	○	ST	2	支承	12000000	13		
CO	42	伸縮装置	1800000	23	○	○	ST	2	耐震性	12000000	14		
ST	15	伸縮装置	1860000	27	○	○	CO	19	支承	12000000	26		
CO	35	伸縮装置	1950000	46	○	○	CO	19	耐震性	12000000	34		
CO	41	伸縮装置	1950000	46	○	○	ST	2	床版	15007500	44		
CO	26	伸縮装置	2025000	23	○	○	ST	7	安定材質	15834000	18		
CO	38	伸縮装置	2100000	23	○	○	ST	14	床版	18270000	22		
CO	40	伸縮装置	2190000	23	○	○	ST	4	地覆高欄	21280000	21		
CO	24	伸縮装置	2260000	23	○	○	ST	6	床版	31350000	22		
CO	25	伸縮装置	2565000	23	○	○	ST	4	床版	46550000	22		
ST	4	伸縮装置	2625000	27	○	○	ST	7	床版	54600000	22		
ST	13	地覆高欄	2720000	21	○	○	CO	22	耐震性	63000000	17		
CO	20	地覆高欄	2720000	23	○	○	ST	10	塗装	86485313.4	8		
ST	17	伸縮装置	2820000	27	○	○	CO	22	安定材質	90045000	32		
ST	3	耐震性	3000000	14	○	○	ST	1	床版	131100000	44		
ST	9	耐震性	3000000	14	○	○	CO	22	床版	310500000	30		
ST	12	支承	3000000	13	○	○	ST	10	床版	386650000	22		

表- 2.13 予算5千万円の評価値増加量降順におけるδの違いによる比較

橋種	橋架番号	部位	補修費用 単位:円	評価値増加量	δ		橋種	橋架番号	部位	補修費用 単位:円	評価値増加量	δ	
					10 ⁰	10 ⁵						10 ⁰	10 ⁵
CO	20	伸縮装置	4800000	46	○		ST	13	橋面舗装	280500	17	○	
CO	24	地覆高欄	3280000	46	○	○	ST	15	橋面舗装	316200	17	○	○
CO	35	伸縮装置	1950000	46			ST	17	橋面舗装	174840	17	○	○
CO	36	伸縮装置	1590000	46	○	○	CO	18	橋面舗装	734400	17	○	○
CO	37	地覆高欄	1600000	46	○	○	CO	20	橋面舗装	408000	17	○	○
CO	37	伸縮装置	1350000	46	○	○	CO	22	耐震性	6300000	17		
CO	39	伸縮装置	1500000	46	○	○	CO	25	橋面舗装	355680	17		
CO	41	伸縮装置	1950000	46	○	○	CO	26	橋面舗装	257850	17	○	○
ST	1	床版	131100000	44			CO	26	耐震性	6000000	17		
ST	2	床版	15007500	44			CO	28	橋面舗装	459000	17	○	○
ST	3	橋面舗装	1103850	34		○	CO	28	耐震性	8000000	17		
ST	14	橋面舗装	548100	34	○	○	CO	29	耐震性	3000000	17	○	○
ST	16	橋面舗装	360000	34	○	○	CO	30	耐震性	6000000	17		
CO	19	耐震性	12000000	34			CO	31	耐震性	8000000	17		
CO	27	橋面舗装	382500	34	○	○	CO	34	橋面舗装	210000	17	○	○
CO	37	橋面舗装	135000	34	○	○	CO	34	耐震性	3000000	17	○	○
CO	43	橋面舗装	273600	34	○	○	CO	35	耐震性	4000000	17		
CO	22	安定材質	90045000	32			CO	36	橋面舗装	166950	17		○
CO	30	安定材質	6786000	32			CO	36	耐震性	3000000	17		
CO	37	安定材質	1305000	32	○	○	CO	37	耐震性	3000000	17		
CO	22	床版	310500000	30			CO	38	耐震性	4000000	17		
ST	1	伸縮装置	5400000	27			CO	39	橋面舗装	150000	17	○	○
ST	2	伸縮装置	4140000	27	○	○	CO	39	耐震性	3000000	17		
ST	3	伸縮装置	1650000	27	○	○	CO	41	耐震性	4000000	17		
ST	4	伸縮装置	2625000	27			CO	42	橋面舗装	342000	17		○
ST	6	伸縮装置	3300000	27			ST	2	塗装	1584792	16		
ST	12	伸縮装置	1500000	27		○	CO	20	排水装置	170000	16		○
ST	13	伸縮装置	1650000	27	○	○	ST	2	排水装置	217500	14	○	○
ST	14	伸縮装置	1350000	27	○	○	ST	2	耐震性	12000000	14		
ST	15	伸縮装置	1860000	27	○	○	ST	3	耐震性	3000000	14		
ST	17	伸縮装置	2820000	27	○	○	ST	5	耐震性	4000000	14		
CO	19	支承	12000000	26			ST	6	耐震性	9000000	14		
CO	18	伸縮装置	5400000	23			ST	9	耐震性	3000000	14		
CO	20	地覆高欄	2720000	23	○	○	ST	12	耐震性	3000000	14		
CO	21	伸縮装置	1710000	23	○	○	ST	14	耐震性	3000000	14		
CO	24	伸縮装置	2250000	23			ST	17	耐震性	5000000	14		
CO	25	地覆高欄	3328000	23			ST	1	床組構	510000	13	○	○
CO	25	伸縮装置	2565000	23	○	○	ST	2	支承	12000000	13		
CO	26	地覆高欄	3056000	23			ST	11	支承	4000000	13		
CO	26	伸縮装置	2025000	23	○	○	ST	12	床組構	425000	13		○
CO	31	地覆高欄	7936000	23			ST	12	支承	3000000	13		
CO	33	地覆高欄	10657600	23			ST	17	支承	5000000	13		
CO	33	伸縮装置	5400000	23			CO	23	床組構	739500	13	○	○
CO	34	伸縮装置	1500000	23			CO	26	支承	6000000	13		
CO	36	地覆高欄	1680000	23	○	○	CO	27	床組構	637500	13	○	○
CO	38	伸縮装置	2100000	23			CO	28	床組構	765000	13	○	○
CO	40	伸縮装置	2190000	23			CO	28	支承	8000000	13		
CO	42	伸縮装置	1800000	23		○	CO	29	支承	3000000	13		
ST	4	床版	46550000	22			CO	30	支承	6000000	13		
ST	6	床版	31350000	22			CO	34	床組構	425000	13	○	○
ST	7	床版	54600000	22			CO	35	支承	4000000	13		
ST	10	床版	386650000	22			CO	36	支承	3000000	13		
ST	12	床版	6200000	22			CO	37	支承	3000000	13		
ST	14	床版	18270000	22			CO	38	床組構	595000	13		○
ST	15	床版	10540000	22			CO	39	床組構	425000	13	○	○
ST	2	地覆高欄	3480000	21			CO	39	支承	3000000	13		
ST	3	地覆高欄	10704000	21			CO	40	床組構	620500	13	○	○
ST	4	地覆高欄	21280000	21			CO	40	支承	4000000	13		
ST	13	地覆高欄	2720000	21			CO	41	支承	4000000	13		
ST	14	地覆高欄	6496000	21			ST	8	塗装	1484700	8		
ST	16	地覆高欄	3200000	21			ST	10	塗装	86485313.4	8		
ST	3	安定材質	10670550	18			ST	12	塗装	1309440	8		
ST	5	安定材質	9541000	18			ST	13	塗装	900022.2	8		
ST	7	安定材質	15834000	18			ST	14	塗装	3727080	8		
ST	4	橋面舗装	1396500	17			CO	18	排水装置	272000	8		○
ST	11	橋面舗装	558000	17		○	CO	40	排水装置	117000	8	○	○
ST	12	橋面舗装	186000	17		○	ST	14	排水装置	406000	7		○

表- 2.14 予算5千万円の投資効率降順における δ の違いによる比較

橋種	橋梁番号	部位	投資効率	δ		橋種	橋梁番号	部位	投資効率	δ	
				10^0	10^5					10^0	10^5
CO	37	橋面舗装	0.000251852	○	○	ST	16	地覆高欄	6.5625E-06		
CO	43	橋面舗装	0.000124269	○	○	ST	2	伸縮装置	6.52174E-06	○	
CO	39	橋面舗装	0.000113333	○	○	ST	12	塗装	6.10948E-06		
CO	36	橋面舗装	0.000101827	○	○	ST	2	地覆高欄	6.03448E-06		
ST	17	橋面舗装	9.72318E-05	○	○	CO	29	耐震性	5.66667E-06	○	
ST	16	橋面舗装	9.44444E-05	○	○	CO	34	耐震性	5.66667E-06	○	
CO	20	排水装置	9.41176E-05	○	○	CO	36	耐震性	5.66667E-06		
ST	12	橋面舗装	9.13978E-05	○	○	CO	37	耐震性	5.66667E-06		
CO	27	橋面舗装	8.88889E-05	○	○	CO	39	耐震性	5.66667E-06		
CO	34	橋面舗装	8.09524E-05	○	○	ST	8	塗装	5.38829E-06		
CO	40	排水装置	6.83761E-05	○	○	ST	1	伸縮装置	0.000005		
CO	26	橋面舗装	6.59298E-05	○	○	CO	30	安定材質	4.71559E-06		
ST	2	排水装置	6.43678E-05	○	○	ST	3	耐震性	4.66667E-06		
ST	14	橋面舗装	6.20325E-05	○	○	ST	9	耐震性	4.66667E-06		
ST	13	橋面舗装	6.06061E-05	○	○	ST	12	耐震性	4.66667E-06		
ST	15	橋面舗装	5.37634E-05	○	○	ST	14	耐震性	4.66667E-06		
CO	42	橋面舗装	4.97076E-05	○	○	ST	12	支承	4.33333E-06		
CO	25	橋面舗装	4.77958E-05	○	○	CO	29	支承	4.33333E-06		
CO	20	橋面舗装	4.16667E-05	○	○	CO	36	支承	4.33333E-06		
CO	28	橋面舗装	3.7037E-05	○	○	CO	37	支承	4.33333E-06		
CO	37	伸縮装置	3.40741E-05	○	○	CO	39	支承	4.33333E-06		
ST	3	橋面舗装	3.08013E-05	○	○	CO	18	伸縮装置	4.25926E-06		
CO	39	伸縮装置	3.06667E-05	○	○	CO	33	伸縮装置	4.25926E-06		
ST	12	床組構	3.05882E-05	○	○	CO	35	耐震性	0.00000425		
CO	34	床組構	3.05882E-05	○	○	CO	38	耐震性	0.00000425		
CO	39	床組構	3.05882E-05	○	○	CO	41	耐震性	0.00000425		
ST	11	橋面舗装	3.04659E-05	○	○	ST	12	床版	3.54839E-06		
CO	18	排水装置	2.94118E-05	○	○	ST	5	耐震性	0.0000035		
CO	36	伸縮装置	2.89308E-05	○	○	ST	11	支承	0.00000325		
CO	37	地覆高欄	0.00002875	○	○	CO	35	支承	0.00000325		
ST	1	床組構	2.54902E-05	○	○	CO	40	支承	0.00000325		
CO	37	安定材質	2.45211E-05	○	○	CO	41	支承	0.00000325		
CO	35	伸縮装置	2.35897E-05	○	○	ST	14	地覆高欄	3.23276E-06		
CO	41	伸縮装置	2.35897E-05	○	○	ST	2	床版	2.93187E-06		
CO	18	橋面舗装	2.31481E-05	○	○	CO	31	地覆高欄	2.89819E-06		
CO	38	床組構	2.18487E-05	○	○	CO	19	耐震性	2.83333E-06		
CO	40	床組構	2.09508E-05	○	○	CO	26	耐震性	2.83333E-06		
CO	27	床組構	2.03922E-05	○	○	CO	30	耐震性	2.83333E-06		
ST	14	伸縮装置	0.00002	○	○	ST	17	耐震性	0.0000028		
ST	12	伸縮装置	0.000018	○	○	ST	17	支承	0.0000026		
CO	23	床組構	1.75794E-05	○	○	CO	19	支承	2.16667E-06		
ST	14	排水装置	1.72414E-05	○	○	CO	26	支承	2.16667E-06		
CO	28	床組構	1.69935E-05	○	○	CO	30	支承	2.16667E-06		
ST	3	伸縮装置	1.63636E-05	○	○	CO	33	地覆高欄	2.15808E-06		
ST	13	伸縮装置	1.63636E-05	○	○	ST	14	塗装	2.14645E-06		
CO	34	伸縮装置	1.53333E-05	○	○	CO	28	耐震性	0.000002125		
ST	15	伸縮装置	1.45161E-05	○	○	CO	31	耐震性	0.000002125		
CO	24	地覆高欄	1.40244E-05	○	○	ST	15	床版	2.08729E-06		
CO	36	地覆高欄	1.36905E-05	○	○	ST	3	地覆高欄	1.96188E-06		
CO	21	伸縮装置	1.34503E-05	○	○	ST	5	安定材質	1.86659E-06		
CO	42	伸縮装置	1.27778E-05	○	○	ST	3	安定材質	1.68689E-06		
ST	4	橋面舗装	1.21733E-05	○	○	CO	28	支承	0.000001625		
CO	26	伸縮装置	1.1358E-05	○	○	ST	6	耐震性	1.55556E-06		
CO	38	伸縮装置	1.09524E-05	○	○	ST	14	床版	1.20416E-06		
CO	40	伸縮装置	1.05023E-05	○	○	ST	2	耐震性	1.16667E-06		
ST	4	伸縮装置	1.02857E-05	○	○	ST	7	安定材質	1.13679E-06		
CO	24	伸縮装置	1.02222E-05	○	○	ST	2	支承	1.08333E-06		
ST	2	塗装	1.0096E-05	○	○	ST	4	地覆高欄	9.86842E-07		
CO	20	伸縮装置	9.58333E-06	○	○	ST	6	床版	7.01754E-07		
ST	17	伸縮装置	9.57447E-06	○	○	ST	4	床版	4.7261E-07		
CO	25	伸縮装置	8.96686E-06	○	○	ST	7	床版	4.0293E-07		
ST	13	塗装	8.88667E-06	○	○	CO	22	安定材質	3.55378E-07		
CO	20	地覆高欄	8.45688E-06	○	○	ST	1	床版	3.35022E-07		
ST	6	伸縮装置	8.18182E-06	○	○	CO	22	耐震性	2.69841E-07		
ST	13	地覆高欄	7.72059E-06	○	○	CO	22	床版	9.66184E-08		
CO	26	地覆高欄	7.52618E-06	○	○	ST	10	塗装	9.25013E-08		
CO	25	地覆高欄	6.91106E-06	○	○	ST	10	床版	5.6899E-08		

表-2.12 から、全体的な傾向として $\delta = 10^5$ の場合補修費用の低額な部位から選択していることがわかる。 $\delta = 10^0$ においては比較的高額な ST 橋・橋梁番号 2・伸縮装置と CO 橋・橋梁番号 20・伸縮装置が補修部位として選択されている。

表-2.13 より、 $\delta = 10^5$ においては評価値増加量が高い値を示している部位であっても補修費用が高額である場合は補修部位として選択されていないことがわかる。 $\delta = 10^0$ の場合は、橋種 CO・橋梁番号 20・伸縮装置と橋種 CO・橋梁番号 35・伸縮継手のように同じ評価値増加量であったとしても補修費用が高額な部位を補修部位として選択するような結果となっている。

表-2.14 を見ると、これまで評価してきた 2 つの表の評価を総合した結果となっていることが分かる。 $\delta = 10^5$ においては、ほぼ投資効率の高い部位から順に補修部位を選択しているが、 $\delta = 10^0$ においては、必ずしも投資効率を考慮した補修部位の選択が行われておらず、この関数の目的に促した結果が得られていない。

これらの結果から考察しても、 δ の値が小さい場合は評価値増加量をより高くすることよりも、予算を使いきることに主眼が置かれており、それに対して $\delta = 10^5$ においてはより多くの橋梁部位が補修部位として選択されており、また十分に投資効率を考慮した補修部位の選択が行われているといえる。

2.8.3 関数の無次元化

無次元化した目的関数

$$F_b = \frac{\sum_{i=1}^n (e_{i2} - e_{i1})}{\zeta} - \frac{g}{E} \Rightarrow \max \quad (2.14)$$

制約条件

$$g = E - C \geq 0$$

- n : 橋梁数
- e_{i2} : 補修後の i 番目の橋梁の評価値
- e_{i1} : 補修前の i 番目の橋梁の評価値
- E : 予算
- C : 補修費用合計
- ζ : 補正值

2 つ目の関数として無次元化したデータを扱う関数式 (2.14) を使用する。投資効率を考慮した関数においては、お金と橋梁の評価値という異なった値を扱っている。無次元化関

数では2つの異なった値の無次元化を行った後に、1つの関数で扱おうとする関数である。

無次元化した評価値の増加量から無次元化した余予算を除算するという関数であり、評価値増加量をなるべく高く、予算を使いきる橋梁補修計画案となるような橋梁補修部位の選択を行うという基本的な考え方は投資効率関数と同じである。

g は予算で割ることにより無次元化を行い、評価値の増加量は ζ で割ることにより無次元化を図る。これより ζ の値の決定を行うのに以下に示す2つの数値を候補としてあげ、予算5千万・1億円・2億円においてそれぞれ解析を行い ζ の値の選定を行う。

1. $\zeta = \text{総評価値増加量}$
2. $\zeta = \text{総評価値増加量} \times \frac{\text{予算}}{\text{総費用}}$

なお、総評価値増加量とは橋梁補修の対象となる43橋の要補修部位となる137部位すべてを補修した場合得られる評価値増加量を示しており、数値としては2814となる。また総費用とは137部位を補修するのに必要なお金であり、16億5607万7千円である。表-2.15に ζ の予算ごとの値を示す。

図-2.18と図-2.19は、10回ずつ解析を行いそれぞれ評価値の増加量と補修費用の平均値を表したグラフである。評価値増加量においては予算5千万・1億・2億いずれにおいても2番目の δ の値を使用した解析がわずかであるが、1番目の δ より高い評価値増加量を示す補修部位選択を行っていることが分かる。これは、1番目の ζ の値に比べ2番目の ζ の値が小さい値になっているため、式(2.14)の第二項に対する第一項の重みが大きくなったため評価値増加量の値が大きくなったのではないかと思われる。ただ、予算に関してはほとんど差が確認できなかった。

これらの結果を受けて、無次元化関数において評価値増加量の無次元化を行う ζ の値は $\zeta = \text{総評価値増加量} \times \frac{\text{予算}}{\text{総費用}}$ を採用する。

表- 2.15 ζ の値

ζ	5千万円	1億円	2億円
1	2814		
2	84.96	169.92	339.84

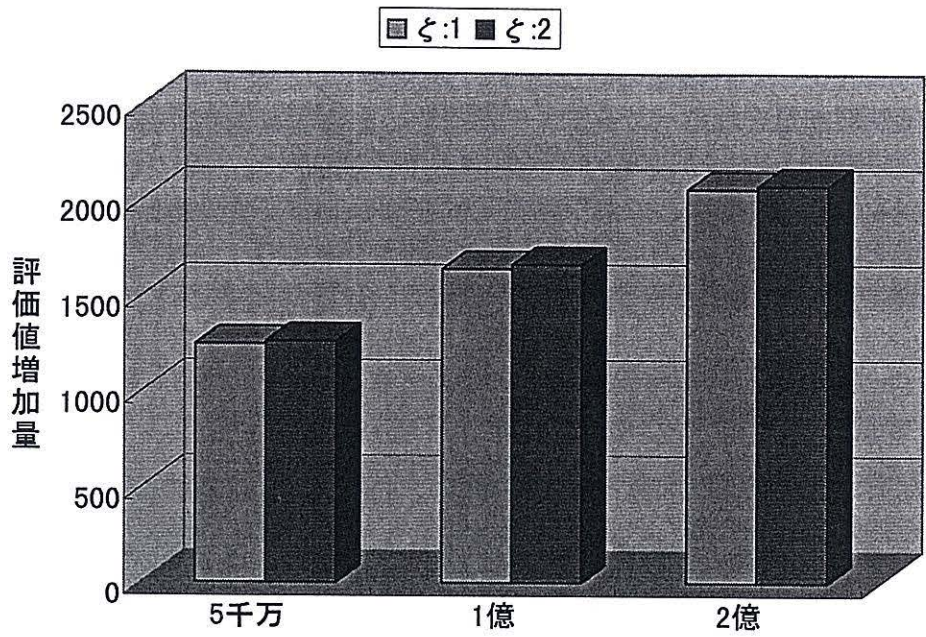


図- 2.18 ζによる評価値増加量への影響

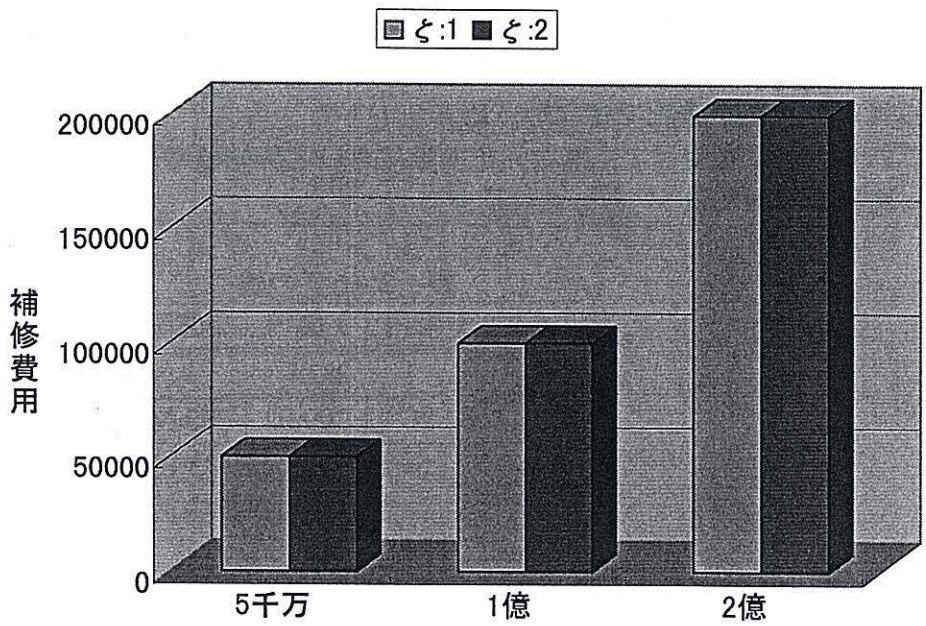


図- 2.19 ζによる補修費用への影響

2.8.4 2種類の関数の比較

ここではこれまで説明と解析を行ってきた、投資効率関数の結果と無次元化関数の解析結果の比較・検討を行う。

まず、図-2.20に予算5千万円・1億円・2億円として、それぞれの関数において採用した補正值を用いて行った10回の解析の評価値増加量の平均値を表して比較を行う。

この結果を見ると、どの予算に対しても2つの関数の差はほとんどなく、同等な効果が得られる橋梁補修部位の選択が行われていることが分かる。

具体的に、どのような補修部位の選択が行われているのか比較するために予算5千万円において、10回の解析の中でそれぞれ最も高い評価値増加量を示した補修部位選択が行われた解析の結果を表-2.17・表-2.18・表-2.19に示す。表の種類は投資効率関数で用いたものと同じであり、 F_a は投資効率関数、 F_b は無次元化関数の結果である。表-2.16に比較を行った橋梁補修計画案の詳しいデータを示す。

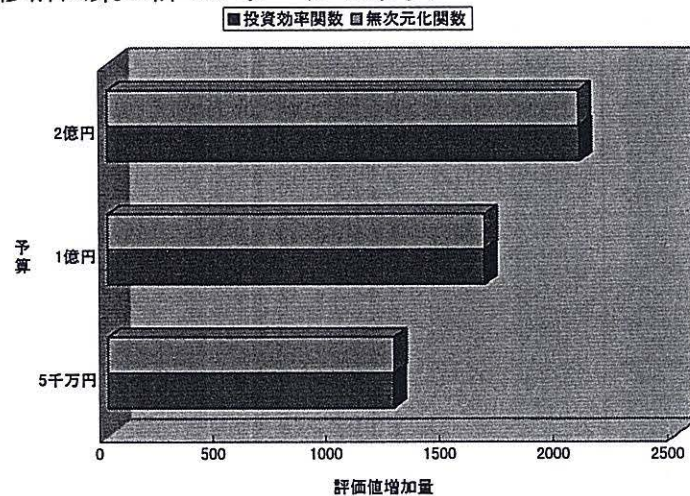


図- 2.20 2種類の関数の総合評価

表- 2.16 橋梁補修計画案の諸値

目的関数	投資効率関数	無次元化関数
補正值	$\delta = 10^5$	$\zeta = 84.96$
評価値増加量	1266	1268
補修費用(千円)	49904	49978
選択補修部位数	54	55

表-2.17 関数による補修部位選択の違い-予算5千万円の補修費用昇順-

橋種	橋梁番号	部位	補修費用 単位:円	評価値増加量	目的関数		橋種	橋梁番号	部位	補修費用 単位:円	評価値増加量	目的関数	
					F_a	F_b						F_a	F_b
CO	40	排水装置	117000	8	○	○	ST	12	耐震性	3000000	14		
CO	37	橋面舗装	135000	34	○	○	ST	14	耐震性	3000000	14		
CO	39	橋面舗装	150000	17	○	○	CO	29	支承	3000000	13		
CO	36	橋面舗装	166950	17	○	○	CO	29	耐震性	3000000	17		
CO	20	排水装置	170000	16	○	○	CO	34	耐震性	3000000	17		
ST	17	橋面舗装	174840	17	○	○	CO	36	支承	3000000	13		
ST	12	橋面舗装	186000	17	○	○	CO	36	耐震性	3000000	17		
CO	34	橋面舗装	210000	17	○	○	CO	37	支承	3000000	13		
ST	2	排水装置	217500	14	○	○	CO	37	耐震性	3000000	17		
CO	26	橋面舗装	257850	17	○	○	CO	39	支承	3000000	13		
CO	18	排水装置	272000	8	○	○	CO	39	耐震性	3000000	17		
CO	43	橋面舗装	273600	34	○	○	CO	26	地覆高欄	3056000	23		
ST	13	橋面舗装	280500	17	○	○	ST	16	地覆高欄	3200000	21		
ST	15	橋面舗装	316200	17	○	○	CO	24	地覆高欄	3280000	46	○	○
CO	42	橋面舗装	342000	17	○	○	ST	6	伸縮装置	3300000	27		
CO	25	橋面舗装	355630	17	○	○	CO	25	地覆高欄	3328000	23		
ST	16	橋面舗装	360000	34	○	○	ST	2	地覆高欄	3480000	21		
CO	27	橋面舗装	382500	34	○	○	ST	14	塗装	3727080	8		
ST	14	排水装置	408000	7	○	○	ST	5	耐震性	4000000	14		
CO	20	橋面舗装	408000	17	○	○	ST	11	支承	4000000	13		
ST	12	床組構	425000	13	○	○	CO	35	支承	4000000	13		
CO	34	床組構	425000	13	○	○	CO	35	耐震性	4000000	17		
CO	39	床組構	425000	13	○	○	CO	38	耐震性	4000000	17		
CO	28	橋面舗装	459000	17	○	○	CO	40	支承	4000000	13		
ST	1	床組構	510000	13	○	○	CO	41	支承	4000000	13		
ST	14	橋面舗装	548100	34	○	○	CO	41	耐震性	4000000	17		
ST	11	橋面舗装	558000	17	○	○	ST	2	伸縮装置	4140000	27		
CO	38	床組構	595000	13	○	○	CO	20	伸縮装置	4800000	46		
CO	40	床組構	620500	13	○	○	ST	17	支承	5000000	13		
CO	27	床組構	637500	13	○	○	ST	17	耐震性	5000000	14		
CO	18	橋面舗装	734400	17	○	○	ST	1	伸縮装置	5400000	27		
CO	23	床組構	739500	13	○	○	CO	18	伸縮装置	5400000	23		
CO	28	床組構	765000	13	○	○	CO	33	伸縮装置	5400000	23		
ST	13	塗装	900022.2	8	○	○	CO	26	支承	6000000	13		
ST	3	橋面舗装	1103850	34	○	○	CO	26	耐震性	6000000	17		
CO	37	安定材質	1305000	32	○	○	CO	30	支承	6000000	13		
ST	12	塗装	1309440	8	○	○	CO	30	耐震性	6000000	17		
ST	14	伸縮装置	1350000	27	○	○	ST	12	床版	6200000	22		
CO	37	伸縮装置	1350000	46	○	○	ST	14	地覆高欄	6496000	21		
ST	4	橋面舗装	1396500	17	○	○	CO	30	安定材質	6786000	32		
ST	8	塗装	1484700	8	○	○	CO	31	地覆高欄	7936000	23		
ST	12	伸縮装置	1500000	27	○	○	CO	28	支承	8000000	13		
CO	34	伸縮装置	1500000	23	○	○	CO	28	耐震性	8000000	17		
CO	39	伸縮装置	1500000	46	○	○	CO	31	耐震性	8000000	17		
ST	2	塗装	1584792	16	○	○	ST	6	耐震性	9000000	14		
CO	36	伸縮装置	1590000	46	○	○	ST	5	安定材質	9541000	18		
CO	37	地覆高欄	1600000	46	○	○	ST	15	床版	10540000	22		
ST	3	伸縮装置	1650000	27	○	○	CO	33	地覆高欄	10657600	23		
ST	13	伸縮装置	1650000	27	○	○	ST	3	安定材質	10670550	18		
CO	36	地覆高欄	1680000	23	○	○	ST	3	地覆高欄	10704000	21		
CO	21	伸縮装置	1710000	23	○	○	ST	2	支承	12000000	13		
CO	42	伸縮装置	1800000	23	○	○	ST	2	耐震性	12000000	14		
ST	15	伸縮装置	1860000	27	○	○	CO	19	支承	12000000	26		
CO	35	伸縮装置	1950000	46	○	○	CO	19	耐震性	12000000	34		
CO	41	伸縮装置	1950000	46	○	○	ST	2	床版	15007500	44		
CO	26	伸縮装置	2025000	23	○	○	ST	7	安定材質	15834000	18		
CO	38	伸縮装置	2100000	23	○	○	ST	14	床版	18270000	22		
CO	40	伸縮装置	2190000	23	○	○	ST	4	地覆高欄	21280000	21		
CO	24	伸縮装置	2250000	23	○	○	ST	6	床版	31350000	22		
CO	25	伸縮装置	2565000	23	○	○	ST	4	床版	46550000	22		
ST	4	伸縮装置	2825000	27	○	○	ST	7	床版	54600000	22		
ST	13	地覆高欄	2720000	21			CO	22	耐震性	63000000	17		
CO	20	地覆高欄	2720000	23			ST	10	塗装	86485313.4	8		
ST	17	伸縮装置	2820000	27	○		CO	22	安定材質	90045000	32		
ST	3	耐震性	3000000	14			ST	1	床版	131100000	44		
ST	9	耐震性	3000000	14			CO	22	床版	310500000	30		
ST	12	支承	3000000	13			ST	10	床版	386650000	22		

ただし、 F_a は投資効率関数、 F_b は無次元化関数を表わしている。

表- 2.18 関数による補修部位選択の違い-予算5千万円の評価値増加量降順-

橋種	橋架番号	部位	補修費用 単位:円	評価値増加量	目的関数		橋種	橋架番号	部位	補修費用 単位:円	評価値増加量	目的関数	
					F_a	F_b						F_a	F_b
CO	20	伸縮装置	4800000	46			ST	13	橋面舗装	280500	17	○	○
CO	24	地覆高欄	3280000	46	○	○	ST	15	橋面舗装	316200	17	○	○
CO	35	伸縮装置	1950000	46	○	○	ST	17	橋面舗装	174840	17	○	○
CO	36	伸縮装置	1590000	46	○	○	CO	18	橋面舗装	734400	17	○	○
CO	37	地覆高欄	1800000	46	○	○	CO	20	橋面舗装	408000	17	○	○
CO	37	伸縮装置	1350000	46	○	○	CO	22	耐震性	63000000	17		
CO	39	伸縮装置	1500000	46	○	○	CO	25	橋面舗装	355680	17	○	○
CO	41	伸縮装置	1950000	46	○	○	CO	26	橋面舗装	257850	17	○	○
ST	1	床版	131100000	44			CO	26	耐震性	6000000	17		
ST	2	床版	15007500	44			CO	28	橋面舗装	459000	17	○	○
ST	3	橋面舗装	1103850	34	○	○	CO	28	耐震性	8000000	17		
ST	14	橋面舗装	548100	34	○	○	CO	29	耐震性	3000000	17		
ST	16	橋面舗装	360000	34	○	○	CO	30	耐震性	6000000	17		
CO	19	耐震性	12000000	34			CO	31	耐震性	8000000	17		
CO	27	橋面舗装	382500	34	○	○	CO	34	橋面舗装	210000	17	○	○
CO	37	橋面舗装	135000	34	○	○	CO	34	耐震性	3000000	17		
CO	43	橋面舗装	273600	34	○	○	CO	35	耐震性	4000000	17		
CO	22	安定材質	90045000	32			CO	36	橋面舗装	168950	17	○	○
CO	30	安定材質	6786000	32			CO	36	耐震性	3000000	17		
CO	37	安定材質	1305000	32	○	○	CO	37	耐震性	3000000	17		
CO	22	床版	310500000	30			CO	38	耐震性	4000000	17		
ST	1	伸縮装置	5400000	27			CO	39	橋面舗装	150000	17	○	○
ST	2	伸縮装置	4140000	27			CO	39	耐震性	3000000	17		
ST	3	伸縮装置	1850000	27	○	○	CO	41	耐震性	4000000	17		
ST	4	伸縮装置	2025000	27			CO	42	橋面舗装	342000	17	○	○
ST	6	伸縮装置	3300000	27			ST	2	塗装	1584792	16		
ST	12	伸縮装置	1500000	27	○	○	CO	20	排水装置	170000	16	○	○
ST	13	伸縮装置	1650000	27	○	○	ST	2	排水装置	217500	14	○	○
ST	14	伸縮装置	1350000	27	○	○	ST	2	耐震性	12000000	14		
ST	15	伸縮装置	1860000	27	○	○	ST	3	耐震性	3000000	14		
ST	17	伸縮装置	2820000	27	○	○	ST	5	耐震性	4000000	14		
CO	19	支承	12000000	26			ST	6	耐震性	9000000	14		
CO	18	伸縮装置	5400000	23			ST	9	耐震性	3000000	14		
CO	20	地覆高欄	2720000	23			ST	12	耐震性	3000000	14		
CO	21	伸縮装置	1710000	23	○	○	ST	14	耐震性	3000000	14		
CO	24	伸縮装置	2250000	23			ST	17	耐震性	5000000	14		
CO	25	地覆高欄	3328000	23			ST	1	床組構	510000	13	○	○
CO	25	伸縮装置	2565000	23			ST	2	支承	12000000	13		
CO	26	地覆高欄	3066000	23			ST	11	支承	4000000	13		
CO	26	伸縮装置	2025000	23	○	○	ST	12	床組構	425000	13	○	○
CO	31	地覆高欄	7938000	23			ST	12	支承	3000000	13	○	○
CO	33	地覆高欄	10657600	23			ST	17	支承	5000000	13		
CO	33	伸縮装置	5400000	23			CO	23	床組構	739500	13	○	○
CO	34	伸縮装置	1500000	23	○	○	CO	26	支承	6000000	13		
CO	36	地覆高欄	1680000	23	○	○	CO	27	床組構	637500	13	○	○
CO	38	伸縮装置	2100000	23	○	○	CO	28	床組構	765000	13	○	○
CO	40	伸縮装置	2190000	23			CO	28	支承	8000000	13		
CO	42	伸縮装置	1800000	23	○	○	CO	29	支承	3000000	13		
ST	4	床版	46550000	22			CO	30	支承	6000000	13		
ST	6	床版	31350000	22			CO	34	床組構	425000	13	○	○
ST	7	床版	54600000	22			CO	35	支承	4000000	13		
ST	10	床版	386650000	22			CO	36	支承	3000000	13		
ST	12	床版	6200000	22			CO	37	支承	3000000	13		
ST	14	床版	18270000	22			CO	38	床組構	595000	13	○	○
ST	15	床版	10540000	22			CO	39	床組構	425000	13	○	○
ST	2	地覆高欄	3480000	21			CO	39	支承	3000000	13		
ST	3	地覆高欄	10704000	21			CO	40	床組構	620500	13	○	○
ST	4	地覆高欄	21280000	21			CO	40	支承	4000000	13		
ST	13	地覆高欄	2720000	21			CO	41	支承	4000000	13		
ST	14	地覆高欄	6496000	21			ST	8	塗装	1484700	8		
ST	16	地覆高欄	3200000	21			ST	10	塗装	86485313.4	8		
ST	3	安定材質	10670550	18			ST	12	塗装	1309440	8		
ST	5	安定材質	9541000	18			ST	13	塗装	900022.2	8		○
ST	7	安定材質	15834000	18			ST	14	塗装	3727080	8		
ST	4	橋面舗装	1396500	17		○	CO	18	排水装置	272000	8	○	○
ST	11	橋面舗装	558000	17	○	○	CO	40	排水装置	117000	8	○	○
ST	12	橋面舗装	186000	17	○	○	ST	14	排水装置	406000	7	○	○

表- 2.19 関数による補修部位選択の違い-予算5千万円の投資効率降順-

橋種	橋梁番号	部位 単位:円	投資効率	目的関数		橋種	橋梁番号	部位 単位:円	投資効率	目的関数	
				F _a	F _b					F _a	F _b
CO	37	橋面舗装	0.000251852	○	○	ST	16	地覆高欄	6.5625E-06		
CO	43	橋面舗装	0.000124269	○	○	ST	2	伸縮装置	6.52174E-06		
CO	39	橋面舗装	0.000113333	○	○	ST	12	塗装	6.10948E-06		
CO	36	橋面舗装	0.000101827	○	○	ST	2	地覆高欄	6.03448E-06		
ST	17	橋面舗装	9.72318E-05	○	○	CO	29	耐震性	5.66667E-06		
ST	16	橋面舗装	9.44444E-05	○	○	CO	34	耐震性	5.66667E-06		
CO	20	排水装置	9.41176E-05	○	○	CO	36	耐震性	5.66667E-06		
ST	12	橋面舗装	9.13978E-05	○	○	CO	37	耐震性	5.66667E-06		
CO	27	橋面舗装	8.88889E-05	○	○	CO	39	耐震性	5.66667E-06		
CO	34	橋面舗装	8.09524E-05	○	○	ST	8	塗装	5.38829E-06		
CO	40	排水装置	6.83761E-05	○	○	ST	1	伸縮装置	0.000005		
CO	26	橋面舗装	6.59298E-05	○	○	CO	30	安定材質	4.71559E-06		
ST	2	排水装置	6.43678E-05	○	○	ST	3	耐震性	4.66667E-06		
ST	14	橋面舗装	6.20325E-05	○	○	ST	9	耐震性	4.66667E-06		
ST	13	橋面舗装	6.06061E-05	○	○	ST	12	耐震性	4.66667E-06		
ST	15	橋面舗装	5.37634E-05	○	○	ST	14	耐震性	4.66667E-06		
CO	42	橋面舗装	4.97076E-05	○	○	ST	12	支承	4.33333E-06		
CO	25	橋面舗装	4.77958E-05	○	○	CO	29	支承	4.33333E-06		
CO	20	橋面舗装	4.16667E-05	○	○	CO	36	支承	4.33333E-06		
CO	28	橋面舗装	3.7037E-05	○	○	CO	37	支承	4.33333E-06		
CO	37	伸縮装置	3.40741E-05	○	○	CO	39	支承	4.33333E-06		
ST	3	橋面舗装	3.08013E-05	○	○	CO	18	伸縮装置	4.25926E-06		
CO	39	伸縮装置	3.06667E-05	○	○	CO	33	伸縮装置	4.25926E-06		
ST	12	床組構	3.05882E-05	○	○	CO	35	耐震性	0.00000425		
CO	34	床組構	3.05882E-05	○	○	CO	38	耐震性	0.00000425		
CO	39	床組構	3.05882E-05	○	○	CO	41	耐震性	0.00000425		
ST	11	橋面舗装	3.04659E-05	○	○	ST	12	床版	3.54839E-06		
CO	18	排水装置	2.94118E-05	○	○	ST	5	耐震性	0.0000035		
CO	38	伸縮装置	2.89308E-05	○	○	ST	11	支承	0.00000325		
CO	37	地覆高欄	0.00002875	○	○	CO	35	支承	0.00000325		
ST	1	床組構	2.54902E-05	○	○	CO	40	支承	0.00000325		
CO	37	安定材質	2.45211E-05	○	○	CO	41	支承	0.00000325		
CO	35	伸縮装置	2.35897E-05	○	○	ST	14	地覆高欄	3.23276E-06		
CO	41	伸縮装置	2.35897E-05	○	○	ST	2	床版	2.93187E-06		
CO	18	橋面舗装	2.31481E-05	○	○	CO	31	地覆高欄	2.89819E-06		
CO	38	床組構	2.18487E-05	○	○	CO	19	耐震性	2.83333E-06		
CO	40	床組構	2.09508E-05	○	○	CO	26	耐震性	2.83333E-06		
CO	27	床組構	2.03922E-05	○	○	CO	30	耐震性	2.83333E-06		
ST	14	伸縮装置	0.00002	○	○	ST	17	耐震性	0.0000028		
ST	12	伸縮装置	0.000018	○	○	ST	17	支承	0.0000028		
CO	23	床組構	1.75794E-05	○	○	CO	19	支承	2.16667E-06		
ST	14	排水装置	1.72414E-05	○	○	CO	26	支承	2.16667E-06		
CO	28	床組構	1.69935E-05	○	○	CO	30	支承	2.16667E-06		
ST	3	伸縮装置	1.63636E-05	○	○	CO	33	地覆高欄	2.15808E-06		
ST	13	伸縮装置	1.63636E-05	○	○	ST	14	塗装	2.14645E-06		
CO	34	伸縮装置	1.53333E-05	○	○	CO	28	耐震性	0.000002125		
ST	15	伸縮装置	1.45161E-05	○	○	CO	31	耐震性	0.000002125		
CO	24	地覆高欄	1.40244E-05	○	○	ST	15	床版	2.08729E-06		
CO	36	地覆高欄	1.36905E-05	○	○	ST	3	地覆高欄	1.96188E-06		
CO	21	伸縮装置	1.34503E-05	○	○	ST	5	安定材質	1.88659E-06		
CO	42	伸縮装置	1.27778E-05	○	○	ST	3	安定材質	1.88689E-06		
ST	4	橋面舗装	1.21733E-05	○	○	CO	28	支承	0.000001825		
CO	26	伸縮装置	1.1358E-05	○	○	ST	6	耐震性	1.55556E-06		
CO	38	伸縮装置	1.09524E-05	○	○	ST	14	床版	1.20416E-06		
CO	40	伸縮装置	1.05023E-05	○	○	ST	2	耐震性	1.16667E-06		
ST	4	伸縮装置	1.02857E-05	○	○	ST	7	安定材質	1.13679E-06		
CO	24	伸縮装置	1.02222E-05	○	○	ST	2	支承	1.08333E-06		
ST	2	塗装	1.0096E-05	○	○	ST	4	地覆高欄	9.86842E-07		
CO	20	伸縮装置	9.58333E-06	○	○	ST	6	床版	7.01754E-07		
ST	17	伸縮装置	9.57447E-06	○	○	ST	4	床版	4.7261E-07		
CO	25	伸縮装置	8.96686E-06	○	○	ST	7	床版	4.0293E-07		
ST	13	塗装	8.88867E-06	○	○	CO	22	安定材質	3.55378E-07		
CO	20	地覆高欄	8.45588E-06	○	○	ST	1	床版	3.35622E-07		
ST	6	伸縮装置	8.18182E-06	○	○	CO	22	耐震性	2.69841E-07		
ST	13	地覆高欄	7.72059E-06	○	○	CO	22	床版	9.66184E-08		
CO	26	地覆高欄	7.52618E-06	○	○	ST	10	塗装	9.25013E-08		
CO	25	地覆高欄	6.91106E-06	○	○	ST	10	床版	5.6899E-08		

これら、3つの表を比較してみると選択された補修部位にもほとんど違いが見られないことが分かる。全体的な傾向として、投資効率の高い部位から順に、そして補修費用が低額な部位を補修部位として選択している。若干ではあるが、投資効率関数が投資効率に重きを置いた補修部位の選択を行っているとも言える。

これまでの結果を総合的に判断して、以下の理由から以後の足場費用の考慮・スケジューリングに無次元化関数を用いることとする。

- どちらの関数も補修部位の選択には同等な効果をあげている。
- 無次元化関数を用いても投資効率を十分に考慮した補修部位選択が行われている。
- 補正值を設定するにあたって、投資効率関数では予算との相関がわかりにくいため最適な補正值 (δ) の設定が困難であるが、無次元化関数においては予算と補正值 (ζ) の相関がわかりやすく予算によって補正值の設定が可能である。

2.8.5 足場の考慮

これまで、橋梁の補修費用を算出するにあたって、橋梁各部位の補修費用のみを考慮してきたが、ここではそれに加えて補修における足場の費用についての考慮した橋梁補修計画案の作成を行う。足場の種類・費用についてはすでに第2章において述べているので、ここでは省略する。参考までに、43橋・134部位をすべて補修したとして補修費用を算出した金額を表-2.20に示す。なお、総補修費用とは橋梁各部位の補修費用と足場の費用を加えた金額である。

足場の費用を考慮することによって、橋梁補修部位選択にどのような変化が現れるのが比較するために、無次元化関数を用いて足場の費用を考慮した場合と考慮しない場合で、予算を5千万円・1億円・2億円としてそれぞれ10回ずつ解析を行った。

表- 2.20 補修費用

総補修費用	17億8220円
足場費用	1億2613円
足場費用の占める総補修費用に対する割合	7.08%

選択補修部位数の比較

図-2.21 と図-2.22 は、10回ずつ行った解析の中でそれぞれ最も評価値増加量の高い解析の結果を選び、各橋梁部位において要補修となっている部位のうち何%の部位が補修部位として選択されたのかを示したグラフである。参考までに、要補修部位数を点検項目別に図-2.23 補修部位として選択された部位数表-2.21 のとおりである。

表- 2.21 選択補修部位数

予算(円)	足場込み	足場抜き
5千万	52	55
1億	70	73
2億	95	97

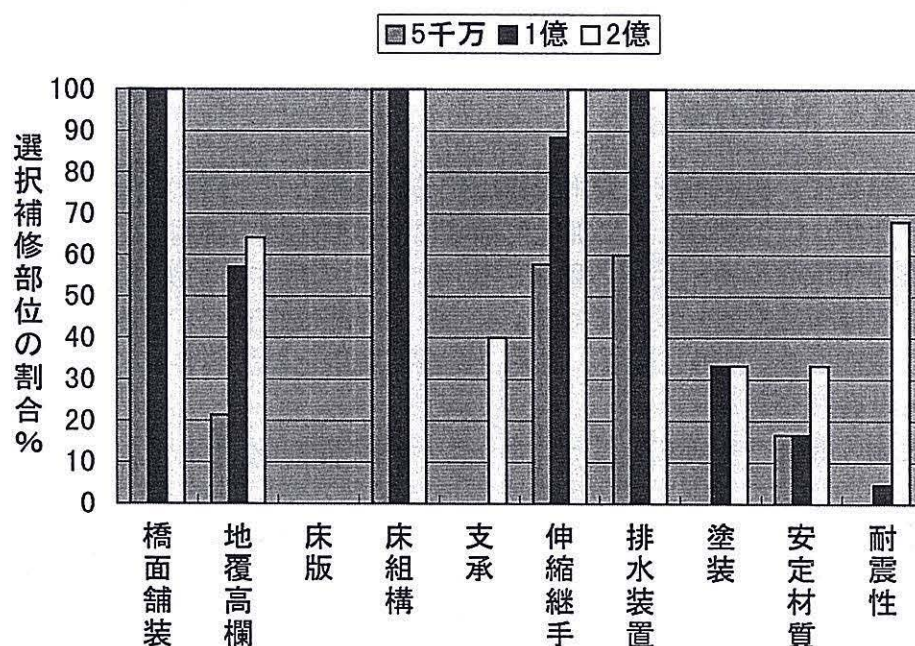


図- 2.21 足場費用を含んだ場合の補修部位の割合

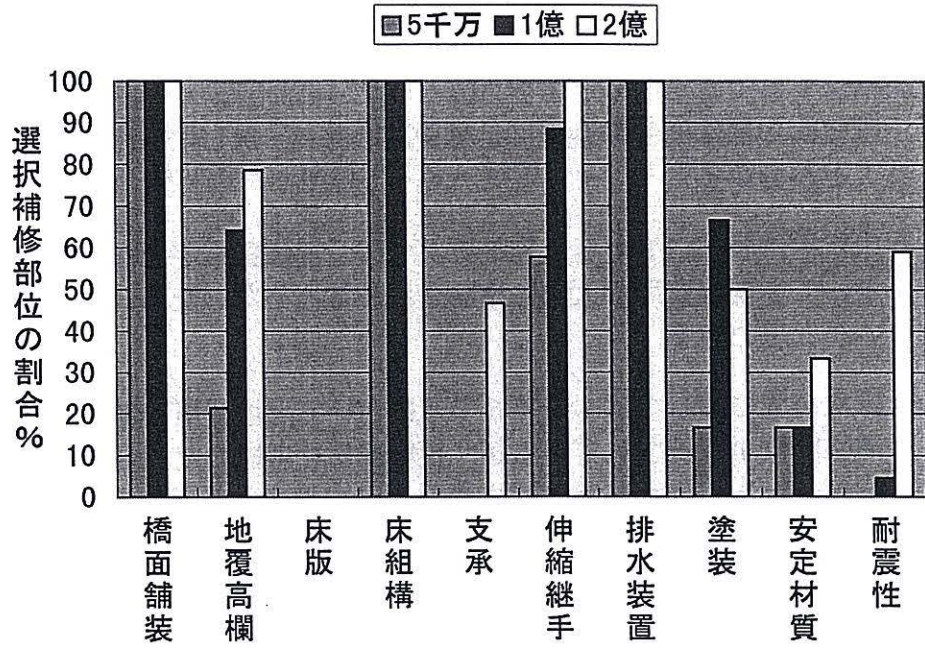


図- 2.22 足場費用を抜いた場合の補修部位の割合

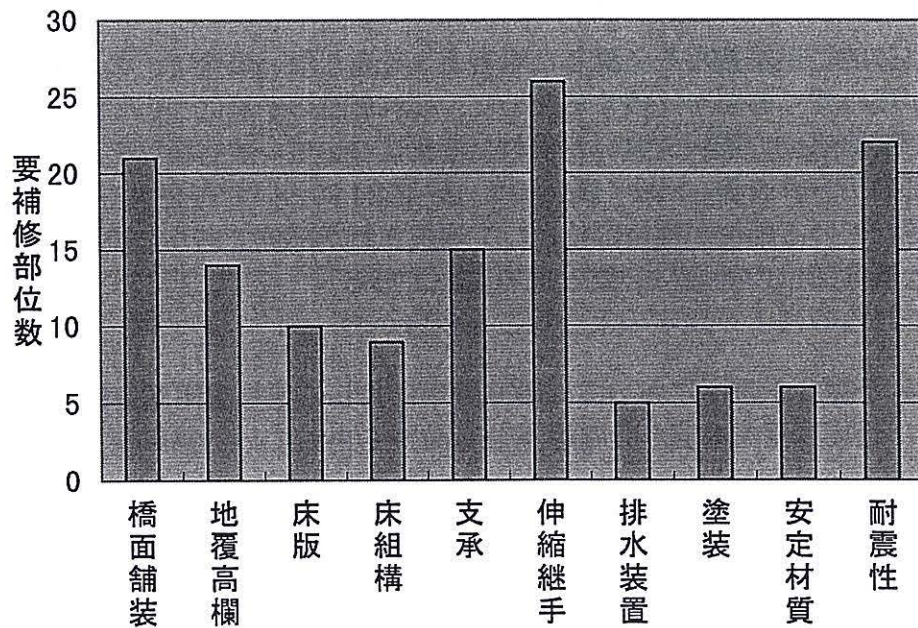


図- 2.23 各点検項目の要補修部位数

これらのグラフから以下のような考察を行った。

- 補修するにあたって足場を必要としない橋面舗装・伸縮継手の2部位と、安定材質・床組構の補修部位の選択は足場費用のあり・なしに関わらず同じ結果となっていることが分かる。
- 床組構は表-2.17からも分かるように、補修費用が低額であり足場の費用を加えたとしても補修費用が大幅に増額されることもないため、同じ結果になったのではないかとと思われる。
- 排水装置に関しては、予算5千万において足場費用を考慮しない場合、すべての要補修部位が選択されているが、足場費用を考慮した場合選択されない橋梁があることが分かる。排水装置は補修費用と足場の費用が単価5千円/m²と同じであり、足場の費用を考慮したとすると補修費用が2倍に跳ね上がってしまうためと思われる。
- 塗装に関しても補修費用が4千円/m²に対して足場費用が2.5千円/m²であり、足場費用を考慮した場合補修費用が1.5倍以上に増加してしまうため、補修部位として選択されにくくなっているのではないかとと思われる。

選択された補修部位の比較

具体的に足場費用を考慮した場合と、考慮しない場合でどのような補修部位の選択の違いが現れてくるのか比較するために、それぞれの予算について足場込みと足場抜きの場合とで補修部位の選択で違いが現れた橋梁について、該当橋梁の要補修となっている部位すべてのデータを表-2.22・表-2.23・表-2.24において表記する。なお、それぞれ“○”が記されている部位が補修部位として選択された部位であり、補修費用とは足場の費用を除いた橋梁部位の補修に必要となる金額である。

これらの表より以下のようなことが分かる。

- 予算5千万円においては補修費用に対して足場費用が高額である排水装置・塗装が足場費用を考慮した場合補修部位からもれている橋梁がある。
- 予算が1億円・2億円になると要補修となっている排水装置は、足場費用を考慮した場合でもすべて補修部位として選択されており、塗装と地覆高欄が補修部位として選択されていない橋梁がある。
- 予算2億円において耐震性が足場費用を考慮した場合2つ多く選択されているが、これは足場費用を加えると地覆高欄に比べ補修費用が低額になっているためと思われる。

これらの結果を総合的に評価して、足場費用を考慮することにより補修費用の単価が上がり選択される補修部位の数は減少しており、足場費用が高額である部位は選択されにくくなっていることが分かる。また、足場を共有できる部位を第2章に示しているが、3種類の設定予算では補修部位選択に影響がなく、相関のない結果となった。

表- 2.22 予算5千万円における補修部位選択の違い

橋種	橋梁	部位	補修費用(円)	足場費用(円)	評価値増加量	足場込み	足場抜き
ST	4	橋面舗装	1396500	0	17	○	○
ST	4	地覆高欄	21280000	2660000	21		
ST	4	床版	46550000	2327500	22		
ST	4	伸縮装置	2625000	0	27		○
ST	13	橋面舗装	280500	0	17	○	○
ST	13	地覆高欄	2720000	340000	21		
ST	13	伸縮装置	1650000	0	27	○	○
ST	13	塗装	900022	562513	8		○
ST	14	橋面舗装	548100	0	34	○	○
ST	14	地覆高欄	6496000	812000	21		
ST	14	床版	18270000	913500	22		
ST	14	伸縮装置	1350000	0	27	○	○
ST	14	排水装置	406000	406000	7		○
ST	14	塗装	3727080	2329425	8		
ST	14	耐震性	3000000	27000	14		
CO	18	橋面舗装	734400	0	17	○	○
CO	18	伸縮装置	5400000	0	23		
CO	18	排水装置	272000	272000	8		○
CO	26	橋面舗装	257850	0	17	○	○
CO	26	地覆高欄	3056000	382000	23		
CO	26	支承	6000000	40500	13		
CO	26	伸縮装置	2025000	0	23	○	
CO	26	耐震性	6000000	40500	17		
CO	40	床組構	620500	36500	13	○	○
CO	40	支承	4000000	43800	13		
CO	40	伸縮装置	2190000	0	23	○	
CO	40	排水装置	117000	117000	8	○	○
CO	42	橋面舗装	342000	0	17	○	○
CO	42	伸縮装置	1800000	0	23		○

表- 2.23 予算1億円における補修部位選択の違い

橋種	橋梁	部位	補修費用(円)	足場費用(円)	評価値増加量	足場込み	足場抜き
ST	8	塗装	1484700	927937	8		○
ST	14	橋面舗装	548100	0	34	○	○
ST	14	地覆高欄	6496000	812000	21		
ST	14	床版	18270000	913500	22		
ST	14	伸縮装置	1350000	0	27	○	○
ST	14	排水装置	406000	406000	7	○	○
ST	14	塗装	3727080	2329425	8		○
ST	14	耐震性	3000000	27000	14		
ST	16	橋面舗装	360000	0	34	○	○
ST	16	地覆高欄	3200000	400000	21		○
CO	36	橋面舗装	166950	0	17	○	○
CO	36	地覆高欄	1680000	210000	23	○	○
CO	36	支承	3000000	31800	13		
CO	36	伸縮装置	1590000	0	46	○	○
CO	36	耐震性	3000000	31800	17	○	
CO	37	橋面舗装	135000	0	34	○	○
CO	37	地覆高欄	1600000	200000	46	○	○
CO	37	支承	3000000	27000	13		
CO	37	伸縮装置	1350000	0	46	○	○
CO	37	安定材質	1305000	135000	32	○	○
CO	37	耐震性	3000000	27000	17		○

表- 2.24 予算2億円における補修部位選択の違い

橋種	橋梁	部位	補修費用(円)	足場費用(円)	評価値増加量	足場込み	足場抜き
ST	12	橋面舗装	186000	0	17	○	○
ST	12	床版	6200000	310000	22		
ST	12	床組構	425000	25000	13	○	○
ST	12	支承	3000000	30000	13	○	○
ST	12	伸縮装置	1500000	0	27	○	○
ST	12	塗装	1309440	818400	8		○
ST	12	耐震性	3000000	30000	14	○	○
ST	14	橋面舗装	548100	0	34	○	○
ST	14	地覆高欄	6496000	812000	21		○
ST	14	床版	18270000	913500	22		
ST	14	伸縮装置	1350000	0	27	○	○
ST	14	排水装置	406000	406000	7	○	○
ST	14	塗装	3727080	2329425	8		
ST	14	耐震性	3000000	27000	14	○	○
CO	26	橋面舗装	257850	0	17	○	○
CO	26	地覆高欄	3056000	382000	23	○	○
CO	26	支承	6000000	40500	13		
CO	26	伸縮装置	2025000	0	23	○	○
CO	26	耐震性	6000000	40500	17	○	
CO	30	支承	6000000	54000	13		
CO	30	安定材質	6786000	702000	32	○	○
CO	30	耐震性	6000000	54000	17	○	
CO	31	地覆高欄	7936000	992000	23		○
CO	31	耐震性	8000000	58500	17		
CO	40	床組構	620500	36500	13	○	○
CO	40	支承	4000000	43800	13		○
CO	40	伸縮装置	2190000	0	23	○	○
CO	40	排水装置	117000	117000	8	○	○

2.9 スケジューリング

2.9.1 GA への適用

スケジューリング問題に GA を適用すると、ナップサック問題を扱う場合と大きな違いが現れてくる。まず、遺伝子が 0, 1 で表されないこと。遺伝子にそれぞれ番号が振られているため交叉を行う場合、ナップサック問題のような手順での交叉が行えず、交叉方法に工夫が必要となってくる。これらの点を考慮しつつ GA への適用を行う。

初期線列の生成

GA へスケジューリング問題を適用するにあたって、まず初期線列の生成方法について述べる。コーディングの手順は以下のとおりである。なお、線列 (A) の下段に示されている” (1)-4” とは橋梁 (1) の 4 番目の部位のことを表しており、また、ここでは p は橋梁数を、 q は補修部位選択で選ばれた総補修部位数を、そして r とは人口数を表している。

- 1: 補修部位選択によって選択された部位へ橋梁 (1) から橋梁 (p) まで、通しナンバーを 1 から q まで順番に割り振る。
- 2: 第 2 章の図-2.8 のルールに従うようにして、橋梁別にランダムに左から順に補修部位を並べ替える。例えば、床組構の補修を行う前に床版の補修が行われるような並べ替えは行われないようにするわけである。
- 3: 補修部位数の割合に応じて 1 から p までの整数 i を乱数として発生させ、橋梁 (i) の補修部位を左から 1 部位取り出し、線列 (j) の左から順に遺伝子として割り振られた通しナンバーをいれる。これを、 q 回繰り返すことにより線列 (A) のような線列を持った個体が生成される。

以上の手順を j が r になるまで繰り返し、 r 個の個体を生成する。

橋梁 (1)	:	1	3	4	6	13	14	橋梁 (2)	:	2	6	7	13
通しナンバー		1	2	3	4	5	6			7	8	9	10
<div style="display: flex; justify-content: center; gap: 100px;"> <div style="text-align: center;"> \Downarrow </div> <div style="text-align: center;"> \Downarrow </div> </div>													
橋梁 (1)	:	4	13	3	6	14	1	橋梁 (2)	:	6	13	2	7
		3	5	2	4	6	1			8	10	7	9
<div style="display: flex; justify-content: center; gap: 100px;"> <div style="text-align: center;"> \Downarrow </div> </div>													
線列 (A)	:	3	5	8	2	10	7	4	6	1	9		
		(1)-4	(1)-13	(2)-6	(1)-3	(2)-13	(2)-2	(1)-6	(1)-14	(1)-1	(2)-7		

交叉と突然変異

スケジューリングにおいてGAを適用する場合、図-2.8のルールに従うようにして初期線列を生成しており、交叉・突然変異によって補修順序が壊れてしまうと補修計画案の作成に支障をきたす。突然変異を行うと補修順序のルールを破る個体が生成される可能性があるため、スケジューリングにおいては突然変異を行わないこととし、交叉についても補修部位選択に用いた交叉とは異なった補修順序が壊されない方法を用いる。なお、交叉点数は1点として行う。

- 1: 0から1の乱数を発生させ交叉率(0~1で設定している)より値が小さい場合に交叉を行う。
- 2: 0から遺伝子の数までの乱数を発生させ、線列の切断箇所を決定し、切断された線列の長い部分をそのまま次の世代に残すものとし、短い部分の線列を交叉を行う個体から受け継ぐ。以下に示す例においては↑の記されている4番目の遺伝子と5番目の遺伝子の部分で切断が行われ、線列の左側が次世代へそのまま受け継がれている。
- 3: 交叉を行う前の相手の線列を左端から順番にスキャンしていき、短い部分の線列を相手の線列の並びと同じ順番に並べ替える。個体(A)においては線列右側は154の順で並んでいるが、個体(B)においては451の順で並んでおり、個体(A)'においては個体(B)を受け継いで右側の線列を451に変える。個体(B)においても同様に交叉を行い、これによって次世代に新たな個体として個体(A)と個体(B)が残される。

どちらの個体も補修順序の条件を満たした個体であり、この方法を用いて交叉を行えば互いの補修順序を基に、補修順序の条件を満たした新しい個体を生成することが可能である。

個体(A)	:	7	2	3	6	↑	1	5	4	個体(B)	:	6	2	3	4	↑	7	5	1
								↓										↓	
個体(A)'	:	7	2	3	6		4	5	1	個体(B)'	:	6	2	3	4		7	1	5

GAの諸値

以上に述べてきたGAの設定に加え、GAの諸値を表-2.25に示す。

スケジューリングにおいては突然変異を行わないので、個体の種類が固定されないように人口数を補修部位選択に比べて2倍にした。

表- 2.25 スケジューリングにおける GA の諸値

集団数	: 1
人口数	: 200
エリート個体数	: 1
交叉	: 一点交叉
交叉率	: 0.6

なお、淘汰方法は補修部位選択と同様にエリートルールを用い、収束条件についても補修部位選択と同じ3条件を用いる。

2.9.2 スケジューリング問題の適用

補修工事の各事業への割り当て

本研究においては、橋梁補修計画案を作成するにあたって橋梁補修を実施する事業を考えながら行う。事業とは補修を行う系列を表しており、事業ごとに限られた日数(例えば1年間ならば365日)で割り当てられた補修工事を実施し、かつ年間予算内で最も効果をあげるような橋梁補修計画案の作成を目指す。

各事業への補修工事の割り当ては、事業数を n 、総補修部位数を m とすると、1から m 番目まで並んだ遺伝子を左側から順番に第1事業から第 n 事業まで順番に割り当てる方法とする。1番目の遺伝子から n 番目までの遺伝子を第1事業から第 n 事業まで割り当て、 $n+1$ 番目の遺伝子から $2 \times n$ 番目までの遺伝子を再び第1事業から順に第 n 事業まで割り当てる。これを、遺伝子 m 番目が割り当てられるまで行う。以下に、総補修部位数:10・事業数:3 の場合の補修部位割り当て方法の例を示す。

個体(A)	:	3	5	8	2	10	7	4	6	1	9
						↓					
第1事業	:	3	2	4	9						
第2事業	:	5	10	6							
第3事業	:	8	7	1							

スケジューリングを行うにあたって、以下のルールに従って行う。

- 1: 事業年数を p 年・1年当たりの工事実施日数を q 日・年間予算を E 円とすると、割り当てられた補修工事を順番に工期と予算の加算を行い、工事日数が q 日を越える工事は翌年に実施することとする。つまり、2年間に渡って行われるような工事実施の計画は行わず、各事業年内で工事を終了するように補修計画案の作成を行う。
- 2: n 事業の補修費用を割り当て順に加算して E 円をオーバーした場合は、予算を超えた工事以降に割り当てられている補修工事は翌年以降に持ち越すこととする。

このルールに従って、 p 年までスケジューリングすることによって数年の橋梁補修計画案の作成を行う。

目的関数の設定

スケジューリングにおいて、各個体の適応度の算出を行う目的関数を4種類設定する。目的関数を設定するにあたって無次元化関数式 (2.14) にスケジューリングにおいて考慮した項を加える形の関数とする。

目的関数

obj1

$$F_1 = \frac{\sum_{i=1}^n (e_{i2} - e_{i1})}{\zeta} - \frac{g}{E} + \frac{\sum_{j=1}^P \sum_{k=1}^Y (D - d_{jk})}{D} \Rightarrow \max \quad (2.15)$$

obj2

$$F_2 = \frac{\sum_{i=1}^n (e_{i2} - e_{i1})}{\zeta} - \frac{g}{E} - \frac{\sum_{i=1}^n b}{D \times n} \Rightarrow \max \quad (2.16)$$

obj3

$$F_3 = \frac{\sum_{i=1}^n (e_{i2} - e_{i1})}{\zeta} - \frac{g}{E} - \frac{\sum_{i=1}^n a}{D \times n} \Rightarrow \max \quad (2.17)$$

obj4

$$F_4 = \frac{\sum_{i=1}^n (e_{i2} - e_{i1})}{\zeta} - \frac{g}{E} - \frac{\sum_{i=1}^n (a + b)}{D \times n} \Rightarrow \max \quad (2.18)$$

制約条件

$$g = E - C \geq 0$$

- n : 橋梁数
- e_{i2} : 補修後の i 番目の橋梁の評価値
- e_{i1} : 補修前の i 番目の橋梁の評価値
- E : 予算
- C : 補修費用合計
- ζ : 補正值
- D : 一年間の工事実施日数
- Y : 事業年数
- P : 事業数
- d_{jk} : j 事業における k 事業年の最終工事終了日
- b : 同一橋梁のインターバル
- a : 同一足場のインターバル

obj1: 式 (2.15) の第3項は、橋梁の補修工事を行うにあたって各事業・各事業年毎に実施される工事が早く終了するような橋梁補修計画案の作成を目指すため加えた。ただ、第3項は補修部位選択において補修部位と選択された部位すべてを補修することが可能なスケジューリングが行われた場合のみ加算されるようにする。

obj2: 式 (2.16) の第3項で用いている同一橋梁のインターバル: b とは、事業ごとに割り当てられた補修工事のスケジューリングを行い作成された橋梁補修計画案を、橋梁単位で調べ、スケジューリングの初期線列の生成で記したように橋梁ごとの補修順序を基にし、 j 番目の補修部位の工事開始日から $j - 1$ 番目の補修部位の工事終了日を除算した値を各橋梁ごとに加算し、 n 橋分のインターバルを加算した値である。同一橋梁の工事が同時に行われているような場合、つまり $j - 1$ 番目と j 番目のインターバルがマイナスになるような場合にもそのまま加算し、インターバルが減少するように設定する。このようにインターバルが大きければ目的関数値が減少するように設定することによって、同一橋梁の補修工事が一時期に集中して実施されるような橋梁補修計画案の作成を目指す。

obj3: 第2章 補修足場の費用、表-2.6で記したように、補修するにあたり床版・床組構・排水装置・塗装は同一のつり足場を利用することができ、支承・耐震性は同一の張り出し足場の共有が可能である。そこで、式 (2.17) においては同一の足場を使用する部位についてはなるべくインターバルなく補修が行われるのが好ましいと考え、設定を行った。同一橋梁のインターバルの算出方法と同様に、各橋梁ごと同一足場を使用する補修部位のインターバルを加算した値が a である。

obj4: 式 (2.18) は, 第3項において同一橋梁と同一足場のインターバルを加えることにより, 式 (2.16) と式 (2.17) を両方兼ね備えたような関数とする. つまり, 同一橋梁は集中して補修を行い, 更にその中で同一足場の補修部位のインターバルが短くなるような橋梁補修計画案の作成を目指す.

なお, それぞれの第3項も D もしくは $D \times n$ で割ることによって無次元化を図っている.

2.9.3 解析結果の比較

スケジューリングを行うにあたって, 補修部位選択において予算2億円とし無次元化関数を用いて行った10回の解析の中で最も高い評価値増加量を示した解析において選択された補修部位を基に, 事業数3; 事業年数4; 1日当たりの工事実施時間を8時間; 工事実施日数を週5日間, 年間260日として解析を行う. つまり, 一年当たりの予算を5千万円と設定して橋梁補修計画案の作成を行う. 表-2.26に補修部位選択において作成された, スケジューリングで使用するデータを示す.

設定した4種類の目的関数を用いてそれぞれ5回ずつ解析を行った. 図-2.24は目的関数別の同一橋梁のインターバルを, 図-2.25には同一足場のインターバルを表したグラフである. なお, それぞれの値は5回行った解析の平均値である.

これらのグラフより, それぞれの目的関数の特徴がはっきりと現れていることが分かる. obj2は同一橋梁のインターバルが最も少ない値となり, 同一足場のインターバルについては, obj3が最も少ない値を示しており, この値から同一足場の橋梁部位はほとんどインターバルなく補修が実施されていることがわかる. obj4についてはどちらの値についても2番目に少ないインターバルを示しており, 目的にあった結果となっている. 特に同一橋梁のインターバルについてはobj2と大きな差のない値を示している.

表- 2.26 補修部位データ

橋種	橋梁番号	橋面舗装	地覆高欄	床版	床組構	主構	支承	伸縮継手	排水装置	塗装	洗掘変動	躯体変動	安定構造	安定材質	耐震性
ST	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0
ST	2	0	1	0	0	0	0	1	1	1	0	0	0	0	0
ST	3	1	0	0	0	0	0	1	0	0	0	0	0	0	1
ST	4	1	0	0	0	0	0	1	0	0	0	0	0	0	0
ST	5	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ST	6	0	0	0	0	0	0	1	0	0	0	0	0	0	0
ST	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ST	8	0	0	0	0	0	0	0	0	1	0	0	0	0	0
ST	9	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ST	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ST	11	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ST	12	1	0	1	1	0	0	1	0	1	0	0	0	0	1
ST	13	1	1	0	0	0	0	1	0	1	0	0	0	0	0
ST	14	1	0	0	0	0	0	1	1	0	0	0	0	0	1
ST	15	1	0	0	0	0	0	1	0	0	0	0	0	0	0
ST	16	1	1	0	0	0	0	0	0	0	0	0	0	0	0
ST	17	1	0	0	0	0	0	1	0	0	0	0	0	0	1
CO	18	1	0	0	0	0	0	1	1	0	0	0	0	0	0
CO	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CO	20	1	1	0	0	0	0	1	1	0	0	0	0	0	0
CO	21	0	0	0	0	0	0	1	0	0	0	0	0	0	0
CO	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CO	23	0	0	0	1	0	0	0	0	0	0	0	0	0	0
CO	24	0	1	0	0	0	0	1	0	0	0	0	0	0	0
CO	25	1	1	0	0	0	0	1	0	0	0	0	0	0	0
CO	26	1	1	0	0	0	0	1	0	0	0	0	0	0	0
CO	27	1	0	0	1	0	0	0	0	0	0	0	0	0	0
CO	28	1	0	0	1	0	0	0	0	0	0	0	0	0	0
CO	29	0	0	0	0	0	1	0	0	0	0	0	0	0	1
CO	30	0	0	0	0	0	0	0	0	0	0	0	0	1	0
CO	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CO	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CO	33	0	0	0	0	0	0	1	0	0	0	0	0	0	0
CO	34	1	0	0	1	0	0	1	0	0	0	0	0	0	1
CO	35	0	0	0	0	0	0	1	0	0	0	0	0	0	1
CO	36	1	1	0	0	0	1	1	0	0	0	0	0	0	1
CO	37	1	1	0	0	0	1	1	0	0	0	0	0	1	1
CO	38	0	0	0	1	0	0	1	0	0	0	0	0	0	1
CO	39	1	0	0	1	0	1	1	0	0	0	0	0	0	1
CO	40	0	0	0	1	0	1	1	1	0	0	0	0	0	0
CO	41	0	0	0	0	0	0	1	0	0	0	0	0	0	1
CO	42	1	0	0	0	0	0	1	0	0	0	0	0	0	0
CO	43	1	0	0	0	0	0	0	0	0	0	0	0	0	0

1:補修部位として選択された部位

0:補修部位として選択されなかった部位, もしくは補修の必要ない部位

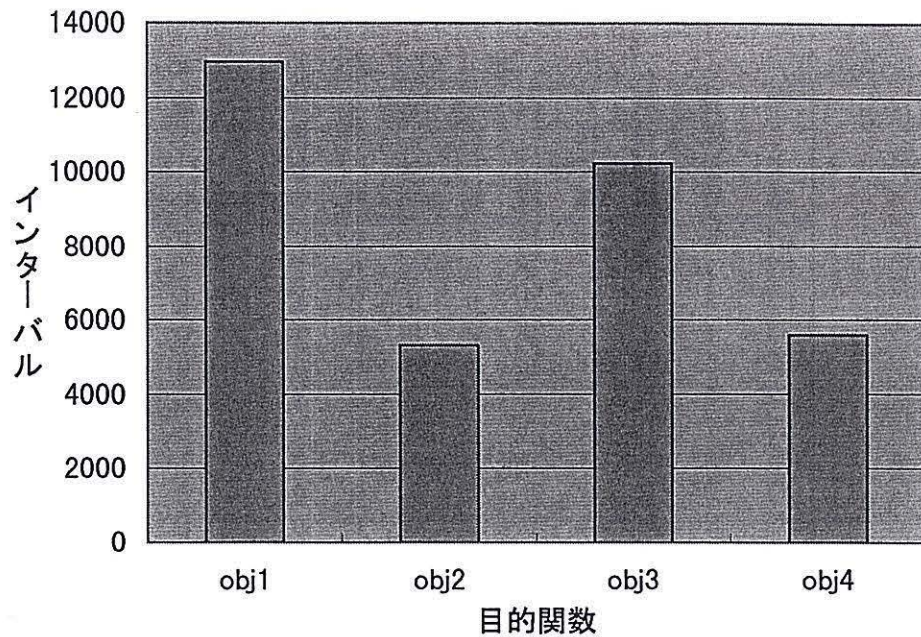


図- 2.24 同一橋梁のインターバル

事業ごとの結果の比較

実際にどのような橋梁補修計画案が作成されているか比較するために、それぞれの目的関数において最も目的関数値が高い結果となった解析の橋梁補修計画案の比較を行う。表-2.27 に各目的関数において採用した橋梁補修計画案のデータを記す。なお、補修部位選択で選択された補修部位数は96 部位である。

図-2.26・図-2.27・図-2.28・図-2.29 に事業ごとの橋梁補修計画案を、大まかに補修工事を実施している期間を補修実施期間、補修を実施していないを空白期間として表す。

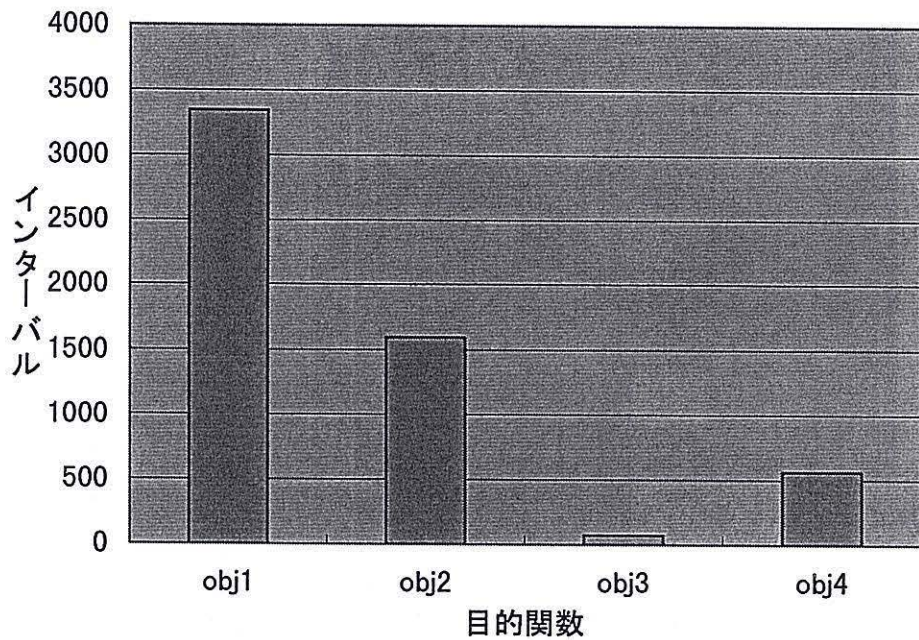


図- 2.25 同一足場のインターバル

表- 2.27 橋梁補修計画案の諸データ

目的関数		obj1	obj2	obj3	obj4
補修部位数		96	96	96	95
費用	1年目	49599	49759	49945	48866
	2年目	49912	49771	49815	49315
	3年目	49985	49958	49757	49835
	4年目	49932	49940	49910	49453
総費用		199428	199428	199428	197467

補修費用の単位は千円である

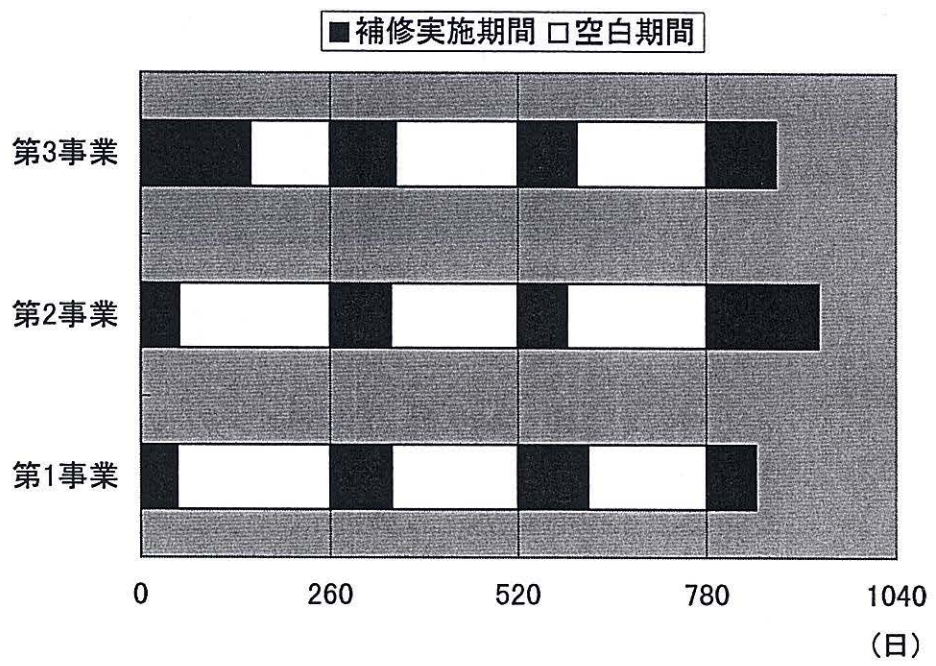


図- 2.26 obj1 における事業別の橋梁補修計画案

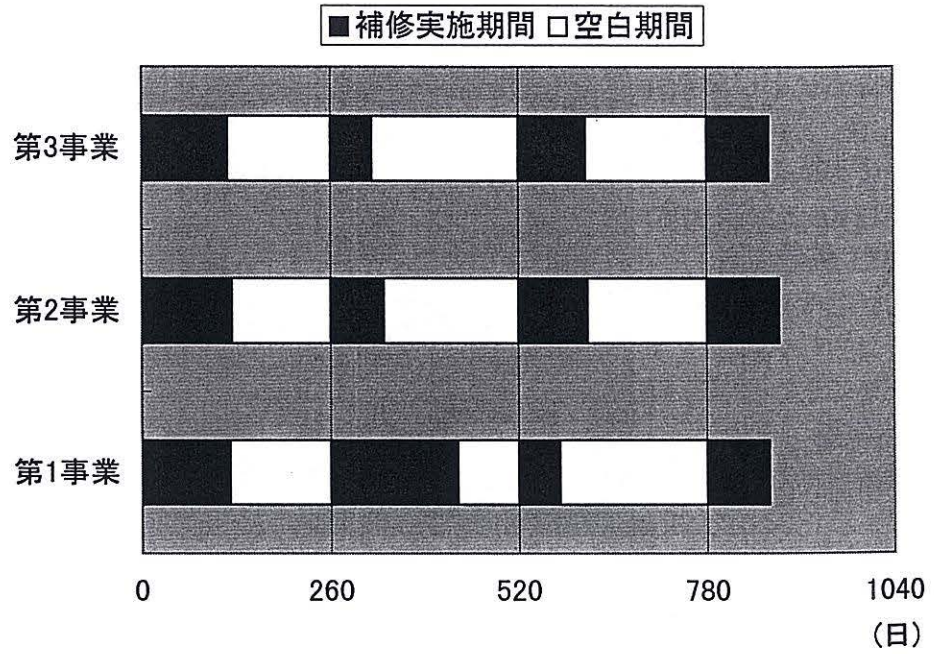


図- 2.27 obj2 における事業別の橋梁補修計画案

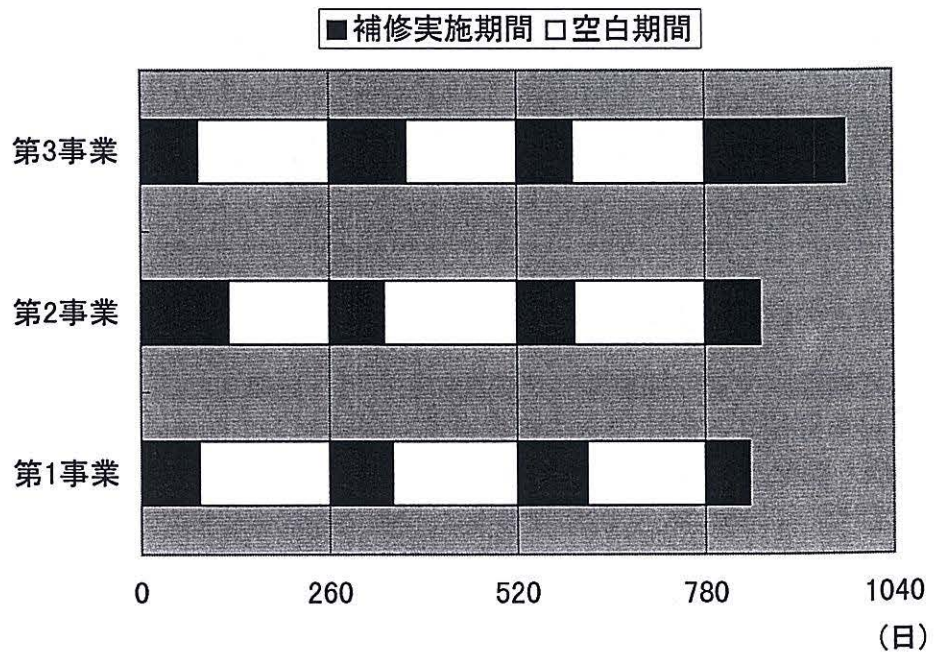


図- 2.28 obj3 における事業別の橋梁補修計画案

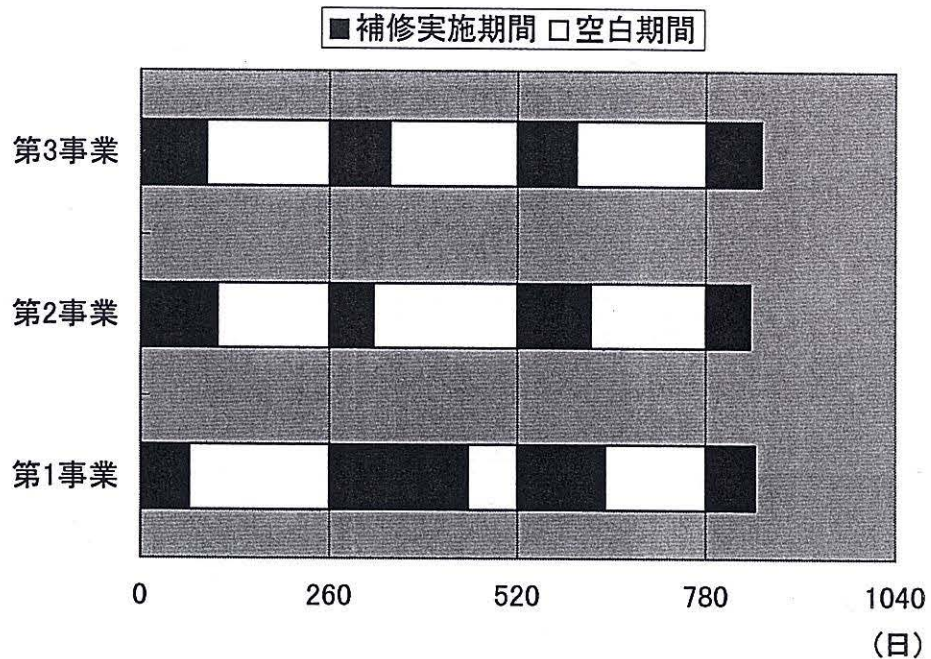


図- 2.29 obj4 における事業別の橋梁補修計画案

これらのグラフより，どの目的関数においても日程的に十分な余裕を持った補修計画案が作成されていることがわかる．それにもかかわらず，obj4 においてはST 橋・橋梁番号 12 番・塗装の部位が補修されていない．obj4 においては負荷条件が大きいため式 (2.18) の第 1 項，2 項に対して第 3 項の値が大きくなり，すべての部位の補修を行うよりも目的関数値が大きくなり適応度が高い個体として残されたのではないかとと思われる．

橋梁ごとの結果の比較

次に橋梁単位での補修計画案の比較を行う．比較する橋梁には，補修部位として選択されている部位が 5 部位以上の橋梁を選んだ．対象となった橋梁は，ST 橋・橋梁番号 12，CO 橋・橋梁番号 36，CO 橋・橋梁番号 37，CO 橋・橋梁 39 の 4 橋梁であり補修部位数はそれぞれ 6，5，6，5 部位である．各橋梁の補修計画を図-2.30・図-2.31・図-2.32・図-2.33 に示す．

これらの補修計画を見ると，すべての橋梁において obj1 に比べ，残りの関数を用いた結果はどれも同一橋梁のインターバルがすくない計画案となっていることがわかる．特に，obj3 を用いた橋梁番号 12 の計画案，そして obj4 を用いた橋梁番号 37 の補修計画案では，ほぼインターバルなしに一時期に補修が実施されている．しかし，obj2 は同一橋梁の補修部位を短期間で補修することを目的としているにもかかわらず，obj2 を用いて

作成された橋梁番号 12 と 37 の補修計画案においてはすべての補修が終了するのに 2 年以上かかっており、これらの結果を見る限り、obj2 を用いた計画案は十分な効果が現れているとは言い難い結果である。

同一足場の部位について比較すると、obj3 を用いることによる効果が十分に現れていることがわかる。支承と耐震性が補修部位となっているすべての橋梁は obj3 を用いることにより、2 つの部位の補修がインターバルなく行われており、橋梁番号 12 の補修が集中的に行われたのも同一足場である床版と床組構のインターバルを減少させようとした結果ではないかと思われる。

obj4 を用いた計画案を見ると同一橋梁のインターバルと同一足場のインターバルを減少させる効果がますます現れているのではないかと思われる。しかし、橋梁番号 36 の補修には長期間を要しており、橋梁番号 39 においては同一足場である支承と耐震性との工事の間隔が大きく開いているなど、まだまだ問題点の多い結果となった。

2.10 結論

橋梁維持管理支援システムは、橋梁点検台帳に点検データに基づいて、限られた予算内で最大限の効果をあげる橋梁補修計画案の作成を目指すシステムである。システムの構築がなされ、様々な改良が加えられてきたが、本研究においては更なる橋梁維持管理支援システムの改良を行った。主な改良点は以下に示す 3 点である。

- 1: 補修部位選択において目的関数として投資効率を考慮した関数と無次元化した関数を用いて解析を行い、結果の比較検討を行った。
- 2: 橋梁各部位の補修費用に加え足場の費用を考慮した補修計画案の作成を行った。
- 3: 橋梁の補修工期を考慮することにより、橋梁補修のスケジューリングを行った。

2 種類の関数を用いた解析の結果、どちらも同等の効果をあげていたが無次元化関数は投資効率を十分考慮した補修部位の選択を行っていることに加え、予算を考慮した補正值を設定できる点などから無次元化関数がより優れていると判断する。

足場を共有することによる補修部位選択への影響は確認できなかったが、足場費用を考慮することにより、より綿密な橋梁補修計画案の作成が可能になった。

スケジューリングにおいて、同一橋梁の補修部位を短期間で補修する計画案、そして同一足場の橋梁部位を短期間で補修する計画案の作成を目指し、その結果後者を十分に満たす補修計画案の作成は可能となったが、前者の目的を満たすのにはまだ不十分な結果であった。

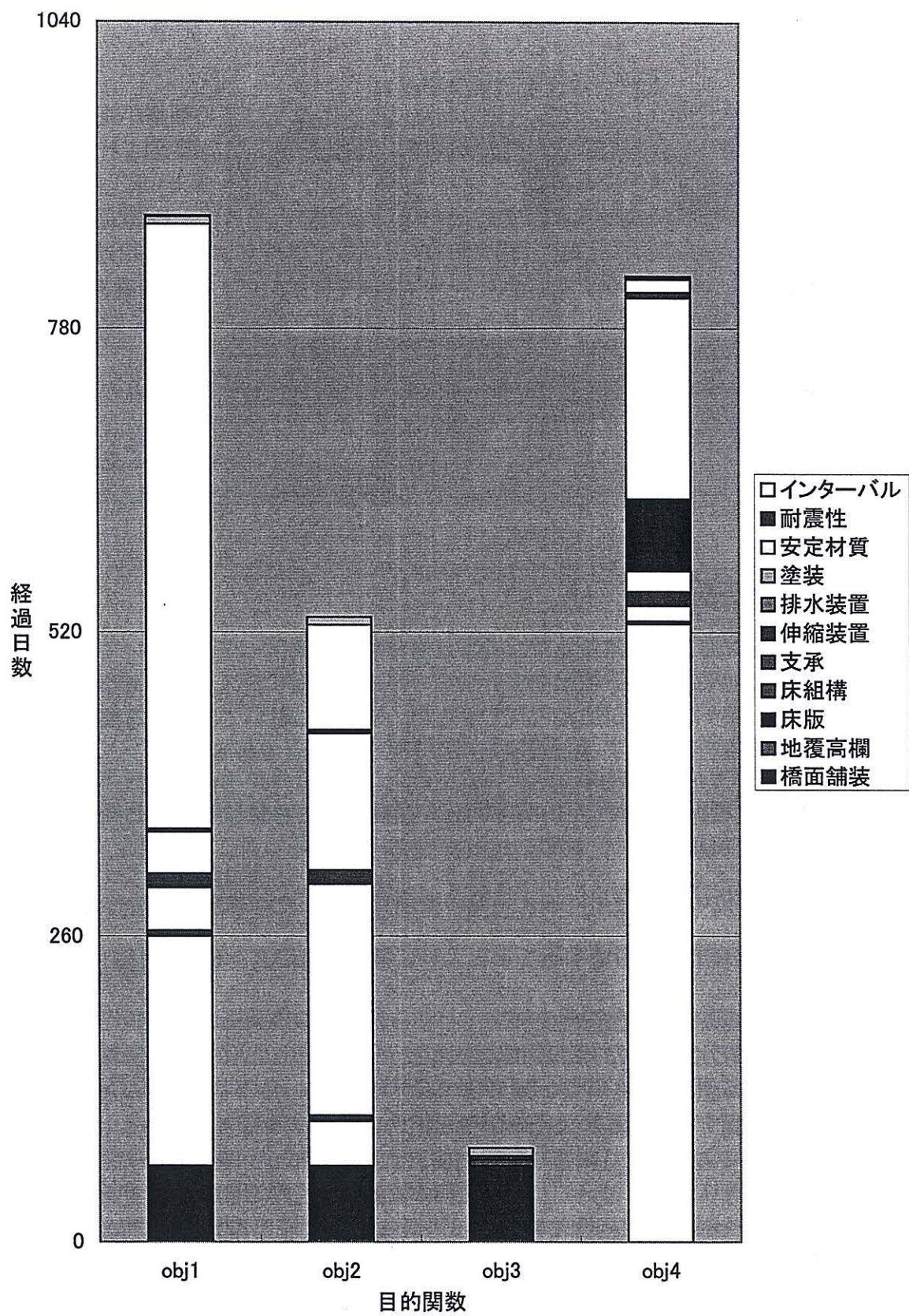


図- 2.30 ST 橋・橋梁番号 12 番の補修計画

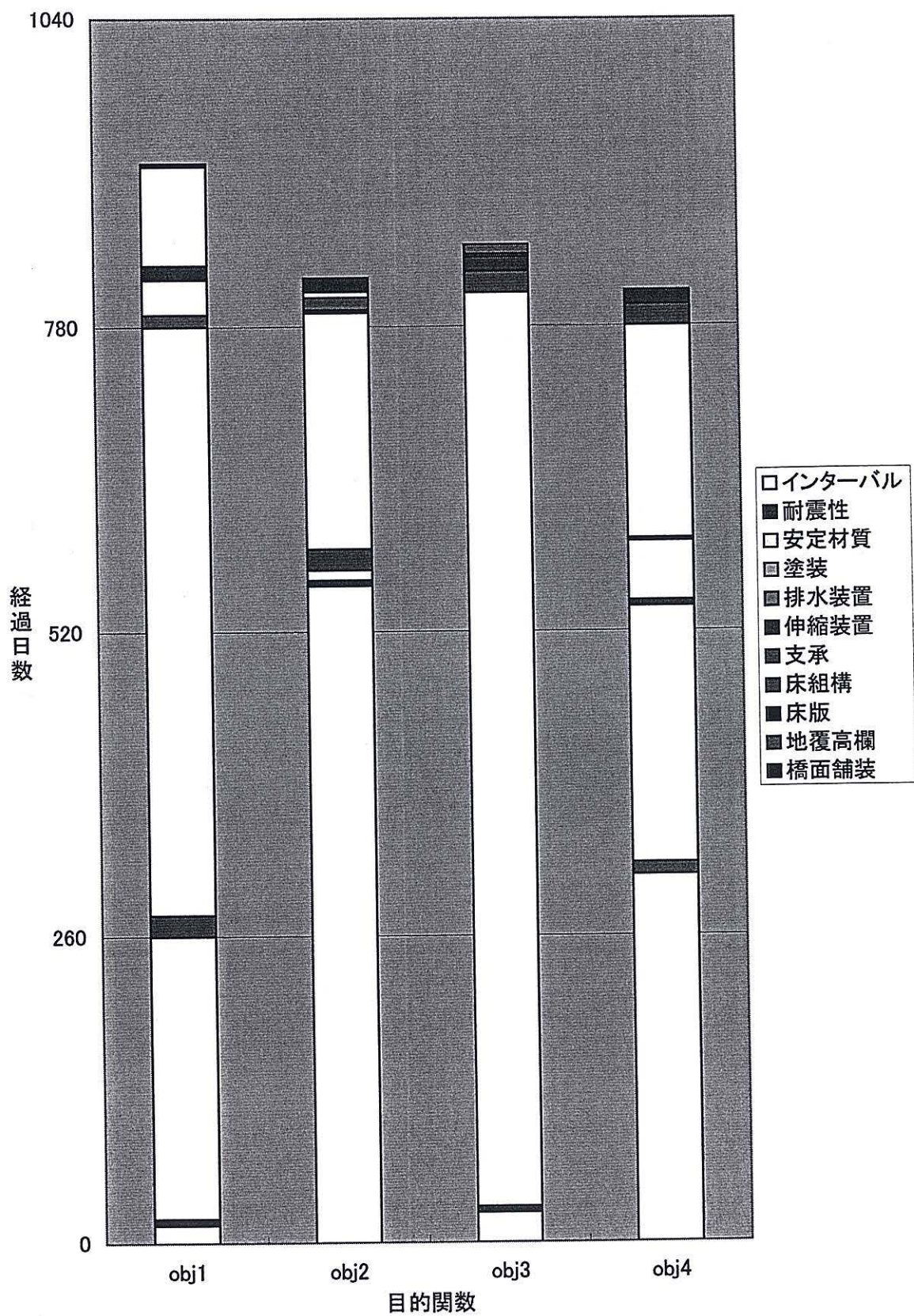


図- 2.31 CO 橋・橋梁番号 36 番の補修計画

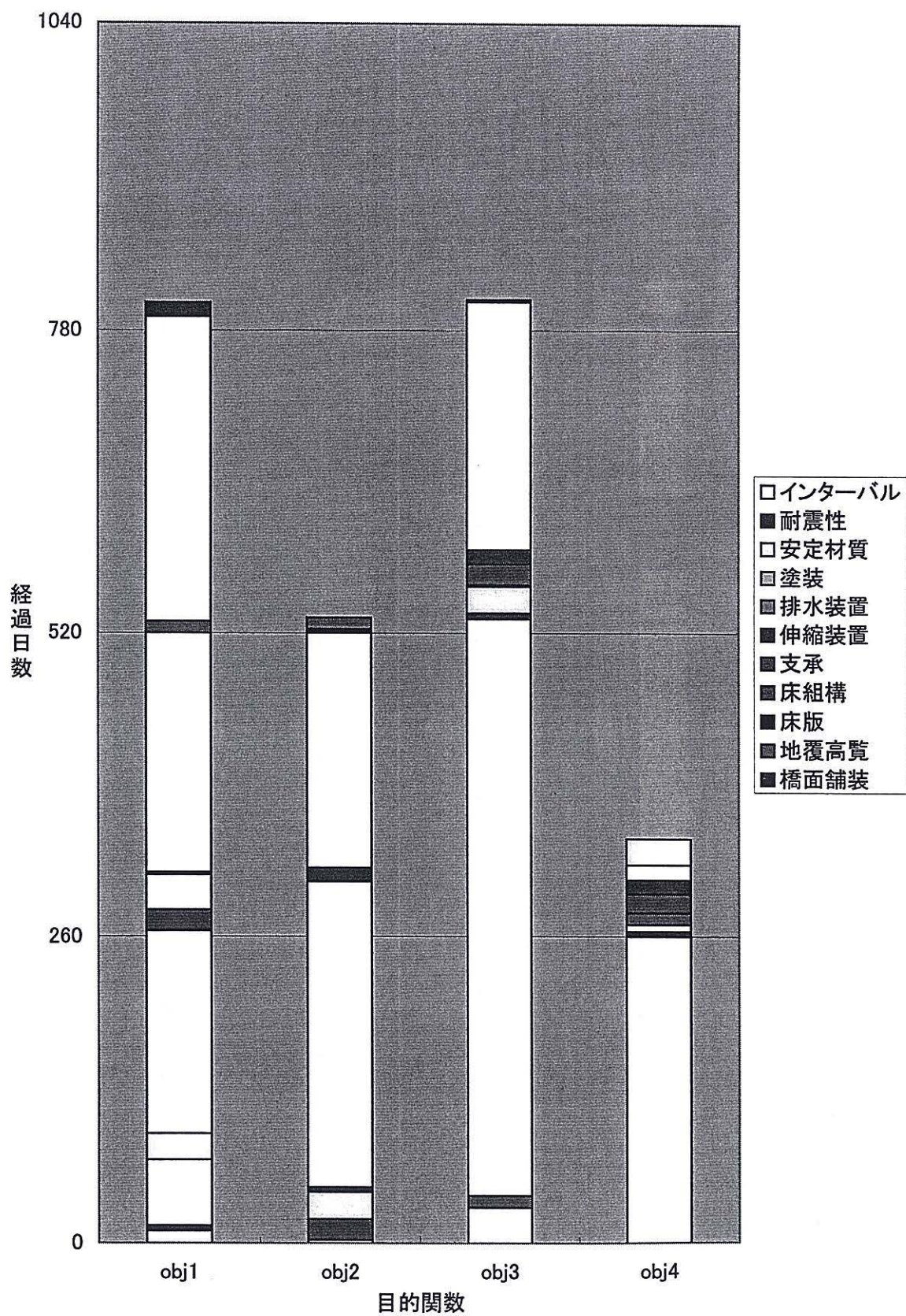


図- 2.32 CO 橋・橋梁番号 37 番の補修計画

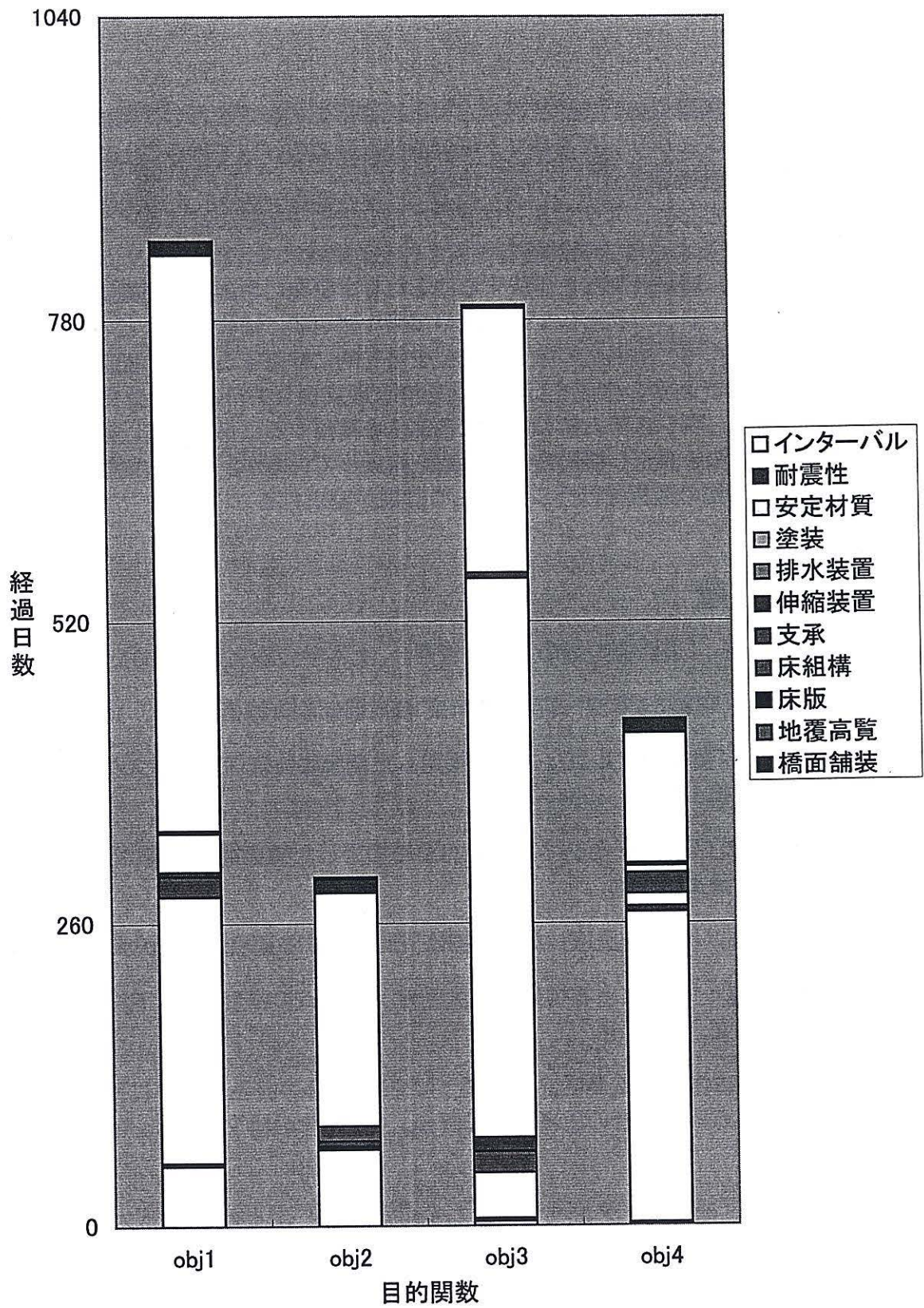


図- 2.33 CO 橋・橋梁番号 39 番の補修計画

本研究では橋梁点検データを基に、橋梁補修部位の選択を行い、選択された補修部位を基にしたスケジューリングを経て、橋梁補修計画案の作成を行うことにより数年のスパンで橋梁補修計画案を立てることが可能となった。

今後の課題として、スケジューリングにおいて同一橋梁の補修をインターバルなしで短期間に補修するような橋梁補修計画案の作成が行えるようなシステムへの改良があげられる。さらに、補修計画案を作成するにあたり本研究においては交通遮断を考慮していないが、交通ネットワークとしての評価や橋梁片側を順に補修するなどの交通への配慮が必要であり、加えて、損傷度の激しい部位を優先的に補修するなどの配慮を行った、より綿密な橋梁補修計画案の作成を行う橋梁維持管理支援システムへの改良が必要になると思われる。また、現システムにおいては各部位を補修するにあたって、補修方法としてそれぞれ1つの方法を設定し橋梁補修計画案を作成しているが、橋梁の種類や損傷度などに合わせた補修方法の選択を行うシステムの構築が望まれる。

参考文献

- 1) 西川 和廣：「道路橋の寿命と維持管理」,土木学会論文集 No.501, 1994.
- 2) 石川県：「昭和 57 年度～昭和 63 年度 橋梁診断調査総括表集」, 1990.
- 3) 橋 謙二：「遺伝的アルゴリズムを援用した橋梁維持管理支援システムに関する研究」, 金沢大学修士論文, 1994.
- 4) 木下 真二：「点検データに基づく GA を援用した橋梁補修計画に関する研究」, 金沢大学修士論文, 1995.
- 5) 福山 貴久：「GA を援用した橋梁補修計画に関する研究」, 金沢大学修士論文, 1997.
- 6) 辻岡 信也：「NN による橋梁損傷度判定システムの精度向上に関する研究」, 金沢大学修士論文, 1998.
- 7) 安田登, 白木渡, 松島学, 堤知明：「ニューラルネットワークに基づいたコンクリート構造物点検技術者の思考過程の評価」, 土木学会論文集, No.496/V-24, pp.42～43, 1994.
- 8) 谷口 健：「橋梁データ管理システムの構築に関する研究」, 金沢大学卒業論文, 1997.
- 9) 「97' Design Data Book」, 社団法人日本橋梁建設協会, pp.15～22, 1997.
- 10) 甘利 憲一：「疑問に答える土木足場の設計・施工ノウハウ」, 近代図書, pp.13～19, 1997.

- 11) 今野 浩, 鈴木 久敏 : 「整数計画法と組み合わせ最適化」, 日科技連, p.149, 1982.
- 12) 藤田 喜久雄 : 「遺伝的アルゴリズムによる最適化問題へのアプローチ」, 日本機会学会関西支部設計 AI 研究懇話会第 4 回資料, p.13, 1993.
- 13) 西川 緯一 : 「遺伝的アルゴリズムとその工学的意味」, システム/制御/情報, Vol.37, No.8, p.447, 1993.
- 14) 安居院 猛, 長尾 智晴 : 「ジェネティックアルゴリズム」, 昭晃堂, pp.10~11, 1993.
- 15) 北野 宏明 : 「遺伝的アルゴリズム」, 産業図書, p.13, 1993.
- 16) 北野 宏明 : 「遺伝的アルゴリズム」, 産業図書, p.15, 1993.

第3章 橋梁データ管理支援

3.1 管理支援システムの提案

3.1.1 概念

今日の急速な情報化の進展に伴い、パーソナルコンピュータの性能も飛躍的に向上した。また価格の低下も後押しし、パーソナルコンピュータの普及は一人一台の利用環境になりつつあるといえる。この急激な環境の変化は、様々な維持管理業務についての再検討・最構築をせざるを得ない状況をもたらしている。現在、橋梁の維持管理には、橋梁の様々なデータを橋梁台帳や橋梁点検台帳という形で紙上管理されているが、情報化の進展に伴い、これまで図面や資料など、紙上での情報を元にして運用されていた業務をパーソナルコンピュータをベースに電子情報化して行うことは、保管スペースの削減をはじめ、新たな情報の活用、共有などの可能性をもたらしてくれるものであり、供用後における橋梁維持管理業務の効率化に有効な手段と考えられる。また、電子化された情報は、誰でも簡単に、必要な部分を、最新の状態で得られるような仕組みを持ち合わせることができ、マルチメディアを活用することで資産価値をさらに向上させることができるものとする。

近年、橋梁における様々な損傷が確認されるようになり、現場での橋梁の維持管理業務が複雑化している。今後、維持管理業務はますます増加傾向にあることは明らかであり、点検業務の迅速性、正確性の向上を目標にコンピュータ上でシステムティックに行うことは、必要不可欠な項目であることは間違いない。このようなことから数々の試みが既に提案されている。本研究では様々な研究・開発資料を元に管理支援システムの一つの形を提案していくものである。

3.1.2 システム設計

データベースシステムの設計は、システムを取り巻く環境と密接な関係をもっている。環境はいつも一定ではなく流動的である。したがって、環境の変化に耐えられる柔軟性のあるシステムを作ると同時に、常に環境を分析し予想して、長期的視野でとらえなくてはならない。また、システムには必ず何かを入力しなければならず、その結果何が出力されるのが一般的な形態である。またその出力が目的にあったかどうかを、評価基

準に照らしあわせて、フィードバックして修正する。つまり、システム設計とは、ある目的に対して、どのような要素がどのように関連をもちながら機能するかを、環境との相互関係との中で構築することであるといえる。それは、現在のシステムの問題点は何か、どのようにすればデータベースがうまく機能するかといった帰納的アプローチではなく、根本的なデータベース理念はどうあるべきか、どのようなデータベース形式であるべきかという理想像を思い描く、目的志向の演繹的アプローチであり、この目的を具体的な機能によって明らかにすることである。このように、データベースシステムの設計は理念・理想をもった意志あるものでなくてはならず、それを考慮する上で、以下の項目を重要視したい。

- 目的の明確化
- 環境の予測
- 現状の把握

3.1.3 目的の明確化

データベースの構築で、最も重要なことは、何のために誰が利用するデータベースなのか、その目的を明確に位置づけることである。データベースの目的、言い換えると、あるべき姿を、データベースを構築・利用するメンバーが充分話し合うことである。ここで、メンバー共通の目的と目標を設定できるか否かが、最終的にそのデータベースの善し悪しを決定することになる。

データベースの企画

データベースの構築は、維持管理業務の手作業を機械に置き換えるだけのことではない。データベースの構築を企画するということは、その維持管理業務が何の目的で、誰のために、どんな機能をもたねばならないかを原点から考え直すということである。さらに、データベースの構築に伴って、業務の流れが変わり、人間の作業が変わる。これまで行ってきた作業を変更するには、それらに携わってきた人々がそのことを理解し、協力して、実行していかなければならない。そのためには、なぜデータベースの構築をするのかについて共通の認識をもたなければ成功しない。しかし現実にはこの点をないがしろにしても、データベースは表面上は何なくできてしまうが、問題は、構築したあと、そのデータベースが、真に機能するかどうかである。機能するというのは、データベースにとっては、情報が適切に入力され、利用者にその情報が利用されるかである。利用者、データ作成者と、データベース構築者の思惑(データベースの目的意識)がそれぞれ異なってく

ると、データベースは機能しないのである。そのためには、どのようなシステムが結果としてできあがれば、成功なのかという、評価基準を先に作っておく必要がある。

データベースの構築において、コンピュータの果たす役割は当然大きい。コンピュータの発達によって、データベースが誕生したのは事実であり、よって、どこのメーカーのハードウェア・ソフトウェアを使うかは大きな問題である。しかしながら、それは、何のためにどのような人を対象に、どんなデータベースを構築するか、データベースの企画が決定してからの選択である。また、コンピュータが空いているから構築するものでもなく、予算がついてしまったから構築するものでもない。考え方として、目的を達成するに足るソフトウェアがないならば、データベース構築を見合わせるくらいの姿勢が必要であると思われる。

ネットワーク化

これからのデータベースは、ネットワーク化が進むであろう。自力で構築するだけでなく、一般に販売されている商用データベースや、ソフトウェア会社のデータベースを利用したり共同で構築したりすることが、通信技術やメディアの発達によって、以前にも増して容易になりつつある。ネットワークとは、まず自分が何を指してデータベースを構築しているかを明確にし、その上で、自分に欠けているものを他者に求め合うものである。みんなが同じような情報源や特徴をもつデータベースを構築しているならば、誰もネットワークを組む必要がない。データベースの構築には多くの費用と人手がかかる。一度構築したデータベースは、そう簡単に変更できるものではなく、情報の蓄積を中止することはできない。本当に価値のある有効なデータベースを、各々が慎重に構築することが必要である。そのためには、既にあるデータベースの調査をして、利用できるものはネットワークを組み、独自で構築しなくてはならないものは何かを決定する。そして、最も慎重で時間をかけねばならないのが、データベースの目的を思い描くことである。

データベースシステムの構想

情報は今や維持管理業務の第四の資源である。データベースの構築を企画することは、その母体の維持管理方針にまで大きな影響を与えることは間違いない。たとえば公共図書館ならば、その自治体が、地域社会や地域経済活動の中で、情報利用や情報流通をどう考えるか、という政策にまでさかのぼることになるであろう。橋梁維持管理ならば、橋梁のこれからの維持管理業務の中で、データベースを位置づけなければならない。そういう視点でデータベースの構築を構想することが大切である。

これら三つの目的をふまえデータベースの構築の流れ図を図 3.1 に示す。

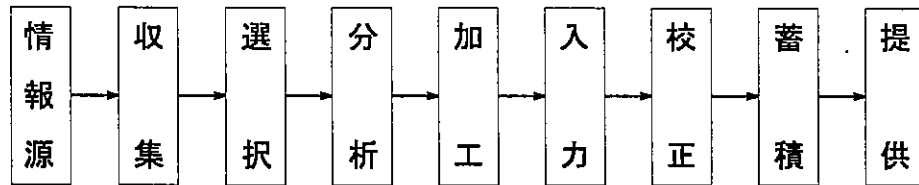


図- 3.1 データベース構築業務の流れ図

3.1.4 環境の予測

我々を取り巻く環境は、急激に変化している。情報通信技術の発達はもちろんのことであるが、一般的に言えば、国際化、ネットワーク化、多様化、個人化などが言われている。技術環境、社会環境、自然環境それぞれが、密接な関係をもって変化し、我々の情報行動に影響し合っている。このように今日、複雑で変化の激しい環境のもとでデータベースを設計しなくてはならない状況にある。

データベースの構築は企画が始まってから、実際に利用されるまで時間がかかる。さらに、一旦できあがってから修正するには時間と大変な労力がかかる。今日のように、新しい情報機器が次々に生まれてくるような状況では、数年前に企画した点で予想した機器は古くなってしまい、できあがったならば、もうそのシステムを取り巻く環境は変わっていて、使いものにならなくなる可能性もある。

図 3.2 のように、時間の流れに従って環境は徐々に変わっていく。データベースを設計する場合には、現在の環境のもとで設計するのではなく、将来の環境を予測しつつ計画することが重要となる。つまり、このデータベースが3年後にできあがる予定ならば、少なくとも5年、10年後その環境がどうなるか、それに対応して情報部門がどうなっていないかを予測して企画するのである。予測は後手にまわっては負けである。むしろ、自らが環境を作り出すような気持ち、あるいは環境を先取りするような気持ちで、将来を予測することである。もちろん、それはたやすいことではないし、予測した通りに環境が変化することの方がむしろ少ないかも知れない。そこで、常に環境を見張りながら見直しをすることである。目的を明確にし、どのように現状を把握し、将来をどのように予測したかという企画書があれば、修正がスムーズに行える。

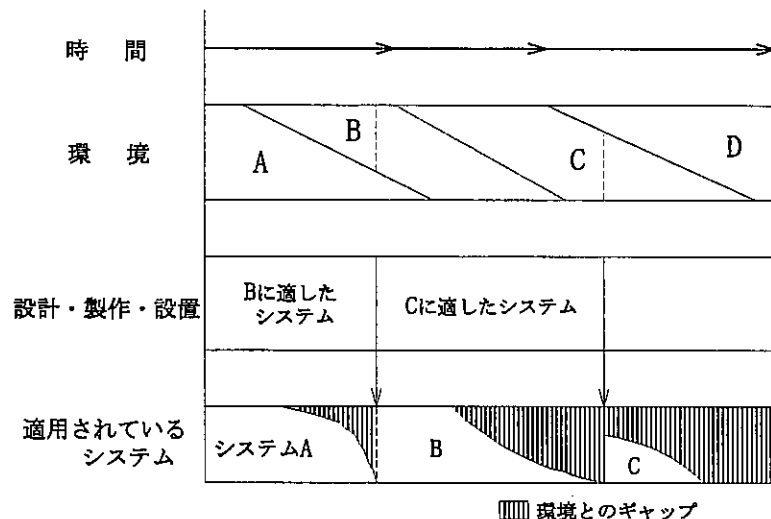


図- 3.2 システム設計と時間の流れ

3.1.5 現状の把握

データベースを設計するには、現在の状況がどのようになっているか、そして状況を構成する要素がどのように結び付き、組織化されているかを正しく把握しなければならない。組織は、なんらかの(明確になっていないにしても)目的のために、それぞれの要素が関連し、機能している。データベースを構築することは、基本的にその目的を問い直すことになるので、現在のそれぞれの要素をそのまま移行することにはならない。目的のためには、組織の変更もありうる。しかしながら、今までどのようなシステムで業務を行っていたか、このシステムに関するであろう各要素がどのように運営されてきたかを把握することは重要である。なぜなら、現状を無視してシステムは構築できないからである。システムは、環境を作り出すが、突然全く異なった環境に移行するのではなく、確実に徐々に移行させなければならない。そのためにも、現状のシステムを正しく客観的に機能に着目して把握しなくてはならない。

現状の把握というと、すぐに現行システムの問題点をあげようとするが、問題点は人それぞれによって異なり、必ずしも一致するわけではない。これは、それぞれの人が思い描く目的(あるべき姿)が異なっているからである。互いに異なる目的に対して、現実とのギャップを問題点としてとりあげても、話し合いはかみ合わない。これは結局現行システムとたいして変わらない改善に終わってしまう可能性がある。

つまり、システムの目的設定や環境の把握なしに、現状のシステムを踏襲しようとしたら、現行の問題点は何かという視点のみで現状の把握をしてはならない。

現状の把握は、一方、システムの制約条件を明らかにする。目的に向かって要素を組織すると言っても、必ずどこかで限界を生じ、制約を受ける。どのようなシステムにも制約

があり、その制約を知った上で、最適なシステムを構築していかなければならない。また現状と言うとき、利用者のニーズの把握がある。利用者のニーズは、アンケート調査などによって調べることが一般的であるが、それらの結果は注意深く参考にしつつ、鵜呑みにしてはならない。データベースは、利用者のものであるが、利用者の言うこと全てを反映することはできず、常に総合的な把握をしていく必要がある。また潜在的なニーズに対しても注意深く考慮していかなければならない。

3.1.6 データベースの運用

データベースの運用とは、データの蓄積とデータベースの利用促進をいう。新しい情報を適切に正確に取り込みながら、利用者の利用要求の変化と外部環境の変化に対応しつつ、蓄積したデータを利用させていくことが、データベースを情報活動に必須の道具として認めてもらう要件である。

データベースの構築後は、データの途切れることのない蓄積と利用促進活動の繰り返しが待っている。運用がうまくいけば、データベースをめぐる善い循環が始まり、データベースは生き生きとしたものになる。運用がうまくいかないと、いいデータがないから利用されない、あるいは、いいデータはあるがよく知られていないことが理由で利用されないということになる。利用されないから予算もつかなくなりデータ量が減り、さらに利用されなくなるという悪循環が始まる。最悪の場合にはデータの蓄積がストップしてしまう。すばらしいデータベースを設計しても、よいデータが蓄積されなければ、またよい利用者がいなければ、それはただの器でしかない。データベースの運用は、データベース構築の上で重要な検討課題である。

データの品質管理

データベースにとってデータは命であるから、データの品質が落ちれば、当然利用者は離れていき、利用されないデータベースとなってしまう。したがって、まず、よい品質のデータを蓄積することが先決である。これは、マニュアル(手作業)のデータベース、たとえば、カード目録にも共通して言えることであるが、データベースに特有な問題も含まれる。なお、データの加工・入力、最終的には作業する人の資質にかかわることであるから、いかに人の能力をあげるか、それを維持するかという根本的な問題でもある。

データに求められる要件としては、以下のようなものがあげられる。

1. 網羅性

データベースが対象としている分野・範囲のデータを網羅しているか。

2. 迅速性

データベースに蓄積されるまでのタイムラグ、更新頻度はどのくらいか。

3. 一貫性

データの表現や索引が統一されているか。

4. 正確さ

データが正しいか。

3.1.7 維持管理システム構築の目的

10数年前から、資料の保管に際して紙上管理されていた資料を電子情報(デジタルデータ)に変えて、電子ファイリングシステムによって運用管理を行うケースが多く見られるようになった。土木分野においても建設CALSをはじめ、情報をデジタルデータで扱うケースが多くなってきている。しかし、これまでの電子ファイリングは専用の機械やシステムが必要で、また専門のオペレータでないと動かせないなどの問題から、高価な割にはあまり有効に機能していなかった。情報の活用、共有などを考えた場合、利用する立場から、日常業務の中で使用しているOA環境の中でシステムを構築し、運用していく必要がある。こういった中最近では、パーソナルコンピュータ、および、周辺機器のコストパフォーマンスの向上、また情報の格納メディアとしてCD-ROMが普及し、安価に、誰でも有効に利用できるシステム構築が可能となってきた。

このように、情報処理技術の進歩によってパーソナルコンピュータ上でデータベースの構築が可能となった現在、橋梁台帳・橋梁点検台帳等の必要な各種データをコンピュータ上でデータベース化しシステム構築することは、データ処理の向上の他、多くの利点を生むことは明らかである。本システムでは以下の3つの条件を主眼に、構築を進めている。

1. 既存環境の有効利用

既存の環境をできるだけ有効に活用できるようなシステムを構築するということである。パーソナルコンピュータの性能が向上し、それらが多くの職場で普及した現在、その環境を無駄なく上手に利用していくことが大切である。これはネットワークの形成が有効であり、個々のシステムによる集合が大規模システムに劣らないデータベースの構築を可能にするものとする。また既存環境の有効利用によるトータルコストの軽減が期待できるという利点もあげられる。

2. データの共有化

データの共有を容易にするため汎用性のあるソフトウェアを利用するということである。システムのプラットフォームをMicrosoft Windows95(以下Windows)にする

ことにより、数多く存在する Windows ソフトウェアとの連携が可能となり、新たなシステムの開発や移行が比較的容易にできるようになると考える。

3. 操作性の向上

ユーザに対し専門的な知識なしで利用できるようなユーザフレンドリなシステムを構築するということである。本研究では市販のデータベースソフトである Microsoft Access97(以下 Access) を使用している。これは第二の条件を満たすとともに、統一化されたインターフェースにより、初心者も含めた多くのユーザが利用しやすいなどの利点を生むものと考えられる。

3.1.8 既存の構築例

今日、道路橋のストックに対する維持管理を支援するためのコンピューターシステムは数多く研究され、開発が試みられている。本研究においてシステムを構築するにあたり、これらの事例を十分考慮したうえで開発を進めることは、柔軟なシステムの構築へ向けての重要なプロセスであると考えられる。

以下に橋梁データの管理に絞って二つのシステムの実例を、GIS と連携した事例を一つあげることとする。

システム例 1

山口大学の宮本らが「Bridge Management System(BMS)における最適維持管理計画の策定」¹⁾という題で公表しているシステムがある。これは橋梁の維持管理等に投資できる予算には上限があり、定期点検結果を基に必要な維持管理対策を順次実行していくと仮定すると、近い将来、老朽化した橋梁の補修・補強費が新設橋梁の建設費を圧迫する事態になることを考え、限られた予算の中で最大の効果を得るための維持管理計画が作成できるシステムの必要性を背景に構築されたものである。

このシステムは、橋梁の劣化診断、劣化予測、経済性および橋梁部材の品質の両方を考慮した最適な維持管理計画の作成を目標にしており、データを「橋梁データベース」という形式で格納し、そこからデータを抽出しシステムに利用する形態を採用している。システムとしては器であるデータベースの部分よりも計画案の作成に重点を置いたシステムであり、付随的なシステムとして利用するものと思われる。

システム例 2

建設省土木研究所が「橋梁マネジメントシステムの開発」²⁾という題で研究するシステムを取りあげている。これは道路橋の維持管理(点検・評価・補修)の中で評価と補修計画の一部をコンピュータ化して合理的に作業が行えるように支援するものである。データベースとしては、道路に関するデータを蓄積、活用できる道路管理データベース(MICHI:建設省道路管理データベースシステム)を構築している。それには橋梁構造データ・点検補修データなどが格納されており、それを利用して補修計画作成モジュールが橋梁ごとの評価を行い、それを利用して管理する橋梁群の補修計画案を立案し、実際の補修計画の作成時に役立てようとするものである。システムはWindows上でデータベースとモジュールを組み合わせ独自に開発しており、これはある特定の限定された技術者を対象に構築したものであると思われる。

システム例 3

「LIFECYCLE MANAGEMENT OF NETWORK-LEVEL BRIDGES」³⁾という題で、名古屋大学の伊藤らが公表しているシステムがある。Bridge Lifecycle Management System(BLMS)においてのデータ作成、情報収集、GIS技術、およびオブジェクト指向のアプローチによるシステムは、GISによるネットワーク解析においてライフサイクル橋の維持・管理における支持ツールとして役立つように開発されている。このシステムにおいて、業務に関係するエンジニアは維持管理戦略をより迅速に識別し指定すること、および総維持管理業務の大幅な効率化が望め、また開発されたシステムを実際に、いくつかの実用的な問題により試験され、適応性をみている。この研究は本格的なGIS(ARC/INFO)を用い構築されており、GISでの代表的な解析手法であるネットワーク解析を用い有効性を模索している。本研究ではこのような本格的なGISは利用できないが、地図データとの連携を目的に最大限のシステム可能性を探りたいと考えている。

上記の研究はそれぞれビジョンを掲げ、独自のフォーマットにより橋梁維持管理システムの構築を行っている。本研究ではそのシステム例とは異なるアプローチで橋梁維持管理システムの構築を行う。

特色として以下のような方針でシステム構築を実行している。

本システムの基本方針

構築するにあたり、基本方針としてデータベースという器の部分を重要視した。これはデータベースの部分をI県庁が実際に利用することから、実用的なシステムでなくてはな

らないからである。たとえば、ユーザーがデータを入力する場合、紙の台帳と同じような感覚で、戸惑いなく管理できるように既存の台帳のイメージをできるだけ再現するように工夫している。また、特別な設備投資なしで、現在のコンピューター環境を最大限に有効利用できるシステムを目的とし、ネットワーク環境からのブラウザのみでの利用を実現している。これは結果として、オープンなデータの共有環境が生まれ、またミニマムコストも可能とさせる。

このようなシステムは、操作が専門的かつ特殊になりがちであるが、本システムは利用者を限定するのではなく、現場での意見を慎重に取り入れ、ユーザーにやさしいシステムの提案を試みている。ソフトウェアは市販のソフトウェアである Access で構築し、キーボードでの入力はいできるだけ避けることにし、データの入力はメニューによる選択、選択肢の絞り込みなどを多く組み込んでいる。Access 単一での構築は他のシステムへ移行する場合でもデータの可搬性があり、データが扱いやすいという利点をも生む。また、Windows 上で Access でのデータベース構築は、数多く存在する Windows ソフトウェアとの ODBC(Open DataBase Conective) を利用した連携の容易さ、また今後の発展からみても有効であると思われる。

このような意味からして、データを蓄積する器は将来進展するであろう総合システムへの基盤であり、器の部分の疎かにしては確実な進展はありえないものとする。今一度データベースシステムのあり方を再考し、維持管理業務が増大する中での橋梁維持管理システムの一構築例として提案したいものとする。

3.2 橋梁データ管理システム

3.2.1 システムの概要

目的と対象

本システムはI県が管理する橋梁の各種情報を Windows 上でデータベース化し、一括管理することを目的としている。

現在、I県が管理する橋梁は約 1500 橋あり、これら全てを単に紙上による台帳管理だけでは、閲覧・修正を行うことは容易ではない。本システムは閲覧・修正だけでなく、新規登録やデータ検索を容易に行うことができるものであり、本研究が提案する橋梁データ管理支援システムの骨組みであるといえる。

今回紹介する橋梁データ管理システムを、維持管理業務のルーチンワークに利用することにより、以下の効果が期待できる。

- 橋梁状態確認の迅速化

机上のノートパソコンを用いて簡単に橋梁台帳・点検台帳の検索が行えるため、要

求に応じリアルタイムに対応できる事が可能になり、維持管理業務の迅速化がはかれる。

- 省スペース化

大量の橋梁台帳がCD-ROM数枚に格納され、業務処理効率の向上が図れるとともに、資料保管スペースの縮小化が図れる。

- 電子情報化による同常業務での図面情報の活用

図面、写真をデジタルデータ化していることにより、ソフトウェア上での連携が図られ、様々な局面で資料として利用できる。

システムの構成

本システムはデータを登録し、閲覧・印刷するといった基本的な機能から、データを修正・削除する補助的な機能、さらに条件付きでデータを探し出すことのできる検索機能を備えている。特に検索機能は、複数の条件下（例えば「道路種別〇〇の□△川に架橋されている橋梁」など）でも検索できるようになっている。そして、システムの操作はパーソナルコンピュータの利用において初心者进行想定し、キーボード操作を極力抑えたマウス操作中心のシステムを構築している。

収録データの内容

紙上管理されていた橋梁台帳・橋梁点検台帳は多くの情報を持っているので8つのテーブルに分割し、それぞれにデータベースファイルを設定している。

橋梁台帳は内容を橋梁データ・材料集計表の2つに分割している。橋梁データには土木事務所名、橋梁名、河川名、所在地、設計・施工業者名、設計条件、交通量といった事項が記載されている。一方、材料集計表には、上部工・下部工・基礎工それぞれの材料の種類、構造形式、断面寸法、工事費(単位工事費)等が記されている。

橋梁点検台帳には外観(異常)調査により、上部工、下部工それぞれの点検項目(材料)の異常の有無、さらにその内容が記されている。ここで、外観調査によるデータについては、外観調査に以下にあげるような点検調査の項目を加えて実施していると捉え、データを入力することにより、橋の現況がさらに明確になると思われる。よって、それぞれにデータベースファイルを設定している。

- 震災点検

- － 耐震性・液状化に対する橋の危険度

- ASR 調査
 - － アルカリ骨材反応による被害の度合い
- 塩害調査
 - － 塩害の度合い

さらに、施工月日・位置・補修工法・補修工費が記録されている補修歴を上部工・下部工についてそれぞれ二つと、4段階評価で、橋全般の、危険度の評価をしている橋梁診断についてデータベースファイルを設ける。

また橋梁台帳・橋梁点検台帳・設計図における画像データの输入のためそれぞれにテーブルを設けている。

分割された6つのテーブルは以下の通りである。

1. 橋梁データ
2. 材料集計表
3. 橋梁点検台帳
4. 画像(橋梁台帳)
5. 画像(橋梁点検台帳)
6. 図面

これらのテーブルは、それぞれのデータベース間を橋梁固有の情報である橋梁番号を主キーに設定し連携を図っている。こうすることによって閲覧の際、一つの橋梁に対して6つの情報の中から選んで操作できるので、情報操作の無駄を省き、時間のロスを少なくすることができる。

3.2.2 システムの説明

データベースシステムを構築するにあたってそのシステムの良否を判別するのに重要なファクターとして

- 機能性
- 信頼性
- 簡便性

の3つがあげられる。これらはこのうちのひとつだけ優れていれば良いというものではなく、全てがバランス良く機能していなければならず、この点が本研究である橋梁データ管理システムを構築するにあたり最も考慮した点である。

本システムは本章のデータ入力・閲覧を中心とした基本システムと次章のネットワークを考慮したシステムによって構成されている。以下はその基本システムの特徴について述べることにする。

閲覧

橋梁台帳・橋梁点検台帳をパーソナルコンピュータ上に移植するにあたり、最も注意した点は紙上の管理と変わらない感覚で閲覧できるということである。橋梁データ画面(図 3.3)と材料集計表画面(図 3.4)を以下に示す。

橋梁データ

橋梁データ

ID: 1002 橋梁名: 高橋 ふりがな: たかまし

作成日付: 97/02/09 管理者: 大聖寺土木事務所 道路種別: 一般国道(指定外)

道路区分: 未記入 路線名: 305号 所在地: 加賀市南郷町

架橋位置 対象: 河川 位置: 三谷川

橋梁改修 改修状況: 改修済 年号: 昭和 60 年度 計画流量(m³/s): 0

河川改修 改修状況: 年号: 年度 確率年数:

交通量(現在)

年号: 年度 自動車: 台 重車両: 台 自転車: 台 歩行者: 人

交通量(将来)

年号: 年度 自動車: 台 重車両: 台 自転車: 台 歩行者: 人

橋種 1: 単純非合成板桁橋 橋種 2: 橋種 3: 橋種 4:

バス路線: 通走路: 都計決定: 都計幅員:

支間割 1: 10.460 支間割 2: 10.460 支間割 3: 10.460 橋長(m): 33.00

支間割 4: .000 支間割 5: .000 支間割 6: .000 橋面積(m²): 264.00

幅員構成(m) 左歩道: .00 左路肩: .00 左車道: 4.00 中央帯: .00

右歩道: .00 右路肩: .00 右車道: 4.00 有効幅員: 8.00

下部工形式 A1: パイルベント P1: パイルベント P2: パイルベント

P3: P4: P5: P6: P7: P8: P9: P10: P11: P12: P13: A2: パイルベント

基礎工形式 A1: RC杭基礎(工法不明) P1: RC杭基礎(工法不明) P2: RC杭基礎(工法不明)

P3: P4: P5: P6: P7: P8: P9: P10: P11: P12: P13: A2: RC杭基礎(工法不明)

平面線形データ

曲率半径R(m): 0 クロソイドパラメータA(m): 0

縦断勾配(%): +.0 ~ -.0 横断勾配(%): +2.0 ~ -.0

斜角1 方向: 53° 30' 0" 斜角2 方向: ° ' 0"

設計荷重 適用示方書: 未記入 荷重1: TL-14 荷重2: 未記入

設計水平震度 Kh: ~ 添架物1: 添架物2:

特記事項: 施工年度(下部): 昭和60年1月-5月 添架物3: 添架物4:

橋下年度(工費): 開始年度: 昭和 60 年度 10 月

1 / 176

図- 3.3 橋梁データ画面

材料集計表

ID: 3002 橋梁名: 直海谷橋

上部工

SM570 (t) 0 SM520 (t) 0 SM490 (t) 0 SS400 (t) 0 その他 (t) 0
合計 (t) 0

伸縮装置 (1) 鋼製ジョイント 上塗装 長油性フタル酸樹脂
伸縮装置 (2) 中塗装 長油性フタル酸樹脂
支承 (1) 下塗装 鉛系錆止めペイント
支承 (2) 塗装面積 (m²) .00

床版厚 (cm) コンクリート体積 (m³) 鉄筋 (t)

コンクリート体積 (PC橋) (m³) PC鋼棒 (PC橋) (t) 鉄筋 (t)

コンクリート体積 (RC橋) (m³) 鉄筋 (RC橋) (t)

下部工形式

A1	重力式	P1		P2	
P3		P4		P5	
P6		P7		P8	
P9		P10		P11	
P12		P13		A2	重力式

下部工高さ (m) 支承条件 下部工コンクリート体積 (m³)

A1		P1		A1		P1	
P2		P3		P2		P3	
P4		P5		P4		P5	
P6		P7		P6		P7	
P8		P9		P8		P9	
P10		P11		P10		P11	
P12		P13		P12		P13	
A2		A2		A2			

基礎工形式

A1	直接基礎	P1		P2	
P3		P4		P5	
P6		P7		P8	
P9		P10		P11	
P12		P13		A2	直接基礎

基礎工直径 (mm) 基礎工長さ (m) 基礎工本数 (本)

A1		P1		A1		P1	
P2		P3		P2		P3	
P4		P5		P4		P5	

1 / 176

図- 3.4 材料集計表画面

入力したデータはそれぞれ3つのシートに分割され印刷する事ができる。印刷のフォーマットは利用者の意見を十分配慮し、最も利用しやすいと思われる図 3.5 のように決定している。

橋梁台帳 Sheet1 日付: 平成11年01月29日

ID:	01002		登録者:	大塚守士 土木事務所		道路種別:	一般国道(指定外)		道路区分:	未記入		作成日:	97/02/09			
ふりがな:	たかはし		路線名:	306号		架橋位置:	河川									
橋名:	高橋		所在地:	加賀市南田町												
改修の道筋:	改修済	存続	昭和	平成	60	改修	河川	存続	平成	昭	平成					
交通量	現在	将来	自動車	台	重車両	台	自転車	台	歩行者	人	計画流量(Q2/Q)	0				
橋種	単線非合成鉄桁橋															
支間割	10.46		10.46		10.46		0		0		橋長	33.0m		橋面積	264.00㎡	
幅員構成	左歩道	0.00m	左路肩	0.00m	左車線	4.00m	中央帯	0.00m	設計決定	橋脚						
	右歩道	0.00m	右路肩	0.00m	右車線	4.00m	全幅員	8.00m	RC基礎	設計幅員						
下部工および 基礎工	A1	パイルベント		P1	パイルベント		P2	パイルベント								
	P3	RC基礎(工法不明)		P4	RC基礎(工法不明)		P5	RC基礎(工法不明)								
	P6			P7			P8									
	P9			P10			P11									
	P12			P13			A2	パイルベント								
								RC基礎(工法不明)								
橋形	断面(左記%)	*0~*0		断面(右記%)	2.0~*0		平面形状(Rm)	0m(%)		0						
設計荷重	斜角	A1橋台		53°30'0"		A2橋台										
適用示方書	未記入	活荷重1	HL-14		活荷重2	未記入		水平態度	~							
建築期	特記事項: 年度(下部) 昭和60年1月~															
施工年度	上部工	昭和34年12月~		下部工	昭和3年3月		上部工	昭和3年12月~		昭和3年2月						
設計業者	森本設計				詳細設計		朝日エン지니어リング									
施工業者	下部工		富士車輦(株)		下部工		大同コンクリート		向出組(昭69)							

図- 3.5 印刷プレビュー画面

橋梁のイメージ画像(図 3.6)もデータとして取り込めるようにした。このほかに橋梁点検時における画像(橋梁点検台帳), また橋梁の設計図(図面)を入力できるフォームも加えている。数値データと画像データの共存は橋梁の全体的なイメージを捉えるのに有効であるとする。

画像(橋梁台帳): フォーム

1002高橋

ID 1002 橋梁名 高橋

作成日

図- 3.6 画像(橋梁台帳)

データ入力

データベースの基本であるデータ入力作業はデータの信頼性という点からしても重要な項目であることは疑いない。本システムは誤データ入力を最低限抑えるため、多くの工夫を凝らしている。

まず第一に操作を極力簡単にし、誰に対しても利用が可能となるユーザーフレンドリなシステムを目指している。操作がいちいちマニュアルを見なくてならないシステムでは、操作ミスをする危険性が高く、誤データ入力という形になりかねない。つまり理想的なシステムは、画面を見れば、だいたい操作が理解できるようなマニュアル不要のシステムでなければならない。よって本システムの操作はアイコン中心で進行し、データの入力はマウスを利用したプルダウンメニューの選択(図 3.7)を多用している。

図- 3.7 プルダウンメニュー表示

またデータの入力に際し、同時に特定されるデータはクエリーによって抽出を行い、選択肢の絞込み(図 3.8)を実行している。これによって関係のない選択肢を排除し「マウスによるプルダウンメニューの選択」+「データの絞込み」により、より信頼度の高いデータ入力が可能になった。

橋梁データ

ID: 1002 橋梁名: 高橋 ふりがな: たかはし

作成日付: 97/02/09 管理者: 大聖寺土木事務所 道路種別: 一般国道(指定外)

道路区分: 未記入 路線名: 305号 所在地: 加賀市南郷町

架橋位置 対象: 河川 位置: 三谷川

橋梁改修 改修状況: 改修済 年号: 昭和 60 年度 計画流量(㎡3/s): 0

河川改修 改修状況: 年号: 年度 確率年数:

交通量(現在)

年号: 年度 自動車: 台 重車両: 台 自転車: 台 歩行者: 人

交通量(将来)

年号: 年度 自動車: 台 重車両: 台 自転車: 台 歩行者: 人

橋種 1: 単純非合成板桁橋 橋種 2: 橋種 3: 橋種 4:

バス路線: 通学路: 都計決定: 都計幅員:

支間割 1: 10.460 支間割 2: 10.460 支間割 3: 10.460 橋長(m): 33.00

図- 3.8 データの絞込み

検索

データベースを利用する際、ユーザーが要求するデータを素早く検索できることは、業務の円滑な執行に役立つものであり、「生きたデータベース」を提供するものである。そのようなシステムを目指し、本システムでは検索機能を充実させている。

図 3.9 は五十音順に検索できるフォームである。画面はア行の文字から始まる橋梁名を抽出している。このフォームにおいてはデータ画面で Access の抽出機能も利用でき、確実な検索を実現している。

ID	橋梁名	管理者名	道路種別	路線名	所在地	対象	位置
1005	越前橋	大聖寺土木事務所	一般国道(指)	305号	加賀市大聖寺越前	河川	大聖寺川熊坂
1051	動橋跨線橋	大聖寺土木事務所	主要地方道	山中伊切線	加賀市動橋町	鉄道	北陸本線
1136	動橋大橋	大聖寺土木事務所	一般県道	串加賀線	加賀市動橋町	河川	動橋川
2029	江指大橋	小松土木事務所	一般国道(指)	416号	小松市江指町	河川	獺川
2057	絵馬堂橋	小松土木事務所	主要地方道	小松山中線	小松市津波倉町	河川	日用川
2088	浮柳新橋	小松土木事務所	主要地方道	小松加賀線	小松市城南町	河川	前川
4131	大俣橋	金沢土木事務所	主要地方道	金沢井波線	金沢市二俣町	河川	田島川
4134	医土山橋	金沢土木事務所	主要地方道	金沢井波線	金沢市二俣町	河川	森下川

図- 3.9 検索画面 1

図 3.10 は条件指定型の検索フォームである。10項目の中から、条件を与えられるものだけ選択し、「検索の実行」ボタンをクリックすることにより、AND 検索によってデータが抽出される。必要とするデータを漠然に知っている場合などの検索に威力を発揮する。

項目	入力形式
ID	テキストボックス
道路種別	プルダウンメニュー
道路区分	プルダウンメニュー (開いている)
路線名	プルダウンメニュー
所在地	プルダウンメニュー
架橋対象	プルダウンメニュー
架橋位置	プルダウンメニュー
適用示方書	プルダウンメニュー
荷重1	プルダウンメニュー
荷重2	プルダウンメニュー

検索の実行

図- 3.10 検索画面 2

図 3.11 は橋梁点検台帳における各橋梁評価値から検索できるフォームである。各評価値を指定することにより各橋梁の損傷の程度が確認でき、維持管理業務に有効に活用できるであろうと考える。

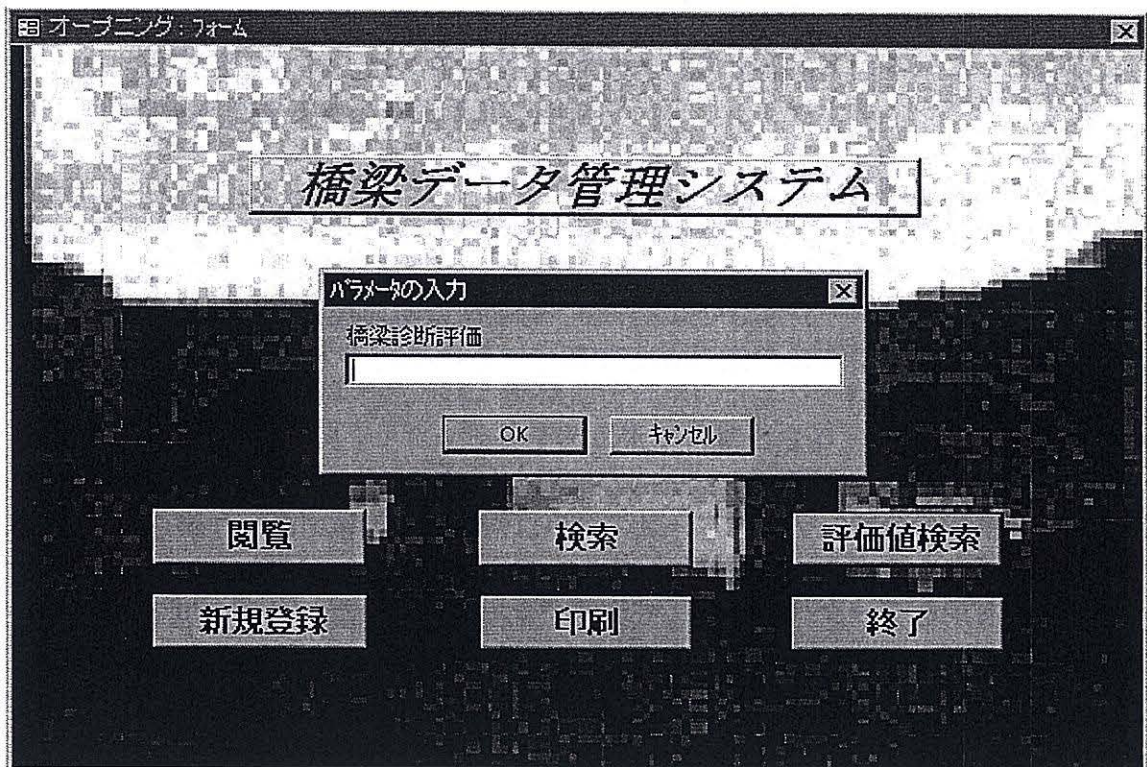


図- 3.11 検索画面 3

3.3 ネットワーク環境を考慮したシステム

3.3.1 本システムの目的

今日のインターネットの爆発的な普及に見られるように、パーソナルコンピュータの環境は大きく様変わりしている。例えば、民間企業にはLAN構築によるイントラネット環境が整備されるものも多く現れ、作業の効率を大幅に上げる意味でも有効であることが証明されている。このようにインターネット・イントラネットをベースにして内外ともシー

ムレスなネットワークを構築しておくことは、これからのデータベースシステムの方向性であることは間違いない。本章では橋梁データ管理システムをネットワーク対応とし、既存のデータベースシステムの新たな発展を試みる。

ネットワークシステムの利点

第3章で紹介した橋梁データ管理システムは、データの登録、閲覧、印刷などの機能を中心とした基本システムであり、1パーソナルコンピュータに閉じたシステムであるといえる。基本システムをネットワーク対応にするにあたって生ずる利点を以下に述べる。

1. オープンな共有環境

ネットワークシステムを構築するにあたり最大のアドバンテージは、機種やアプリケーションに依存しないオープンな環境が構築できるところにある。これは外部とのコミュニケーションではオープンな環境は重要な要素だからである。しかもブラウザというシンプルな道具立てで即座に簡単に利用でき、ユーザーにとっても利点である。

2. フレキシブルなデータベースの構築

情報の公開とデータベースの構築は情報活用の核でありながら、一つのシステムに閉じたものが多く、柔軟なシステム構築の良い方法が無かったことは否めない。つまり有用なデータベースとして実感できるものが少ないのが事実である。しかし、ネットワーク対応にするにあたりブラウザ上での優れた検索機能を利用することで必要なとき必要なだけデータが抽出でき、また枠組みさえ用意すれば今までのデータがそのまま活用でき、不要になれば削除すればいい。このフレキシビリティがデータメンテナンスを容易にし、「使えるデータベース」の提供を可能とするであろう。

3. ミニマムコスト

ネットワークシステムのデータベース構築は小さなコストで始められることも見逃せない。基本的にサーバーとなるコンピュータにデータセットを置くことで、ブラウザを通じ閲覧が可能となる。利用者のコンピュータはブラウザの起動が最低条件であり、それらのコンピュータにはそれほど高い性能を要求しなくてもよい。これは設備投資が最小で、現在の現場システムを最大限に利用できることを意味する。

3.3.2 システムの構成

本システムを構築するにあたり利用した主なソフトウェアを以下に述べる。

1. Microsoft WindowsNT Server

WindowsNT オペレーティングシステムと他のオペレーティングシステムとの重要な相違点の1つは、ネットワーク機能が最初からシステムに組み込まれているということであり、つまりクライアント/サーバーコンピューティングを実現できるように設計されている。これはシングルユーザーの汎用ワークステーション(クライアント)をマルチユーザーの汎用サーバーに接続し、両社で処理負荷を分散することを意味する。クライアントはサービスを要求し、サーバーはサービスを提供することによってこれに応えることができる。また信頼性・堅牢性・互換性が優れており、拡張性についても問題はないことから使用することにした。

2. Microsoft dbWeb

Microsoft dbWeb(以下 dbWeb)は、ODBC データソースと IIS(Internet Information Server)間のゲートウェイである。dbWebを使うと HTML(Hypertext Markup Language)のページで、ODBC データソースのデータを選択して公開することができ、Web ブラウザのユーザーは、簡単にデータのクエリーができ、ハイパーリンクを使いながらデータ間をナビゲートすることが可能となる。dbWebでは、パラメータを作成して、ユーザーがどのデータにアクセスできるかを制御することができる。

3.3.3 IIS について

IISは、Windows NT Server 4.0に付属しているインターネットサーバーであり、WWW サービス、Gopher サービス、FTP サービスという3つのサービスを提供することができる。これらの3つのサービスのうち、特徴的なのはWWW サービスである。一般的なWWWサーバーと比べて、IISのWWWサービスには大きく2つの機能が追加されている。1つは、アプリケーションを作成することなくデータベースにアクセスすることができる「Internet Database Connector(IDC)」という機能である。この機能を用いれば、プログラマでなくとも、Web ブラウザから自由にデータベースに対してアクセスできる環境を構築することができる。もう1つは、サーバー側でプログラムをより高速に動作させることのできる「ISAPI(Internet Server Application Programming Interface)」というインターフェースの搭載である。ISAPIインターフェースを利用したプログラムは、従来のCGIプログラムとは違い、標準入力や標準出力を経由してサーバーとやり取りするのではなく、直接プログラムと通信するため、より高速に動作させることができる。また、プログラム上の制限も緩和されるため、より自由で柔軟なアプリケーション開発が可能になる。以上は、IIS2.0までの機能であるが、Windows NT Service Pack 2に含まれているIIS3.0では、Active Server Pages というモジュールにより、サーバー側で VisualBasicScript(VBScript)

や JavaScript などのスクリプト言語によるプログラムが動作するようになっている。

アクセスのしくみ

まずは、IIS がデータベースに対してどのようにアクセスするか、その概念を説明する。IIS がデータベースに対して操作を行う場合の手順は図 3.12 のようになる。

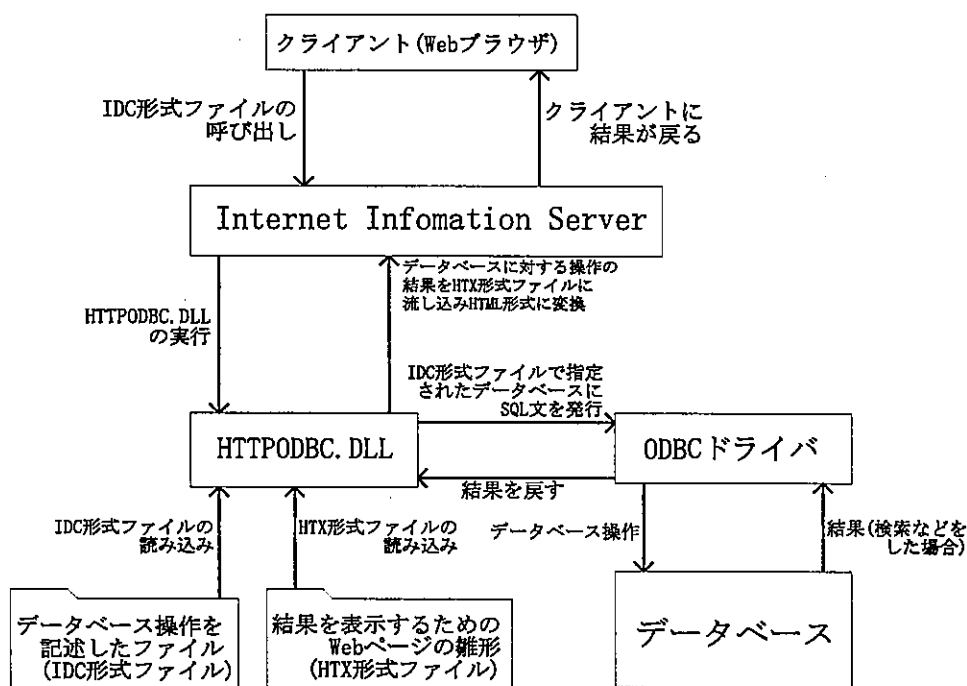


図- 3.12 アクセスのしくみ

この図を見るとわかるように、データベースへのアクセスは、クライアントが IDC 形式ファイルを読み出したタイミングで行われる。IDC 形式ファイルには、どのデータベースをどのように操作するかを記述しておく。IDC 形式ファイルは直接呼び出される場合、Web ページに埋め込まれたフォームから呼び出される場合がある。IDC 形式ファイルが読み込まれると、IIS によって HTTPODBC.DLL が実行される (HTTPODBC.DLL は IIS をセットアップしたときに自動的にインストールされている)。この DLL ファイルが実際の IDC 形式ファイルを解釈し、ODBC ドライバに対してデータベースへの命令を下す。これだけでもデータベースへのアクセスは行われるのであるが、クライアント側に結果を戻す手段がない。そこで、HTTPODBC.DLL は、「HTX 形式ファイル」と呼ばれる HTML 形式の雛形となるファイル (テンプレートファイル) を読み込み (どの雛形を読み込むかは IDC 形式ファイルの中で指定する)、データベースの結果をそのテンプレートファイルに流し込

んでHTML形式に変換したあと、クライアントに送信する。この説明から正確に言うと、データベースアクセス機能はISSそのものが持っているのではなく、HTTPODBC.DLLが持っているのである。HTTPODBC.DLLは、データベースアクセス機能を持った単なるDLLファイルであり、仕組みとしてはサーバー側でプログラムを実行するCGIアプリケーションの類と同等である。

IDC/HTX

IDC(Internet Database Connector)/HTX(HTML Extension)はデータベースのデータを使用して表形式のページを作成するのに非常に適したシステムであり、拡張子が.IDCと.HTXの2つのファイルから構成されている。

IDCファイルにはODBCデータソースに接続するためのデータソース名と、クエリーを実行するためのSQL文が格納されている。

HTXファイルはIDCファイルのためのHTML形式のテンプレートとして使用され、HTXファイルには通常のHTMLタグと、IDCで抽出された各フィールドのデータを適切な場所に挿入するためのプレースホルダ、データのフォーマット方法を制御するためのキーワードが格納される。

IDC/HTXの動作

Webブラウザがページを要求し、IDC/HTXを使用してWebサーバーがページを作成してブラウザに提供するまでの動作の流れは図3.13のようになる。

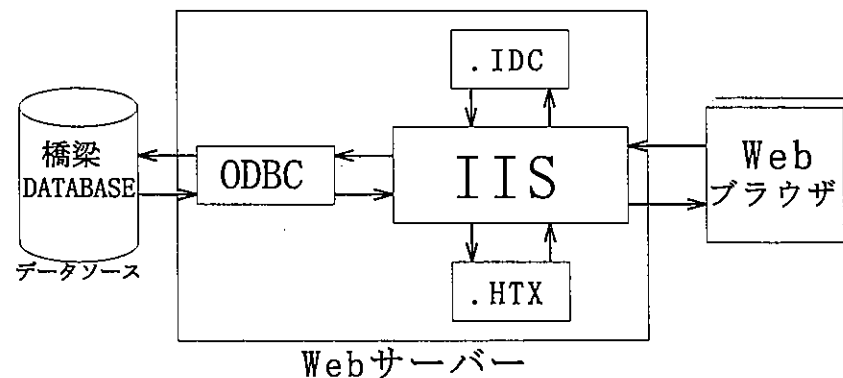


図- 3.13 動作の流れ

1. Webブラウザが .idc の拡張子を持つページを Webサーバーに要求する

2. Web サーバである IIS は IDC ファイルを参照する
3. ODBC を通してデータソースへと接続を行い、クエリーを発行する
4. クエリーの結果を取得する
5. IIS は IDC ファイルの中で指定された HTX ファイルをテンプレートとしてクエリーの結果を HTML 形式にフォーマットする
6. Web ブラウザに提供する

3.3.4 システムの説明

ネットワークシステムの最大の利点はリモート操作によるデータの収集である。基本となるデータをサーバーに格納し、クライアントが欲しいとき欲しいだけデータが得られるシステムはトータルコストを抑える意味でも大変有効である。これはクライアント側のシステムは最低限ブラウザが起動できるものであれば利用可能だからである。本システムにおいてネットワークはインターネットを利用し、LAN の他、電話回線を利用した接続も可能となる。現在、携帯電話を利用したモバイルと呼ばれる携帯データ通信も可能となり、現場での利用に大きく貢献できるものと考えられる。

本システム特徴を以下に示すことにする。

検索

本システムは外部からの検索により必要とするデータをブラウザ上で手軽に入手できるよう配慮を行って構築している。検索(図 3.14)は橋梁データ検索における主要な 6 つのフィールドに条件を与えることにより実行できる。

それぞれの項目における条件は、リストボックス・ボタンなどマウスによって選択(図 3.15)できるようになっている。これは記憶が曖昧な場合に大まかな検索が可能となる。また詳細な指定もすることができ、目的とするデータに素速く到達することができる。

閲覧

検索によって抽出されたデータは図 3.16 のように表示される。検索条件に該当するデータは全て表示され、詳細なデータを要求する場合、橋梁番号をクリックすることにより図 3.17 のような画面で表示される。

以上のようなネットワーク上での検索システムは dbWeb によって細かくカスタマイズすることができる。今回はネットワーク上でのデータの入力・削除機能は省くことにした。これはデータの不正書換えによるデータの信頼性を低下を防ぐためであり、閲覧のみのシステムが安全であると考えたからである。

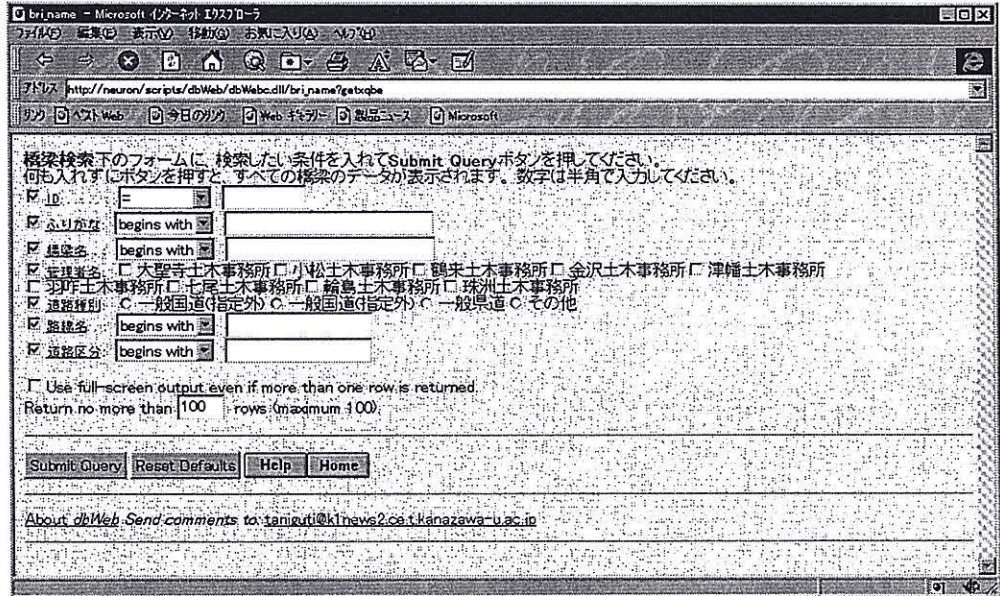


図- 3.14 検索画面

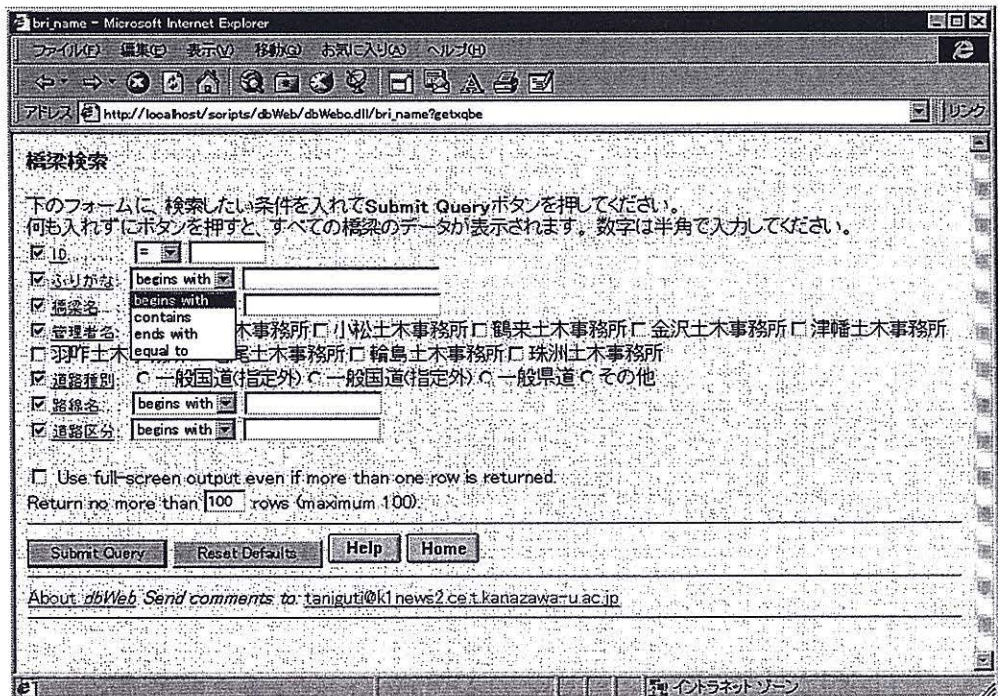


図- 3.15 選択肢表示

ホーム ページ - Microsoft Internet Explorer

アドレス http://localhost/scripts/dbWeb/dbwebcdll/bri_name?getresults

4001	無名 NO1	金沢土木事務所	一般国道(指定外)	304号
4002	無名 NO2	金沢土木事務所	一般国道(指定外)	304号
4003	動清橋	金沢土木事務所	一般国道(指定外)	304号
4004	無名 No.3	金沢土木事務所	一般国道(指定外)	304号
4005	不動寺橋	金沢土木事務所	一般国道(指定外)	304号
4006	無名 No.4	金沢土木事務所	一般国道(指定外)	304号
4007	無名 No.5	金沢土木事務所	一般国道(指定外)	304号
4008	廻落橋	金沢土木事務所	一般国道(指定外)	304号
4009	廻落橋(旧橋)	金沢土木事務所	一般国道(指定外)	304号
4010	無名 No.6	金沢土木事務所	一般国道(指定外)	304号
4011	三保湊橋	金沢土木事務所	一般国道(指定外)	304号
4012	清水谷川橋	金沢土木事務所	一般国道(指定外)	304号
4013	桐木橋	金沢土木事務所	一般国道(指定外)	304号
4014	高落橋	金沢土木事務所	一般国道(指定外)	304号
4015	無名 No.7	金沢土木事務所	一般国道(指定外)	304号
4016	葦の口橋	金沢土木事務所	一般国道(指定外)	304号

図- 3.16 検索結果表示

橋梁台帳1 - Microsoft Internet Explorer

アドレス http://localhost/scripts/dbWeb/dbwebcdll/bri_name?linkresults/obj/%B%B4%97%K%D%831%81%B%83%8E/col/ID/dat/1001

検索結果

ID	1001	管理者	大聖寺土木事務所	道路種別	一般国道(指定外)	道路区分		
ふりがな	むねいぎょう1	路線名	305号	所在地	加賀市	架橋位置	河川	
構築名	無名No1橋	所在地	加賀市	架橋位置	河川			
改修 道路	改修済	昭和 49	改修 河川	改修済	昭和 49	確率年数	0.00	
交通量 現在	昭和 52	自動車	7736	重車両	755	自転車	170	
		自動車	0	重車両	0	自転車	0	
交通量 将来	0	自動車	0	重車両	0	自転車	0	
歩行者	136	歩行者	0					
計西流量(%)	0.00							
構種	単線RC床版橋							
支間割	0.00	0.00	0.00	0.00	0.00	0.00	橋長 0.00	
							橋面積 0.00	
幅員構成	左歩道	0.00	左路肩	0.50	左車道	2.75	中央帯	0.00
	右歩道	0.00	右路肩	0.50	右車道	2.75	全幅員	6.50
							歩道幅員	有
							歩道幅員	有
下部工および基礎工	A1	重力式	P1				P2	
	P3		P4				P5	
	P6		P7				P8	
	P9		P10				P11	

図- 3.17 詳細データ表示

3.3.5 考察

システムをパーソナルコンピュータ上で構築することにより、以下の成果が得られた。

1. 橋梁データをコンピュータ上で管理することにより維持管理の簡便化、一層の操作性の向上が図られた。
2. システムを Windows 上で構築することにより他ソフトウェアとの連携が容易になった。
3. システムのネットワーク化を進めることにより、リモート操作が可能になった。

本研究の目的の一つである、橋梁台帳・橋梁点検台帳等の必要な各種データをコンピュータ上でデータベース化しシステム構築することは、結果として、上記の成果の他、データ処理の向上、統一化されたインターフェースによるユーザフレンドリな環境の構築、また、データベースサーバーを核としたネットワークを形成することによって分散処理が可能になるなどの利点をも生んだ。このように、パーソナルコンピュータ上でのシステムの構築は、予想されうる恩恵以上に、幅広い利用形態に発展できる可能性を秘めるものであり、今後のパーソナルコンピュータ上でのデータベースシステムの整備は維持管理業務の大きな進歩になるものと考えられる。これを土台とし、橋梁損傷要因分析、橋梁補修計画等のプログラムと連携し解析できるようなシステム構築を推進し、また GIS システムについても現在、Windows 対応のものが多く存在するようになったことから、これらの連携を念頭に置いたシステム開発が望まれる。また、ネットワーク環境が急速に整備されている今日、インターネットを考慮したシステムは低コスト化、情報の公開などによる利用形態の広域化が進むものと考えられ、結果として維持管理システムをドラスティックに変化させる可能性がある。このようなことからネットワークシステムへの対応は大胆かつ慎重に推し進めていくべきであり、またネットワークシステムの考慮は、これからのデータベースシステムの必要条件になるであろう。

本システムは、データの登録、閲覧、印刷などの基本システムにインターネットを含むネットワークを利用した閲覧・検索機能を加えることにより、柔軟性のあるシステムとして構築できた。現在、基本システムは実際に I 県庁で橋梁データの管理に活用されていることから、完成度の高いシステムが構築できたものと考えている。また橋梁データの維持管理をするうえで一つの構築例を示すことができたと思う。

今回紹介した橋梁データ管理システムは、上下水道、道路をはじめとした台帳管理を行っている他の業務分野でも同じように活用できると考えられ、ここで示したアプローチのより広い応用が期待できる。

次章に GIS システムとの連携を配慮し検討を行った構築例をあげる。

3.4 GIS との連携

3.4.1 導入

地理情報システム (GIS)¹は、地図を媒介にした様々な情報を総合化するコンピュータシステムのことである。地図には、行政区域、道路、河川、鉄道、各種施設といった位置を持っている基本となるデータ (空間データ基盤) が存在する。これと統計データ、各種台帳などの位置を持つデータ (空間データ) を併せて利用することで、各種維持管理システム、防災システム、教育システム、公共情報検索システムなど多方面で効果を発揮している。特に、コンピュータが普及し、ネットワーク化が急激に進んでいる今日、情報処理の分野における地図情報システムの進展が、パーソナルコンピュータレベルでの地図情報の取扱いも容易なものとしている。さらに、コンピュータで処理できる形態の地図データも、国土地理院をはじめ、カーナビゲーションシステムの一般への普及などにより民間企業からもデータ量が豊富でしかも質の良い「地図データ」の整備が鋭意進められており、低コストで地図情報を利用できる環境が整いつつあると言える。

土木計画分野では、分析、立案、結果の評価に際して地図情報が極めて有用であることが十分認識されつつも、そのハンドリングの煩雑さなどから多くが表計算ソフトなどによる数表整理にとどまっており、豊富なデータを有効に利用し切れないのが現在の状況である。

本研究はパーソナルコンピュータ上で橋梁データ管理システムの橋梁データと地図上の位置と情報とを結び付け、データの相互利用や一層の土木計画業務の質的向上できる環境整備を目標に研究を進める。

3.4.2 本システムとの連携

はじめに

GISとは、「地域から発生する様々な情報を取得し、データベースを作成し、それらデータを必要に応じて引き出し、加工や分析を行う一連の処理過程」と定義することができる。そして、その分析結果はコンピュータグラフィックスによって地図として出力される。しかし、GISの最大の特徴は、地球上の位置を示す地図データとその空間が有する属性データとの有機的結合にあるといえる。この点において、GISが単に大量データ処理の省力化・合理化あるいは瞬時の地図化に対してのみ機能しているのではないことを強調したい。

¹Geographic Information System の略で、日本語では地理情報システムと訳されることが一般的である。コンピュータ上で、地図の作製・保存から、それをもとにした空間分析まで、空間における様々な問題を取り扱うことができ、地理学をはじめ、都市計画や環境保護にいたる多くの分野で利用が行われている。

GIS 利用におけるメリット

橋梁データ管理システムを基礎として GIS を利用するにあたり、3つの利点があげられる。

1. 地図上に情報を表示

情報を地図上で参照することにより、地理的位置との関連性などが視覚化され、情報に対して地理的（エリア的）な思考が可能になる。

2. 地図の特性を利用した情報活用

ある路線の交通量情報、架橋位置における損傷度の変化など、位置情報・位置関係の把握ができ、地図を抜きにしてはなしえない分析が可能になる。

3. レイヤーを用いたデータの統合管理

様々なデータをそれぞれのレイヤー上(図 3.18)で管理することにより、独立していたデータと他データの関連性を視覚的に見いだせる。

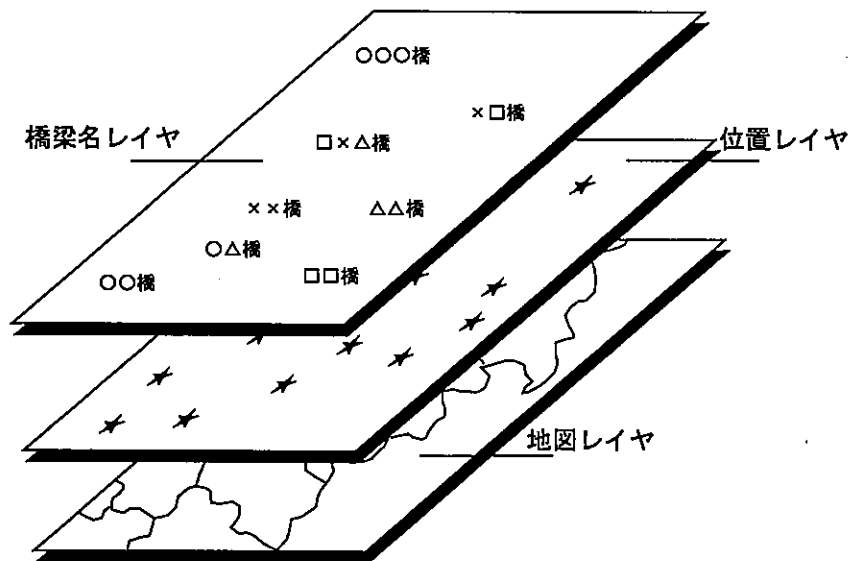


図- 3.18 レイヤー図

MapInfo Desktop-mini

本研究では橋梁データを地理データと連携させるためにソフトウェア「MapInfo Desktop-mini」を使用した。これは GIS ソフトウェアとして定評のある MapInfo Professional を開

発している MapInfo 社の製品であり、MapInfo Professional がハイエンドユーザーを対象とし、高度なシステムを要求するのに対し、本製品は前者と比較すると機能が大幅に制限されるが、手軽に PC 上でデータを分析でき、低予算でシステムを構築する場合に有効である。

地図データについては、個々の点・線の座標が数値化されている「ベクトル形式」と、地図が点列から構成される絵として数値化されている「ラスター形式」とがあるが、今回、ソフトウェアに使用した数値地図は、付属されるベクトル形式の 2 万 5 千分の 1 の地図を利用することにした。

また今回は本ソフトウェアのデータアクセス機能に特に注目した。

MapInfo Desktop-mini(以下 MapInfo) では、Microsoft Excel, Lotus 1-2-3, Borland dBASE, ASCII データなどを読み込んでデータの分析を行える。これは橋梁データ管理システムとデータが共有できることを示し、橋梁データ管理システムを基軸にした管理支援システムの構築が可能となる。また、緯度経度からの地図上への点の作成、データの地理的分布状況図の作成、地図上の任意の図形内に含まれるデータの検索集計などの機能も含み、データを様々な地理的観点から分析することが可能である。

このように本ソフトウェアは本章の目的に十分な機能を兼ね備えており、満足する結果が期待できることから採用した。

データの共有化

MapInfo 上において Access で構築した橋梁データベースを利用するためには、それぞれ各橋梁データに対し「位置データ」を与えなければならない。位置データとは所在地の住所や、その緯度・経度から得られる XY 座標データであり、地図において空間基盤データと呼ばれるものである。本研究ではより正確で確実な位置データを入力するため、橋梁の所在地を緯度・経度によって表す XY 座標による入力方法を選択した。それに伴い、橋梁データベースに XY 座標のフィールドを設定し対応することにした。これは、XY 座標フィールドを軸としたクエリーによって、必要とする橋梁データが自由にエクスポートでき、瞬時に分布図を作成することは極めて容易になることを意味する。

今回対象となった I 県の橋梁データは 176 橋である。これは橋梁台帳・橋梁点検台帳のデータが両方存在するもののみを選択した結果である。以下に GIS ソフトウェアを利用したシステム例を報告していく。

3.4.3 地図上でのデータ表現

地図上でのデータ表現を行うためには、地域の人口構成などといった社会経済指標や、鉄道、路線といった交通インフラに関するデータなど土木計画分野の基礎的データを地図上の位置と結び付けて管理する必要がある。これに対応してデータを地図上の位置と結び付けて一元管理する例を報告する。これらは社会経済指標などの数値データや、交通インフラ関連の座標データはもとより、文章などのデータを地図上の位置に結び付けることも可能である。以下に Mapinfo と橋梁台帳・橋梁点検台帳を連携させた例を示す。図 3.19 はI県の176橋梁の分布図となっている。このような処理はXY座標を入力しているため、容易に可能とする。

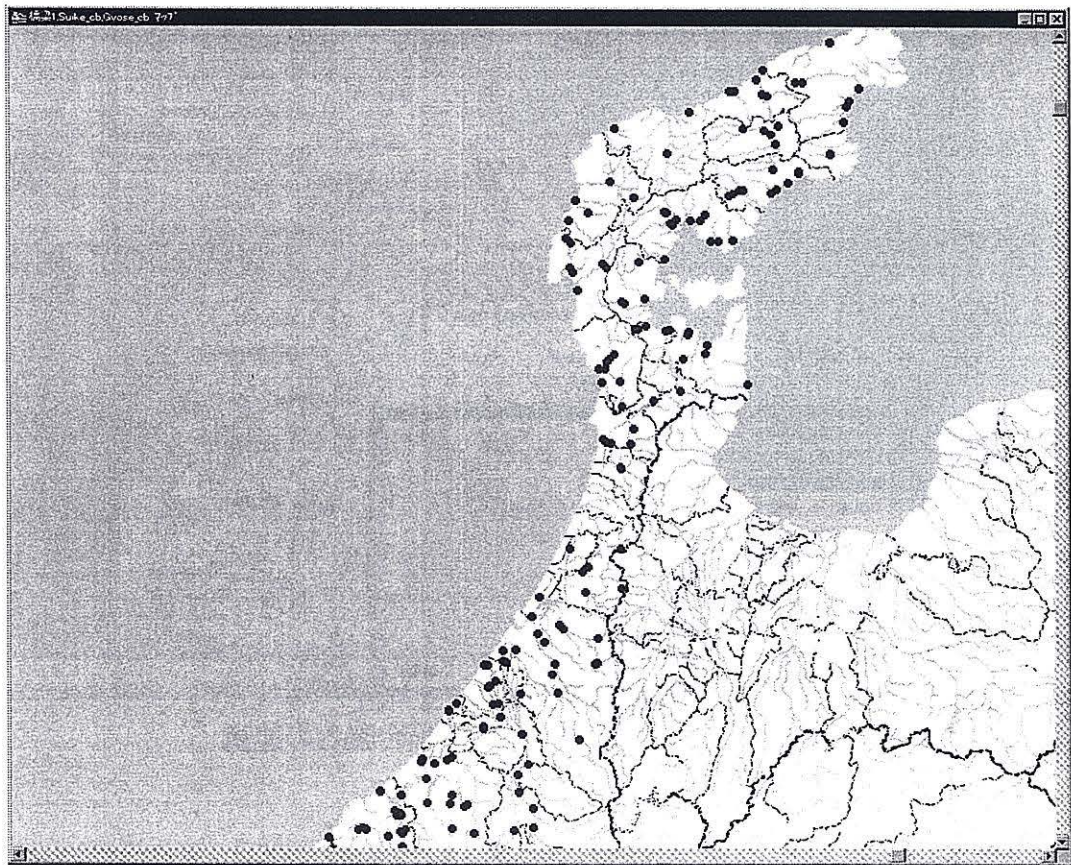


図- 3.19 橋梁台帳との連携

図 3.20 は橋梁台帳と地図データを連携させたものである。ソフトウェアの機能より手取川に架橋されている橋梁を抽出した画面である。テーブルからは7つの橋梁が抽出され、抽出された地図上の橋梁の位置は赤くマークされる。またその中の橋梁の情報を知りたい場合、位置をクリックすることにより画面下に表示される詳細データが得られる。

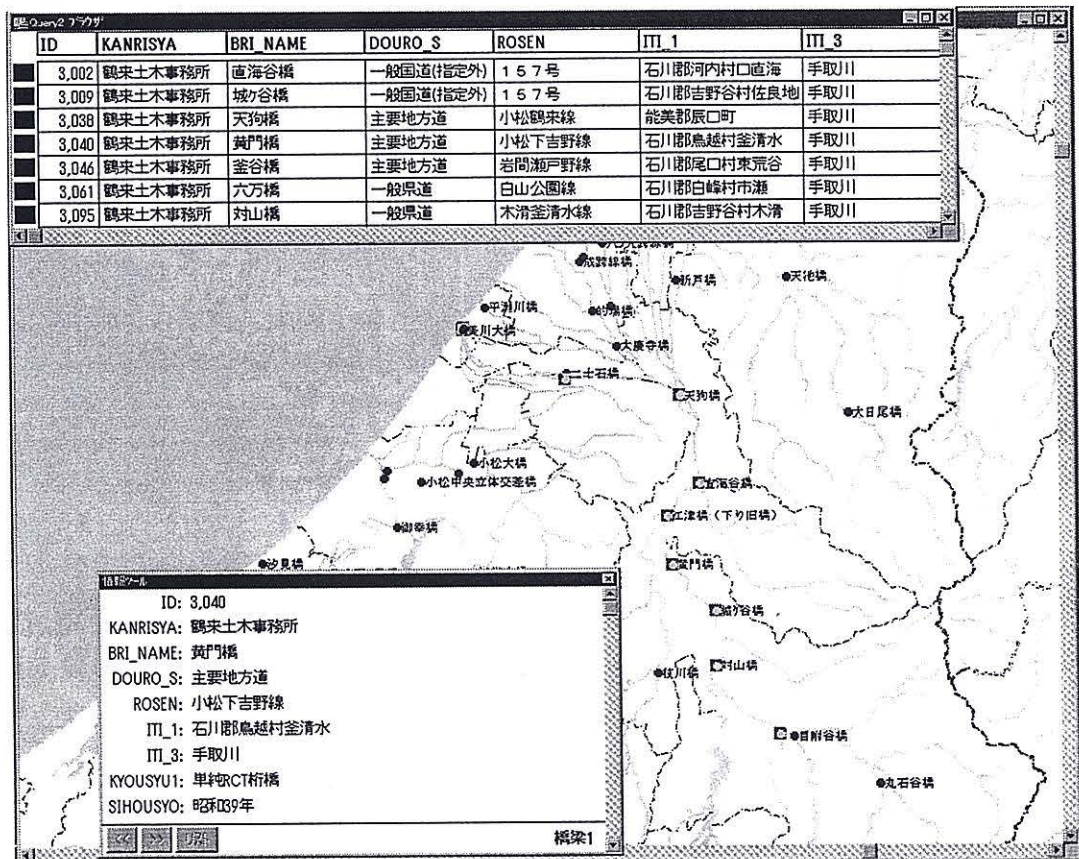


図- 3.20 橋梁点検台帳との連携

図 3.21 は橋梁点検台帳との連携を試みたものである。画面は検索機能より橋梁の総合損傷度が1、つまり A のものを抽出したものである。これも図 3.20 と同じように赤くマークされる。画面右下のウィンドウはある橋梁をクリックしたもので、その詳細が表示される。このように地図データとの連携は橋梁データを全体をみながら視覚的に把握できる利点がある。これは橋梁損傷とエリア的思考が可能となるものである。

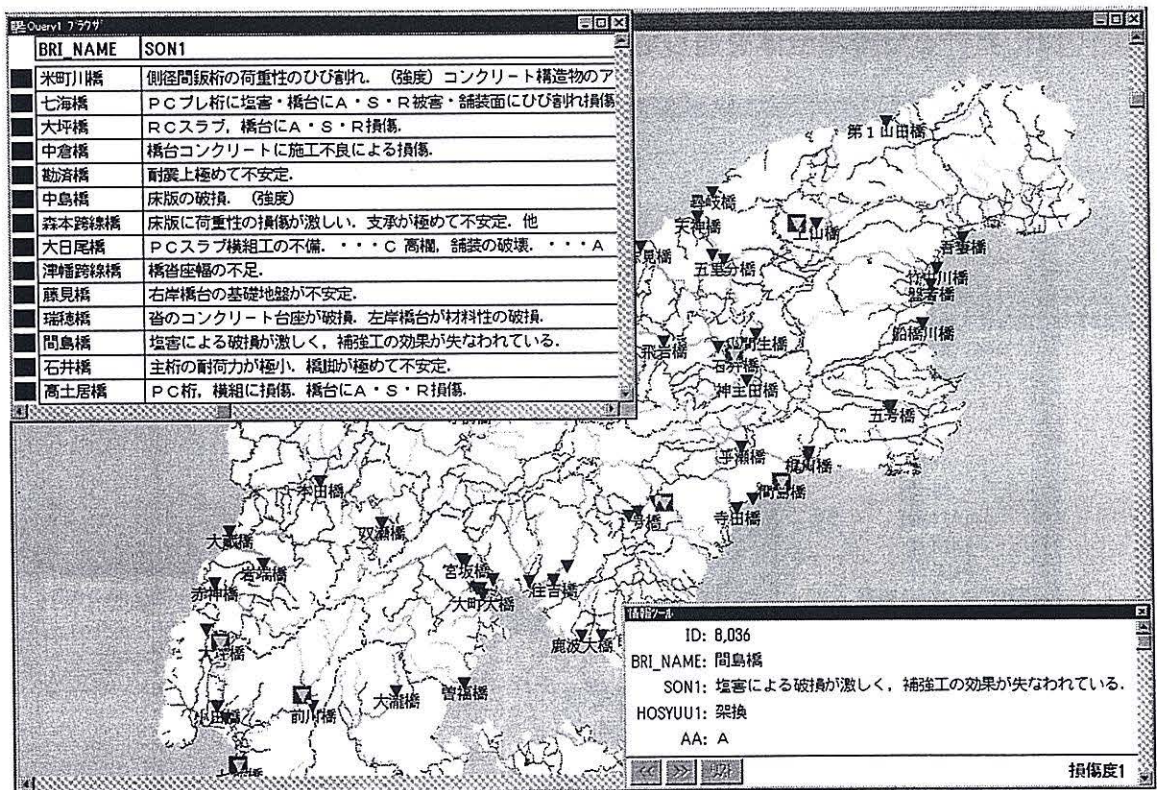


図- 3.21 I 県の橋梁所在地

3.4.4 地図上でのデータ管理

道路における通行車両数、地域間の車両流動など、土木計画分野には地理的な位置と結び付けられる調査データが多い。データ分析の段階で、これらのデータをグラフなどの手段で視覚化すると共に、地図上に空間的に配置することができれば、地域特性の理解など種々の面でデータの特性の理解が一層容易になることが考えられる。今回、MapInfoの機能である主題図によるデータの分析を利用し、データに関連づけ表現した例をあげる。

図 3.22 は橋梁損傷度を画面右図のように固有値別に色分けしたものである。それぞれの値における分布が全体を通して確認できる。

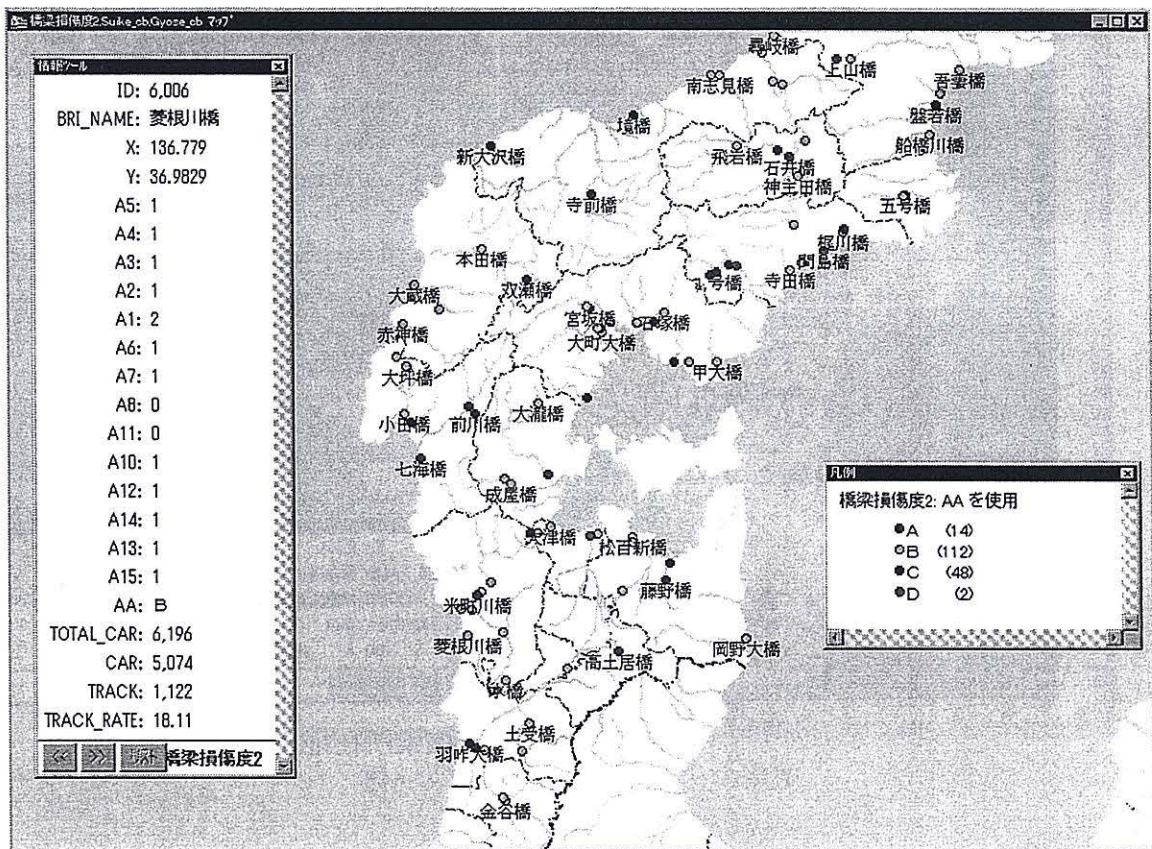


図- 3.22 固有値別橋梁損傷度

図 3.23 は各橋梁における通行車両数を棒グラフに示したものである。損傷度と車両数との関係が位置的に表示される。また大型車と普通車の割合と損傷部位との関連を視覚的に表現でき、データの管理が分析と平行して可能である。

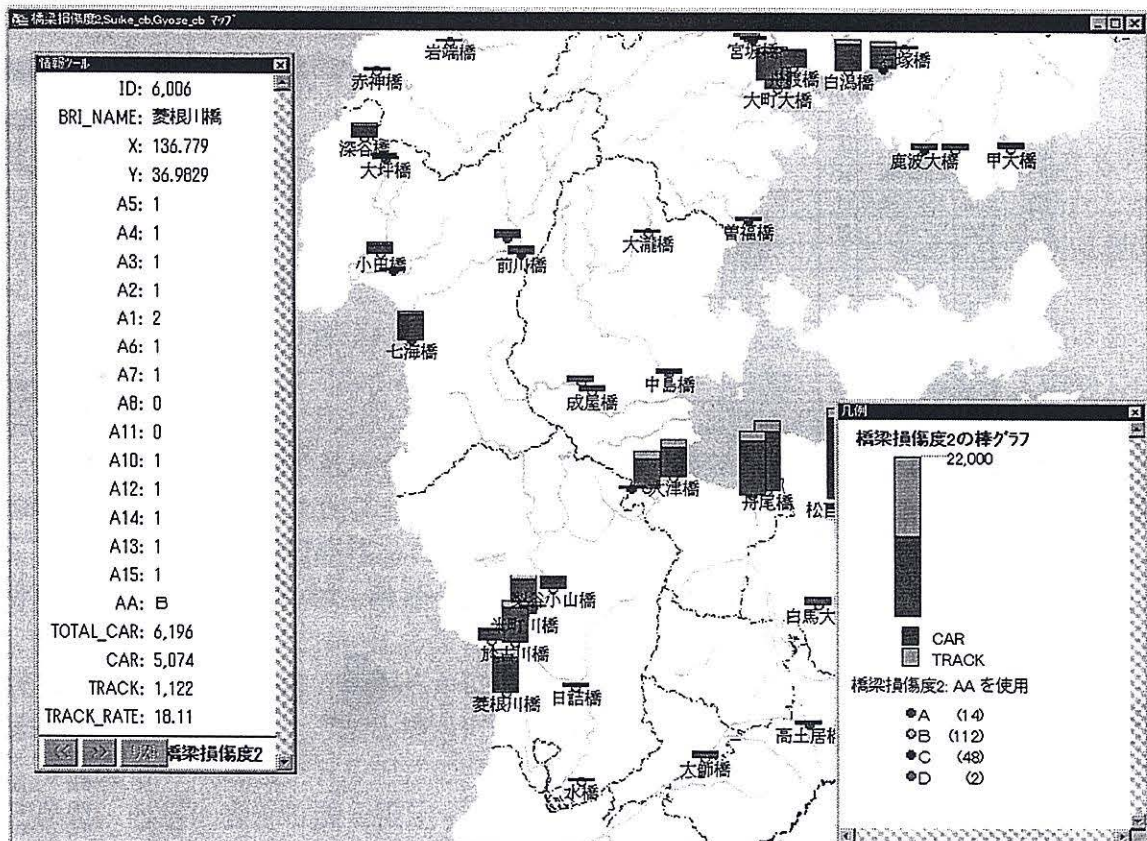


図- 3.23 各橋梁における通行車両数

図 3.24 は総通行車両数に対する大型車両のパーセンテージとその損傷度を示している。また 25 パーセント以上のものを抽出した結果が画面左上のウィンドウであり、損傷度との比較が可能になる。このように今後、データの充実を推進することにより様々な種類のデータが地図上で一元的に管理でき、分析の幅が広がるものとする。

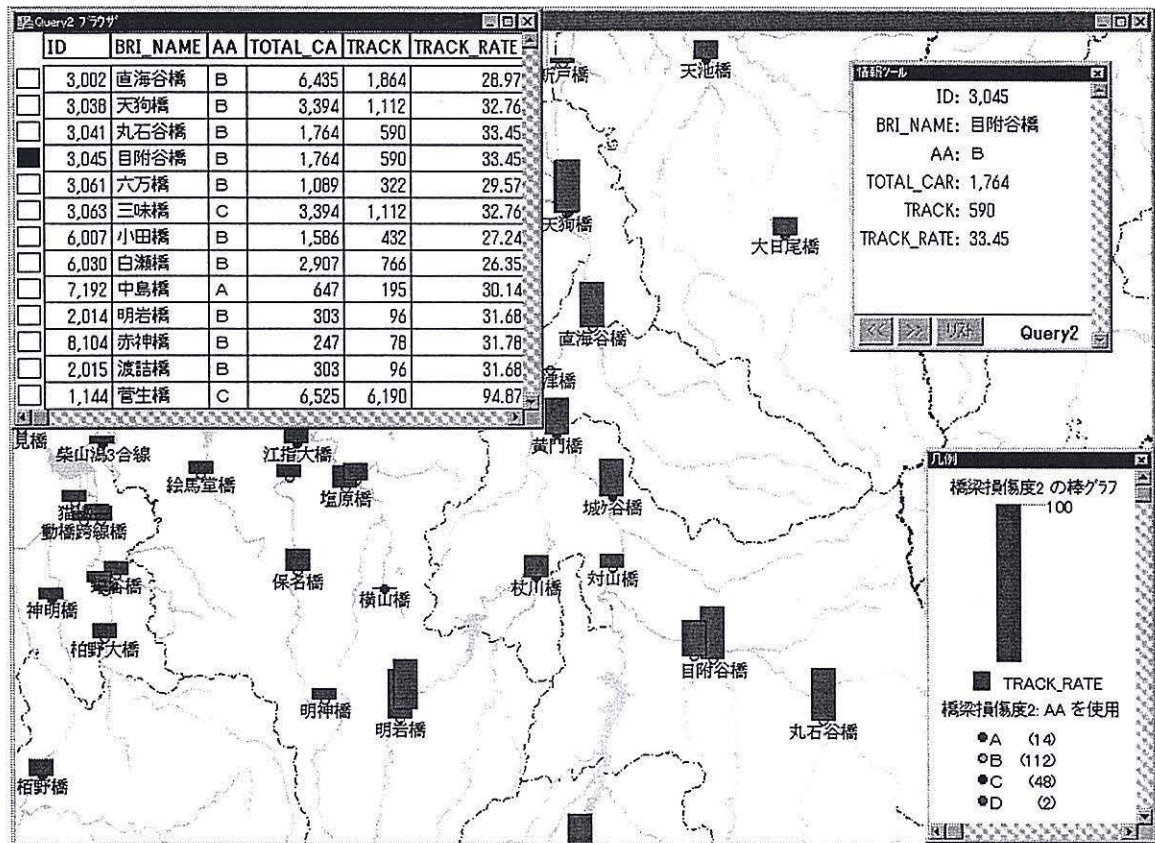


図- 3.24 大型車の割合と損傷度

3.4.5 考察

GIS ソフトウェアとの連携により得られた利点を以下に示す。

- 数的データに対し、エリア的な思考が可能となった。
- データの柔軟性が一層向上した。
- データの維持管理がレイヤーの使用により容易になった。

重要視すべき利点は各データの視覚的表示にある。単に数・文字データであったものが、地理データと連携させることにより地理的位置との関連性などが視覚化され、新たな解析が可能となった。またその解析についても様々な解析手法がソフトウェア上で利用者の要求に応じて簡単に実行できるものである。

レイヤー上でのデータ管理についても、各種インフラに関するデータと関連づけることにより、橋梁データだけでなく統合したデータ管理ができ総合システムとして構築できる可能性を秘めている。このように GIS ソフトウェアとの連携は維持管理業務の効率の向上と、発展に大変有効であろう。

本研究では、GIS ソフトウェアである MapInfo Desktop-mini と廉価な地図データを利用し土木分野での利用可能性を具体的に橋梁データと連携させることで検証した。

今回の連携させる試みは、数値データをより一層、生きたデータに変化させるものであり、管理支援システムにおいて多大に有効であることが分かった。特に視覚的にデータが検証できることは、従来では明らかにならなかった事実を発見するきっかけを与えるものと思われる。今回試みた、ある橋梁における通行車両数のデータをその地点でグラフ化するような場合でも、ベクトル形式の数値地図に橋梁そのものの位置が座標化されていることにより、容易に表示位置を指定することができる。

これは土木分野において実際に地図が利用される局面では、位置と関連付けたデータの管理や分析結果の表現手段として利用される機会も多いこともあり、そうした場合に数値地図のような廉価なデジタル地図を利用できる可能性のあることが示されたものと考えられる。しかし、ソフトウェアにおいて機能的に貧弱な面は否定できず、限定された機能のなかでの解析、システム開発の限界は感じられた。橋梁台帳・点検台帳データと地図データの連携における大きな可能性が見い出されたなかで、本格的な GIS システムとの連携に期待される。

3.5 結論

3.5.1 本システムの有効性

本研究は Access で構築した橋梁データベースを基本システムとし、ネットワークを考慮したシステムに発展させ、さらに試みとして GIS ソフトウェアとの連携を図った。これらシステムの発展は、既存の紙上管理されていた橋梁台帳・橋梁点検台帳をパーソナルコンピュータ上で電子化し、現在幅広く利用されている OS である Windows 上でデータベースを構築することにより実行できたと考えてよい。

現在、数多くのデータベース維持管理システムが様々な形で研究・提案されているが、その中には大規模なシステムを要求するもの、プログラムとして構築されているものなどがある。それぞれのシステムには一長一短があるが、本研究のシステムの有効性について以下に述べる。

まず第一に既存の環境をできるだけ有効に活用できるようなシステムを構築できるということである。パーソナルコンピュータの性能が向上し、それらが多くの職場で普及した現在、その環境を有効に利用しネットワークを形成することにより、大規模システムに劣らないデータベースが構築できるものとする。また無駄なコストを抑えることができるという利点もあげられる。

第二にユーザーに対し専門的な知識なしで利用できるようなユーザーフレンドリーなシステムを構築できることがあげられる。本研究では市販のデータベースソフトである Access97 を使用している。これは第一の条件を満たすとともに、統一化されたインターフェースによって多くのユーザーが利用しやすいなどの利点を生むものとする。

第三に汎用性のあるソフトウェアを利用することでデータの共有を容易にできるということである。システムのプラットフォームを Windows にすることにより、数多く存在する Windows ソフトウェアとの連携が可能となり、新たなシステムの開発や移行が比較的容易にできる。本研究では GIS ソフトウェアとの連携を試みたが Windows 上でのデータ共有の容易さが開発を比較的簡単なものとさせた。これは橋梁損傷要因分析などの連携にも移行しやすいものと考えられる。

これから橋梁の維持管理業務が増加する事は自明であり、特に各種データの保管・運用は重要な業務となるであろう。その課題に対して、数多くの研究や開発が進むなかで、本研究で示した橋梁データ管理支援システムは一つの方法として提示できるものとする。

3.5.2 今後の課題

データベースを構築するにあたって最も重要なことは、なぜデータベースが必要なのかを利用者の立場に立って考え、設計することである。これは、利用者の情報要求を分析し、

適切な情報の発生を調査し、情報を収集し、整理、分析、加工し、いつでも検索できるように、蓄積し、絶えず変化する情報要求と情報の間でコミュニケーションをはかりながら提供しなければならないことを意味する。しかし現実問題として橋梁に関するデータは、本来、維持管理に役立てるという目的で橋梁台帳や橋梁点検台帳に保管しているはずであるが、これらのデータを十分活用して維持管理に役立てていないのが現状であった。これらを考慮した上で、本研究では橋梁の維持管理をスムーズに進めるために、橋梁データを管理するシステムが維持管理への有効利用という立場に立ってシステム構築を遂行し、橋梁データ管理支援システムという形で提案してきた。これは、データベースをベースとして幅広い利用形態に対応すべく、システムを構築できたものと考えている。しかし研究を進める上で新たな課題が生じてきたのも事実である。三つの課題を以下に述べる。

まず第一に GIS システムへの連携の強化である。本研究では MapInfo Desktop-mini という廉価なソフトウェアを用いて連携を模索したわけだが、地図データを利用した橋梁データの管理・解析にある程度の成果を見せた半面、やはりソフトウェアの機能的に貧弱な面は否定できず、限定された機能のなかでの解析、システム開発の限界は感じられた。橋梁台帳・点検台帳データと地図データの連携における大きな可能性が見い出されたなかで、本格的な GIS システムとの連携に期待される。

第二にデータの統一化に関する問題である。橋梁台帳・点検台帳には数種類のフォーマットが存在しデータを入力する場合、それぞれの項目にばらつきがあり、データの入力の際、不備を感じた場面が多くあった。また、橋梁のデータを管理する台帳自体が架設後の維持管理への活用を念頭においた設計になっておらず、ただデータを保存しているだけであると感じた。このように、これから橋梁データをパーソナルコンピュータ上で一元管理していく時代において、フォーマットの統一化は橋梁維持管理業務を遂行していく中で、早急に解決していかなければならない。これは損傷要因分析におけるデータの信頼性も向上させ、解析に関し有効に働くものと考えられる。

最後にシステムのプラットフォームについての問題である。現在パーソナルコンピュータの大部分は Windows を採用しており、本研究においても Windows を OS としてのシステムを構築している。しかしハード・ソフトの絶え間ない技術開発が進行する現在、現状のシステムが継続していくとは考えにくい。データベースは変化する環境の中で、データの品質を維持し、利用者の動向を探りながら、いち早く将来を予測し、新しいデータベースへの脱皮を決断していかなければならない。このようななかで、最近注目を浴びている UNIX をベースとした Linux, Free-BSD などの OS は堅固性、柔軟性などに関し、有利な点を兼ね備えており、今後のデータベースシステムの展開に際し、考慮すべき点であることは間違いないものと思われる。これは将来、数種類の OS が存在する時代が訪れることを予感させ、マルチプラットフォームに対応したデータベースへの移行を見据えた構築が必要になるとと思われる。

参考文献

- 1) 山岡健一他：「Bridge Management System(BMS)における最適維持管理計画の策定」，土木学会第52回年次学術講演会，pp.228-229，1997.
- 2) 荻原勝也他：「橋梁マネージメントシステムの開発」，土木学会第51回年次学術講演会，pp.802-803，1996.
- 3) 伊藤義人：「LIFECYCLE MANAGEMENT OF NETWORK-LEVEL BRIDGES」，NUCE Research Report，pp.127-134，1997.
- 4) 戸島武彦：「橋梁データ管理システム'96の構築に関する研究」，金沢大学卒業論文，1996.
- 5) 藤田節子：「データベース設計入門－利用者の立場から」，紀伊國屋書店，1992.
- 6) 近田他：「データベース構築とデータ分析プログラムのシステム構築に関する研究 (MS-DOS版)」，第16回土木情報システムシンポジウム論文集，1991.
- 7) 早川国男：「橋梁維持管理支援システムの構築に関する研究」，金沢大学卒業論文，1994.
- 8) 木内里美他：「作業所におけるイントラネットの導入」，第22回土木情報システム講演集，pp.191-194，1997.
- 9) 中山敏夫他：「道路台帳のCD-ROM化」，第22回土木情報システム講演集，pp.75-78，1997.
- 10) 立石 亨：「土木分野における数値地図利用の一形態について」，第22回土木情報システム講演集，pp.21-24，1997.

第4章 橋梁管理データベースと補修計画立案支援の連携

4.1 リンクアプリケーションの作成

本章では、既に述べたデータベース管理システム(以下 DBMS)と橋梁補修計画作成システム(以下 GAS)とを繋ぐ新しい橋梁補修作成システムのリンクアプリケーションを構築する。

4.1.1 アプリケーションの設計

リンク構造

前章で述べた、DBMSとGASを繋ぐシステムの構想は、以下の3つのパターンが考えられる。図-4.1に構図を示す。

(1) DBMSにGASが含まれるシステム。

- ・ DBMSにおいてGASの要求する点検項目のデータを決めておき、GASに供給する。

(2) GASにDBMSが含まれるシステム。

- ・ GASにおいて必要な点検データ項目を決めておき、DBMSから書き出す。

(3) DBMSとGASの間に存在するシステム。

- ・ DBMSとGASが、それぞれ別々に存在し、DBMSからはデータを出し GASからはその出力されたデータを読み込む。

これら3つのシステムは簡便性、機能性、信頼性、安全性、発展性について考察すると、簡便性、機能性、信頼性はプログラム作成上の問題であるので、どのシステムが良いと

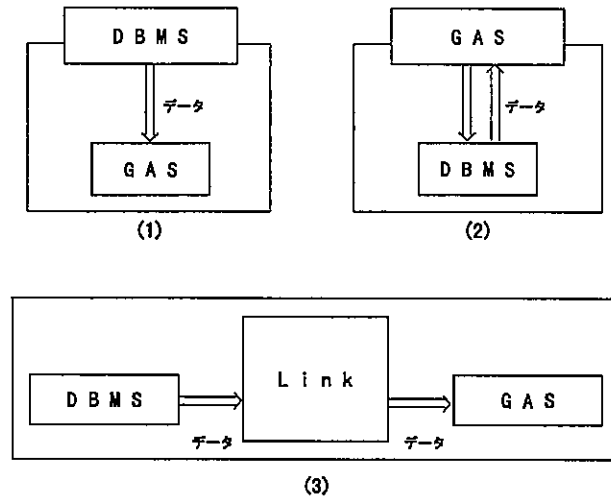


図- 4.1 DBMS と GAS のリンク構造

は言い難い。しかし、安全性において、(1)、(2)は直接データを受け渡しするのに対し、(3)は一度ファイルに保存してデータの流れを止めるため、データの受け渡しをする場合には安全である。

次に、発展性については、以下に示す通りである。

a) DBMS の橋梁データ (GAS に関係しない) が増加した場合

(1)、(2)、(3) のどの場合も DBMS のみで処理できる。

b) GAS の計算に必要なデータ項目が増加した場合

- (1) のとき、DBMS において必要なデータ項目を決めるため、DBMS と GAS の両方の改良が必要になる。
- (2) のとき、GAS において必要なデータ項目を決めるため、GAS だけの改良が必要になる。
- (3) のとき、DBMS と GAS が別々に存在しているので、この場合、GAS だけの改良が必要になる。

以上のことから、リンクアプリケーションを作成していく上で DBMS と GAS の関係は、安全性、発展性を考えると (3) の形式が良いと考えられる。そこで本研究では、(3) の形式を用いたリンクアプリケーションを作成する。

要求される機能

リンクアプリケーションの構築にあたり、システムの概要と必要な機能を以下に示す。

システム概要 このシステムはI県の橋梁に関するデータを利用し、I県の橋梁補修計画代替案を作成する。橋梁に関するデータのほとんどはデータベースに入力されており、そのデータを用いて本システムは補修計画作成に必要なデータを橋梁ごとに表示する。また、補修の対象となる橋梁を選出するための検索機能を持つ。加えて、検索されたデータをファイルに出力し、保存する。そして、保存されたファイルのデータを用いて、様々な条件設定のもとで補修案を作成する。

必要な機能 システム概要より、必要となる機能を大まかに示せば、以下のようになる。

- 橋梁補修計画作成に必要なデータをデータベースとリンクして表示する。
- 表示されたデータを項目の値により検索し、該当した橋梁だけを表示する。
- 検索されたデータをファイルに出力し、保存する。
- 保存されたデータを利用し、補修案を決定する。

開発環境

開発言語 現在市場には、BASIC, C, C++, Pascal, COBOLなど非常に多くの開発言語が存在するが、本研究ではC++を開発言語とする。これは、C++が非常に柔軟性の高い言語であるとともに、多くの開発現場でC言語そしてC++が利用されていることが理由である。また、本節の中でも説明するが、インプライズ社(旧、ポーランド社)のC++BuilderがリリースされていたこともC++を開発言語に選んだ大きな理由である。

プラットフォームの選択 開発環境の選択は、システム構築において非常に大きな影響を及ぼす。開発環境の操作性、容易性が低い場合、優れた操作性、容易性を持った環境で開発する場合の何倍もの時間を費やされることも珍しくない。そこで本研究が開発環境のプラットフォームに選択したのは、マイクロソフト社のWindows 95/98およびWindows NT4.0である。プラットフォームの選択にはUNIXマシンや汎用機なども考えられるが、現在のコンピュータ市場を見る限り、サーバーマシンや一部の業種を除けば、各企業の従業員やオペレータが直接触れる機会が最も多いのは、Windowsオペレーティングシステム以外には考えられない。もちろん多くの現場でその他のアップル社のMacOSやX-WINDOWSなどのUNIX端末も稼働しているが、それでも今後も多くの市場で、端末装置や、小規模、中規模のシステムのOSとしてWindowsが多数を占めることは間違いのないと思われる。

C++Builder 開発環境の決定においては、プラットフォームの選択も重要な要素であるが、さらに重要なのがシステム構築の要となるコンパイラを選択である。そこで本研究で開発環境に選択した C++Builder は、C/C++の柔軟さを持ち、予め用意されたビジュアルコンポーネントライブラリ (VCL) コンポーネントをドラッグ&ドロップするだけで、ユーザーインターフェースを作成することができ、さらに各コンポーネントが生成するイベント関数を自動生成する機能を持っている。これらの機能は、C++の非常に面倒な画面定義とコードの結合を隠蔽して開発者の負担を少なくし、かつ OWL(Object Windows Library) に見られる複雑なクラスライブラリの利用を簡単にすることができる。これにより、C++Builder は非常に高い生産性を持っている。さらに、これはインプライズ社の Borland デスクトップ製品全般に言えることであるが、BDE(Borland DataBase Engine) が実装されており、データベースへのアクセス機能を統一したインターフェースで使用することができる。これらの特徴を生かして、より現実的なシステム構築を目指しリンクアプリケーションの作成を行う。

開発フェーズ

開発ステップは、以下に示すフェーズに分割される。

第1フェーズ

1. データベースにアクセスし、入力されているデータを C++Builder で使用できるようにする。
2. データベースに表示する項目のデータ群を新たに設ける。
3. 表示されているデータ項目の値を用いて検索機能を実装する。
4. 表示されている橋梁分のデータをファイルに出力して保存する。

これらの機能を持つアプリケーションを検索システムとする。

第2フェーズ 既に構築されている GAS におけるアプリケーションと第1フェーズで構築したアプリケーションのプロジェクトの統合を行い、保存されたデータを使用できるようにシステムを整える。

4.1.2 プログラムの概要

プログラムの設計は、先に述べた開発フェーズに従って行った。

データベースへのアクセス

前章に述べたように、実際の橋梁維持管理に用いられている橋梁台帳、橋梁点検台帳は市販のデータベースソフトである Microsoft Access7.0 を用いてデータベース化されている。そこで、C++Builder において Microsoft Access のデータベースファイルにアクセスするためには、ODBC ドライバの設定と BDE 環境設定を行う必要がある。その手順を以下に示す。

(1) Access 用ドライバの確認

Microsoft Access のデータにアクセスするには、マイクロソフト社から提供されている ODBC ドライバが必要である。ドライバの一覧(図-4.2)に、Microsoft Access-Driver(*.mdb)が表示されていれば問題ないが、もし、Access 用のドライバがインストールされていない場合は、そのドライバを入手してインストールする必要がある。

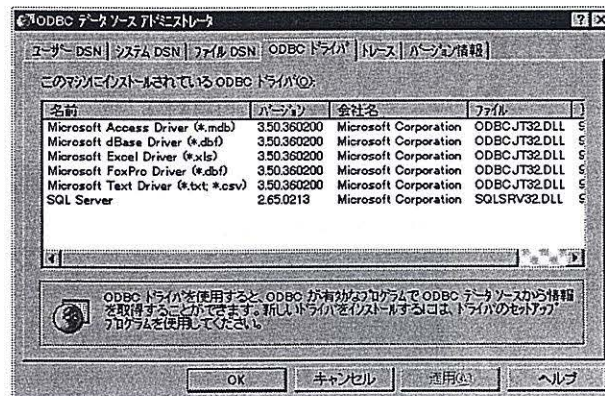


図- 4.2 ドライバの一覧

(2) ODBC データソースの作成

Microsoft Access のデータにアクセスするには、もう一つ、ODBC データソースを設定する必要がある(図-4.3)。データソースの設定は、Windows のコントロールパネルで行う。設定した<データソース名>と<説明>を以下に示す。

<データソース名> Access DBMS

<説明> 総合橋梁 DATA1.mdb

(3) 新規 ODBC ドライバの登録

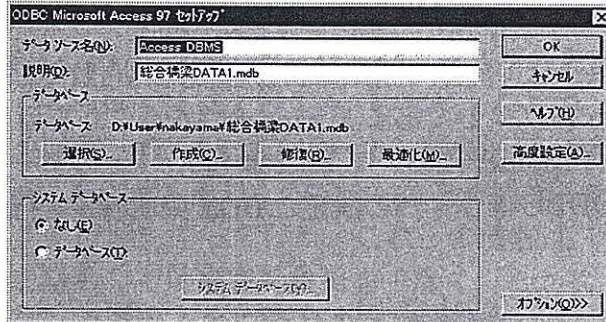


図- 4.3 データソースの設定

C++Builder の BDE 環境設定のドライバページを表示し設定を行う (図-4.4)。設定した<BDE ドライバ名>、<デフォルト ODBC ドライバ>、<デフォルトデータソース>を以下に示す。

<BDE ドライバ名> ODBC_ACCESS_DB

<デフォルト ODBC ドライバ> Microsoft Access Driver(*.mdb)

<デフォルトデータソース> Access DBMS

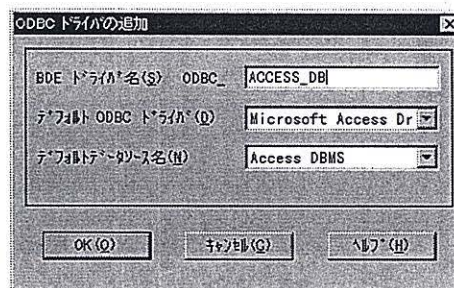


図- 4.4 ドライバの登録

(4) エリアスの設定

C++Builder の BDE 環境設定のエリアスページを表示し設定を行う (図-4.5)。設定した<新規エリアス名>、<エリアスの種類>を以下に示す。

<新規エリアス> ACCESS_DB

<エリアスの種類> ODBC_ACCESS_DB

この作業を終了して、はじめて C++Builder で Microsoft Access のデータを扱うことができる。

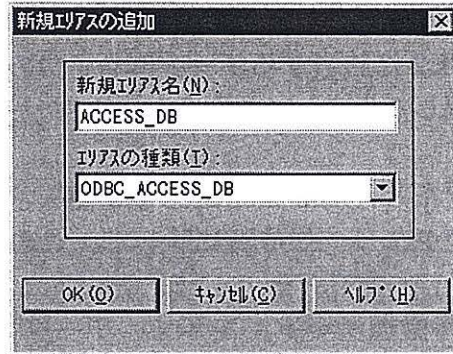


図- 4.5 エリアスの設定

データ表示の機能設定

検索システムに表示するデータは補修計画作成に必要なデータである。データ項目をあげると、表-4.1、表-4.2になる。[ID]と[管理事務所]は、補修計画作成に直接は用いないが、表示される橋梁を認識しやすくするためと、後に述べるデータ検索の項目としての発展性を考えて追加した。

表- 4.1 表示項目 その1

- | | | |
|---------|--------|--------|
| ・ ID | ・ 道路種別 | ・ 橋面積 |
| ・ 橋梁名 | ・ 路線名 | ・ 有効幅員 |
| ・ 管理事務所 | ・ 橋長 | ・ 形式 |

表- 4.2 表示項目 その2

- | | | | | |
|--------|-------|--------|--------|--------|
| ・ 橋面舗装 | ・ 床組構 | ・ 伸縮継手 | ・ 洗掘変動 | ・ 安定材質 |
| ・ 地覆高欄 | ・ 主構 | ・ 排水装置 | ・ 躯体変動 | ・ 耐震性 |
| ・ 床版 | ・ 支承 | ・ 塗装 | ・ 安定構造 | ・ 全体 |

表-4.1のデータ項目のうち、[形式]を除くデータ項目は先に述べたDBMSの「橋梁データ」テーブルに入力されている。また、表-4.2のデータ項目と[形式]は、DBMSの「橋梁点検台帳」テーブルに入力されている。このように、橋梁補修計画に必要なデータ項目はDBMSの別々のデータベーステーブルに入力されている。

クエリーの作成 C++Builderにおいて1つのフォームにデータベース内の「橋梁データ」、「橋梁点検台帳」の2つのテーブルをアクセスすることも可能であるが、その場合、

検索システムの構造が煩雑になる。そこで、データベース内に「橋梁データ」、「橋梁点検台帳」の表-4.1, 表-4.2に該当するデータ群を1つのクエリー(1つまたは複数のテーブルからデータを取り出したもの)にまとめた。クエリーの名前は「橋梁維持管理データ」とした。

なお、[橋面積塀(m²)], [有効幅員(m)]の値が記入されていないデータは、GASに用いることができないのでクエリー作成時に除いた。この結果、「橋梁維持管理データ」に記入されている橋梁は、171橋である。

表-4.3に「橋梁維持管理データ」を示す。データ型、フィールドサイズについては、表-4.4, 表-4.5に示す。

表示の実装 「橋梁維持管理データ」をC++Builderのフォームに表示するためには、開発環境で述べたC++BuilderのBDEの機能を使い予め用意されたビジュアルコンポーネントを用いる。データベースのデータにアクセスする機能を持ったコンポーネント(テーブルコンポーネント)と、データを表示する機能を持ったコンポーネントとの橋渡しをするコンポーネント(データソースコンポーネント)のプロパティを設定し、データを表示するコンポーネントをフォームに配置する。

「橋梁維持管理データ」を配置したフォームを検索結果表示フォームとする。

検索システムの機能設定

検索システムはデータベースに格納されているデータを検索して該当するデータを表示し、保存する機能を持つシステムである。

検索形式 GASにおいて、全208橋のうち対象となった橋梁数は、表-4.3のデータ項目の[主構], [洗掘変動], [躯体変動], [安定構造]の4項目の値が1(損傷がない)の43橋である。これは、上記4項目の損傷が大きい橋梁は架け替えが必要になり、GASの対象外となるためである。既存のGASでは、扱っている点検データはこの43橋の1パターンだけである。点検データの例として、表-4.6に対象となった43橋のデータを示す。

そこで、本研究においては全171橋のうち補修計画作成のために様々なデータを選出するために、表-4.3のデータ項目のうち[橋面舗装]以下14項目の値を用いて検索する。

表-4.3を見てもわかるように、データベースにおいて[橋面舗装]以下14項目には、点検結果である1, 2, 3の値が記入されている。点検結果がないものには0が記入されている。

検索の仕様を以下に示す。

- 1つの項目の値を1つ指定した場合、その項目に指定した値を持つ橋梁を選び出す。

表- 4.3 橋梁維持管理データ

表示項目	フィールド名	データ型	フィールドサイズ
ID 番号	ID	数値	整数
橋梁名	BRI_NAME	テキスト	16
管理事務所	KANRISYA	テキスト	22
道路種別	DOURO_S	テキスト	20
路線名	ROSEN	テキスト	20
橋長 (m)	KYOUTYOU	数値	単精度浮動小数点
橋面積 (m ²)	HASIMENSEKI	数値	単精度浮動小数点
有効幅員 (m)	TOTAL	数値	単精度浮動小数点
形式	KEISIKI2	テキスト	
全体	AA	数値	バイト
橋面舗装	A5	数値	バイト
地覆高欄	A4	数値	バイト
床版	A3	数値	バイト
床組構	A2	数値	バイト
主構	A1	数値	バイト
支承	A6	数値	バイト
伸縮継手	A7	数値	バイト
排水装置	A8	数値	バイト
塗装	A11	数値	バイト
洗掘変動	A10	数値	バイト
躯体変動	A12	数値	バイト
安定構造	A14	数値	バイト
安定材質	A13	数値	バイト
耐震性	A15	数値	バイト

表- 4.4 Access Ver7.0 におけるデータ型プロパティの設定値

設定値	データ型	サイズ
テキスト型	テキスト, またはテキストと数字の組み合わせが格納される. 電話番号などのように計算を必要としない数字も格納することが可能. (既定値)	255 バイト, または "Field-Size/フィールドサイズ" プロパティで設定される長さの, いずれか小さい方のサイズ以内. テキストフィールドの未使用部分の領域は確保されない.
メモ型	長いテキスト, またはテキストと数字の組み合わせが格納される.	64,000 バイト以内
数値型	数学的演算で使用する数値データが格納される.	1, 2, 4, または 8 バイト (レプリケーション ID に限って 16 バイト)
日付/時刻型	日付および時刻の値が格納される. 年数として指定できる値の範囲は 100~9999.	8 バイト
オートナンバー型	テーブルに新しくレコードが追加されるたびに割り当てられる固有のシーケンシャル (1 単位のインクリメント) 番号, または乱数が格納される. オートナンバー型フィールドを更新することはできない.	4 バイト (レプリケーション ID のみ 16 バイト)
Yes/No 型	Yes/No の値, および 2 種類の値のうちのどちらか 1 つの値 (True/False, On/Off) が格納される.	1 ビット
OLE オブジェクト型	Access テーブルで埋め込みまたはリンクされているオブジェクト (Excel ワークシート, Word 文書, グラフィックス, 音声, その他のバイナリデータ) が格納される.	1GB 以内 (利用できるディスク容量に依存する)

表- 4.5 数値型が設定されている場合のプロパティの設定値

設定値	内容	小数の精度	記憶領域サイズ
バイト型	0～255 の範囲の数値が設定できる。ただし、小数は設定できない。	なし	1 バイト
整数型	-32,768～32,767 の範囲の数値が設定できる。ただし、小数は設定できない。	なし	2 バイト
長整数型	-2,147,483,648～2,147,483,647 の範囲の数値が設定できる。ただし、小数は設定できない。(既定値)	なし	4 バイト
単精度浮動小数点型	-3.402823E38～3.402823E38 の範囲の数値が設定できる。	7	4 バイト
倍精度浮動小数点型	-1.79769313486232E308～1.79769313486232E308 の範囲の数値が設定できる。	15	8 バイト
レプリケーション ID 型	GUID が設定できる。	N/A	16 バイト

- 1つの項目の値を2つ指定した場合、その項目に指定した2つの値のどちらかを持っている橋梁を選び出す。
- 1つの項目を全て選出したい場合は、検索条件を指定しない。
- 複数の項目の値をそれぞれ1つ指定した場合、指定した項目に指定した値を持ち、かつ、他に指定した項目に指定した値を持つ橋梁だけを選び出す。
- 複数の項目の値をそれぞれ2つ指定した場合、指定した項目に指定した2つの値のどちらかを持っており、かつ、他に指定した項目に指定した2つの値を持つ橋梁だけを選び出す。

例えば、「主構」が[1]、「安定構造」が[1or2]の場合、該当する橋梁は「主構」が1かつ、「安定構造」が1または2の値を持つものである。

保存形式 検索システムの検索結果表示フォームにおいて、表示されているデータをファイルに保存するときの出力形式は、既存の GAS で使われていた点検データの入力ファイルと同形式で出力する。これは、これまで用いられてきたシステムを有効に利用するためである。加えて、これまでは対象橋梁数が43橋と決まっていたが、検索システムでは該

表-4.6 対象データ

橋梁名	路線名	橋面舗装	地覆高欄	床版	床組構	主構	支承	伸縮継手	排水装置	塗装	洗掘変動	躯体変動	安定構造	安定材質	耐震性	損傷評価	面積	橋長	支間
ST01	主 Route01	1	1	3	2	1	1	2	1	1	1	1	1	1	1	2	1289	219	5
ST02	主 Route02	1	2	3	1	1	2	2	3	3	1	1	1	1	2	2	117	22	3
ST03	一 Route03	3	2	1	1	1	1	2	1	1	1	1	1	2	2	2	368	67	1
ST04	一 Route04	2	2	2	1	1	1	2	1	1	1	1	1	1	2	2	576	133	4
ST05	国 Route05	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	329	47	1
ST06	主 Route06	1	1	2	1	1	1	2	1	1	1	1	1	1	2	2	313	57	3
ST07	主 Route07	1	1	2	1	1	1	1	1	1	1	1	1	2	1	1	191	91	3
ST08	国 Route05	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	141	20	1
ST09	主 Route08	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	92	22	1
ST10	主 Route09	1	1	2	1	1	1	1	1	2	1	1	1	1	1	3	4270	305	7
ST11	主 Route10	2	1	1	1	1	2	1	1	1	1	1	1	1	1	3	165	30	1
ST12	一 Route11	2	1	2	2	1	2	2	1	2	1	1	1	1	2	2	57	12	1
ST13	主 Route12	2	2	1	1	1	1	2	1	2	1	1	1	1	1	3	93	17	1
ST14	一 Route13	3	2	2	1	1	1	2	2	2	1	1	1	1	2	3	182	41	1
ST15	一 Route14	2	1	2	1	1	1	2	1	1	1	1	1	1	1	3	105	17	1
ST16	一 Route15	3	2	1	1	1	1	1	1	1	1	1	1	1	1	3	120	20	1
ST17	一 Route16	2	1	1	1	1	2	2	1	1	1	1	1	1	2	3	45	6	1
RC18	主 Route17	2	1	1	1	1	1	2	2	0	1	1	1	1	1	3	163	27	3
RC19	国 Route18	1	1	1	1	1	3	1	1	0	1	1	1	1	3	1	340	34	2
RC20	一 Route19	2	2	1	1	1	1	3	3	0	1	1	1	1	1	3	126	16	3
RC21	国 Route20	1	1	1	1	1	1	2	1	0	1	1	1	1	1	3	40	6	1
RC22	主 Route01	1	1	2	1	1	1	1	1	0	1	1	1	2	2	2	3107	173	9
RC23	一 Route21	1	1	1	2	1	1	1	1	0	1	1	1	1	1	2	82	13	1
RC24	主 Route22	1	3	1	1	1	1	2	1	0	1	1	1	1	1	3	103	27	2
RC25	主 Route23	2	2	1	1	1	1	2	1	0	1	1	1	1	1	3	119	21	2
RC26	国 Route24	2	2	1	1	1	2	2	1	0	1	1	1	1	2	2	86	19	2
RC27	主 Route25	3	1	1	2	1	1	1	1	0	1	1	1	1	1	2	102	17	1
RC28	国 Route05	2	1	1	2	1	2	1	1	0	1	1	1	1	2	2	170	17	2
RC29	主 Route26	1	1	1	1	1	2	1	1	0	1	1	1	1	2	2	68	15	1
RC30	国 Route05	1	1	1	1	1	2	1	1	0	1	1	1	2	2	2	234	39	2
RC31	主 Route27	1	2	1	1	1	1	1	1	0	1	1	1	1	2	2	273	50	2
RC32	主 Route28	1	1	1	1	1	1	1	1	0	1	1	1	1	1	3	120	20	1
RC33	国 Route05	1	2	1	1	1	1	2	1	0	1	1	1	1	1	3	466	67	2
RC34	主 Route25	2	1	1	2	1	1	2	1	0	1	1	1	1	2	2	70	14	1
RC35	一 Route29	1	1	1	1	1	2	3	1	0	1	1	1	1	2	3	90	12	1
RC36	一 Route30	2	2	1	1	1	2	3	1	0	1	1	1	1	2	3	79	5	1
RC37	主 Route31	3	3	1	1	1	2	3	1	0	1	1	1	2	2	3	45	10	1
RC38	一 Route32	1	1	1	2	1	1	2	1	0	1	1	1	1	2	3	64	15	1
RC39	一 Route33	2	1	1	2	1	2	3	1	0	1	1	1	1	2	3	45	10	1
RC30	国 Route05	1	1	1	2	1	2	2	2	0	1	1	1	1	1	3	85	12	1
RC41	主 Route34	1	1	1	1	1	2	3	1	0	1	1	1	1	2	2	98	15	1
RC42	国 Route05	2	1	1	1	1	1	2	1	0	1	1	1	1	1	3	118	20	1
RC43	一 Route35	3	1	1	1	1	1	1	1	0	1	1	1	1	1	3	56	8	1

面積の単位は m², 橋長は m
 国：国道, 一：一般県道, 主：主要地方道

当する橋梁数が検索のたびに變更されるため、検索の結果該当した橋梁数を既存の GAS の読み込み用にファイルに出力する。

[検索] ボタンの実装 本システムでは、検索システムの [検索] ボタンをクリックすると、検索条件入力ダイアログボックスがモーダルモードで表示される。この場合、ダイアログボックスの入力が完了するまでは、呼び出し側である検索結果表示フォームには制御が戻らない。ダイアログボックスで検索条件を入力し、[OK] ボタンまたは [キャンセル] ボタンをクリックされると、呼び出し元のフォームに制御が戻る。[OK] ボタンをクリックされた場合のみ検索処理を実行する。以下に検索処理の概要を示す。

1. 検索用のデータの取得

- 入力フィールドをチェックし、データが入力されている場合は、データを作業用バッファにコピーする。
- この時に、データをコピーするとともに、どのデータの入力があったかを判定するために、一時領域にデータチェック用のビットを立てる。
- それぞれの入力フィールドをチェックした後で、ビットをチェックし、検索条件が入力されていない場合はメッセージを表示してイベント関数を抜ける。
- 検索データがある場合は、テーブルをオープンする。
- テーブルのオープンに失敗した場合は、メッセージを表示しイベント関数を抜ける。

2. 抽出条件の設定

抽出条件は入力されたデータの項目とその値に該当するデータを Filter プロパティにセットするようにする、

入力フィールドは、誤入力を防ぐためプルダウン方式を採用する。プルダウンリストを図-4.9 に示す。

3. 抽出実行

各項目の抽出条件を結び Filter プロパティにセットすることにより、データを抽出する。該当データが無い場合は、メッセージを表示しイベント関数を抜ける。

[保存] ボタンの実装 テーブルがオープンされている間(データが表示されている間)、[保存] ボタンは有効である。[保存] ボタンをクリックすると、ファイル名を付けて保存ダイアログが表示される。ファイルを選択、または新規作成しダイアログの [保存] ボタンをクリックすると、保存処理が行われる。[ファイル名を付けて保存] ダイアログにおいて、ファイル名とファイルの種類のデフォルトを示す。

- ファイルの種類：入力ファイル (*.dat)
 - GAS で用いる入力データであることを示すため。
- ファイル名：Tenken.dat
 - GAS で用いる入力データの1つである点検データであることを示すため。

保存処理が実行されると、表示されている該当データのレコードを最初に戻し、表示されているデータを順に出力していく。そして、最後のレコードまで出力すると、呼び出し側の検索結果表示フォームに制御が戻る。

プロジェクトの統合

第1フェーズで構築した検索システムと GAS で使われているアプリケーションの統合では、検索システムで出力したデータのファイルを GAS の入力データとして使用できることを目的とする。

検索システムで出力したファイルと GAS で使用されていた入力ファイルの違いは、検索条件に該当した橋梁数が出力されているかどうかである。既存の GAS では、橋梁数が43橋と決められていたので、検索システムで保存したファイルでは既存の GAS の計算に用いることができない。そこで、GAS でのアプリケーションのプログラムの変更が必要となる。

以下に、プログラムの変更内容を示す。

- GAS のパラメータの設定フォームにあった橋梁数の項目欄を削除する。削除後のフォームは図-4.12 のようになる。
- ファイル設定フォームの [入力設定] ページの点検データを選択したときに、そのファイルに出力されている該当橋梁数を読み込む。この変更により、従来パラメータの設定の際に読み込んでいた橋梁数をファイルが選択されたときに読み込むようにすることで、既存システムの変更をできるだけ少なくした。

4.2 橋梁補修計画作成システム

4.2.1 処理の流れ

本システムは開発環境のプラットフォームとしてマイクロソフト社の Windows 95/98 および Windows NT4.0 を選択することによって、システムの操作をマウス中心に行うよう設計されている。

本システムは、パーソナルコンピュータ上で本実行ファイルを開くことにより、システムが起動し、メインフォームが立ち上がる。

図-4.6 にメインフォームを示し、それぞれのメニューの概略を以下に述べる。

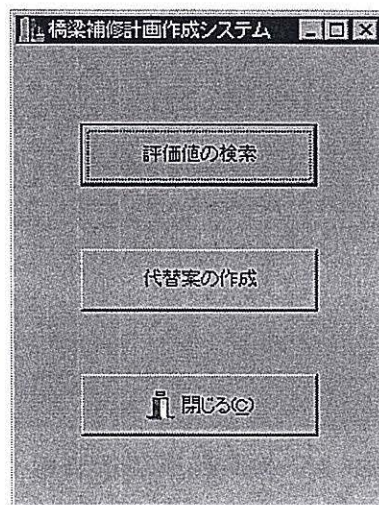


図- 4.6 メインフォーム

1. 評価値の検索：データの検索，保存を行う場合に用いる。
2. 代替案の作成：補修計画を作成する場合に用いる。
3. 閉 じ る：全システムを終了する。

検索

メインフォーム (図-4.6) の [評価値の検索] ボタンをクリックすることにより，検索結果表示フォームが表示される。図-4.7 は，検索結果表示フォームの初期表示画面である。

メニューボタンは4つある。そのうちの [保存] ボタンと [ナビゲーター] ボタンは初期表示画面において，無効である。また，データ項目名の横にある空白フィールドは表示専用であるため，入力することはできない。

次に， [終了] ボタンをクリックすると検索結果表示フォームは閉じられる。
 [検索]， [保存]， [ナビゲーター] ボタンについては， 後で説明する。

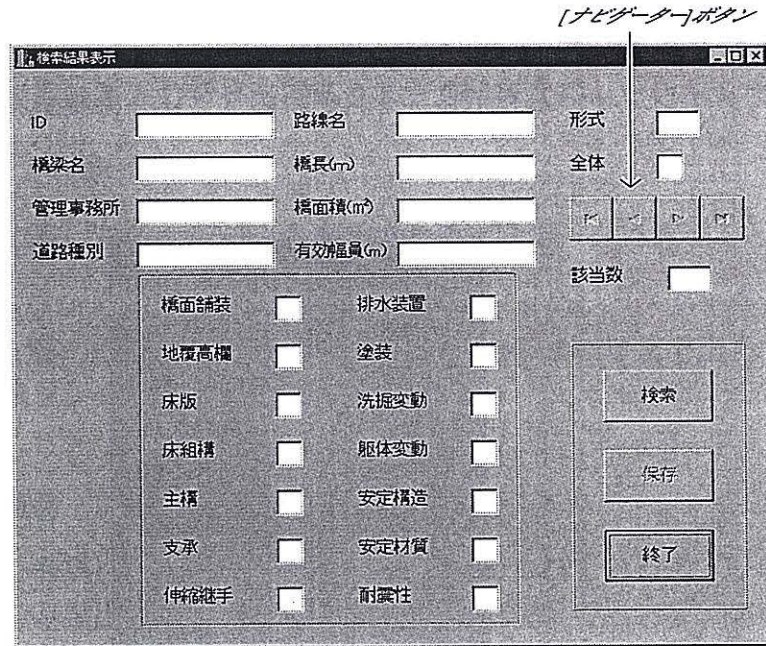


図- 4.7 検索結果表示フォーム (初期表示)

検索条件入力 図-4.7のフォームで[検索]ボタンをクリックすると検索条件入力ダイアログが表示される。図-4.8は、検索条件入力フォームの初期表示画面である。このダイアログに検索条件を入力し[OK]ボタンをクリックすることにより、検索処理が実行される。また、[キャンセル]ボタンをクリックすると検索結果表示フォームに戻る。

検索条件の入力はプルダウン方式である。また、未入力の場合は、入力を要求するエラーメッセージが表示される。

図-4.9にプルダウンリストとエラーメッセージを示す。

保存

検索結果表示フォームにデータが表示されている状態において、[保存]ボタンをクリックすると保存処理を実行することができる。

検索結果表示フォームの[保存]ボタンをクリックすると[ファイル名を付けて保存]ダイアログが表示される。ダイアログボックスのファイルを選択、または新規作成してダイアログの[保存]ボタンをクリックすると、選択または作成したファイルにデータが出力され



図- 4.8 検索条件入力フォーム (初期表示)



図- 4.9 プルダウンリストとエラーメッセージ

る。[キャンセル] ボタンをクリックした場合、検索結果表示フォームに戻る。
図-4.10 に [ファイル名を付けて保存] ダイアログを示す。



図- 4.10 [ファイル名を付けて保存] ダイアログ

GA による解探索

メインフォーム (図-4.6) の [代替案の作成] ボタンをクリックすると代替案の作成フォームが表示される。図-4.11 に代替案の作成フォームを示し、それぞれのメニューボタンをクリックしたときの動作を述べる。

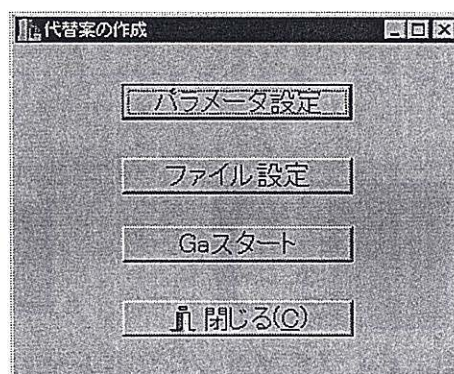


図- 4.11 代替案の作成フォーム

1. パラメータの設定：パラメータ設定フォームが表示される。
2. ファイルの設定：ファイル設定フォームが表示される。

3. Ga スタート

- パラメータとファイルの設定が完了している場合、計算状況を示すフォームが表示され、計算が開始される。
- パラメータの設定およびファイルの設定が完了していない場合、設定を要求するエラーメッセージが表示される。
- エラーメッセージは図-4.9 のエラーメッセージと同形式である。

4. 閉じる：代替案の作成フォームを閉じる。

パラメータの設定 図-4.12, 図-4.13, 図-4.14 にパラメータ設定フォームの初期表示画面を示す。このフォームにおいて各項目の設定を行い、[OK] ボタンをクリックすることで設定を完了する。

パラメータ設定

パラメータ No.1 | パラメータ No.2 | パラメータ No.3

最大世代数 5000 出力を行う世代間隔 5000

人口数 100 集団数 1

予算:千円 50000

重要度係数 ニューラルネットワーク

目的関数 3.評価値の増加量 - α × 残金

OK キャンセル 閉じる

重要度係数

ニューラルネットワーク

数値化理論第II類

ファジー理論(A積分)

ファジー理論(maxmin法)

ニューラルネットワーク

目的関数

3.評価値の増加量 - α × 残金

1.費用 × 評価値の増加量

2.評価値の増加量

3.評価値の増加量 - α × 残金

4.評価値の増加量 + β × (同一橋梁内の補修部位数) - α × 残金

5.評価値の増加量 + γ × (橋梁数 - 路線数) - α × 残金

図- 4.12 パラメータ No.1

パラメータ設定

パラメータ No.1 | **パラメータ No.2** | パラメータ No.3

エリート個体数 突然変異の確率

スケーリング

べき乗スケーリング

係数

世代数

線形スケーリング

係数

係数/2の世代数

OK キャンセル 閉じる

図- 4.13 パラメータ No.2

パラメータ設定

パラメータ No.1 | **パラメータ No.2** | パラメータ No.3

交叉

確率 交叉点数

移住

個体の割合 世代間隔

目的関数の係数

ALPHA BETA GAMMA

OK キャンセル 閉じる

図- 4.14 パラメータ No.3

ファイルの設定 図-4.15, 図-4.16 にファイル設定フォームを示す.

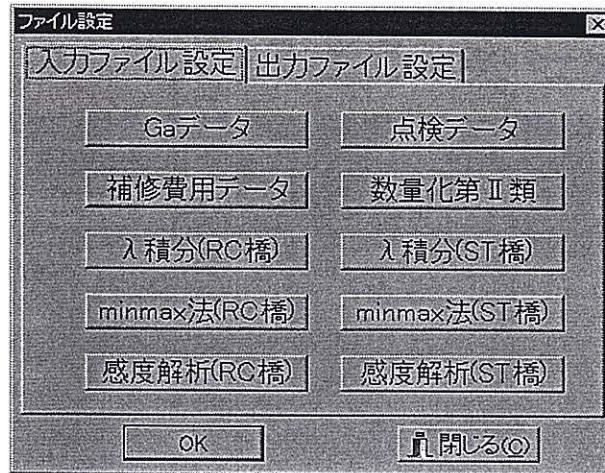


図- 4.15 入力ファイル設定

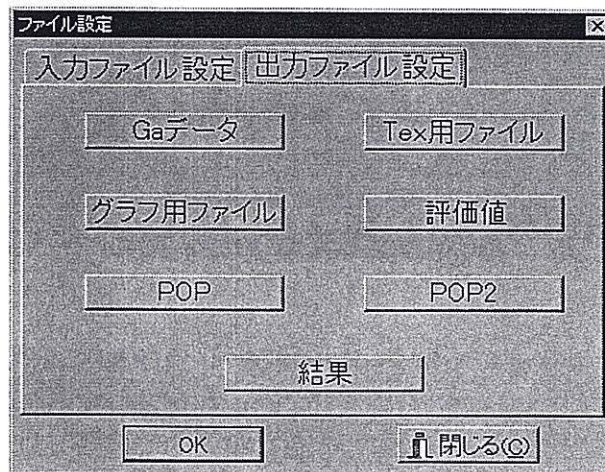


図- 4.16 出力ファイル設定

ファイル設定フォームの各メニューボタンをクリックすると、図-4.10 に示したダイアログと同形式のダイアログが表示される。ダイアログにおいて、対象となるファイルを選択または作成し、全ファイルの設定を終えた後に [OK] ボタンをクリックすることによりファイルの設定を完了する。

4.2.2 実行処理例

実際に、データの検索から代替案作成までの実行例を示す。

検索条件とパラメータの設定

検索条件は図-4.17 に示す通りである。

項目	設定値	項目	設定値
橋面舗装	lor2	排水装置	
地覆高樹	lor3	塗装	
床版		洗掘変動	1
床組構	1	躯体変動	1
主橋	1	安定構造	1
支承		安定材質	
伸縮継手		耐震性	

図- 4.17 検索条件入力例

パラメータの設定は図-4.12, 図-4.13, 図-4.14 に示す値で行った。

検索結果

検索結果は図-4.18 に示す通りであり、該当橋梁数は40 橋である。

[ナビゲーター]ボタン

検索結果表示

ID: 6090 路線名: 志賀富来線 形式: ST

橋梁名: 川尻橋 橋長(m): 91 全体: B

管理事務所: 羽咋土木事務所 橋面積(m²): 774

道路種別: 主要地方道 有効幅員(m): 8

該当数: 40

橋面舗装	1	排水装置	1
地覆高欄	1	塗装	1
床版	2	洗掘変動	1
床組構	1	躯体変動	1
主構	1	安定構造	1
支承	1	安定材質	2
伸縮継手	1	耐震性	1

検索
保存
終了

図- 4.18 検索結果表示例

図-4.18 の右上の楕円に囲まれた [ナビゲーター] ボタンを左から (1), (2), (3), (4) とし、クリックしたときの所定の動作を示す。

- (1) 該当レコードを先頭に移動する。
- (2) 該当レコードを1つ前に移動する。
- (3) 該当レコードを1つ後に移動する。
- (4) 該当レコードを末尾に移動する。

データの出力

図-4.19 に、図-4.18 のデータを出力したファイルを示す。保存したファイル名を Tenken 実行例.dat とする。

行	14	40	型	面積	橋長	橋名	種別	路線名↓
1	1	1	3	211.2	26.4	城谷橋	国	167号↓
2	1	1	3	239.4	39.2	杖川橋	国	小松越鶴来線↓
3	1	1	2	329.0	47.0	於古川橋	国	249号↓
4	1	1	2	104.0	16.0	深谷橋	国	249号↓
5	1	1	2	774.0	91.0	川尻橋	国	志賀富来線↓
6	1	1	1	556.0	40.0	勘済橋	国	304号↓
7	1	1	2	130.0	20.0	岡野大橋	国	160号↓
8	1	1	2	123.5	19.0	丹尾橋	国	249号↓
9	1	1	3	33.0	10.3	藤野橋	一	城山線↓
10	1	1	1	263.5	33.8	中島橋	一	長浦中島線↓
11	1	1	1	104.0	16.0	倉部大橋	主	金沢小松線↓
12	1	1	2	3980.0	398.0	美川大橋	主	金沢小松線↓
13	1	1	1	120.0	20.0	安原川橋	主	松任宇ノ氣線↓
14	1	1	3	537.2	63.7	若松橋	主	金沢井波線↓
15	1	1	2	58.3	6.2	的場橋	一	安吉松任線↓
16	1	1	3	61.8	9.5	新戸橋	一	額谷三浦線↓
17	1	1	2	5913.9	422.4	乙丸跨線橋	一	八田金沢線↓
18	1	1	3	1748.6	178.1	津幡跨線橋	一	中尾津幡線↓
19	1	1	1	98.7	14.1	上矢田橋	一	爪生能瀬線↓
20	1	1	3	114.0	19.0	盤若橋	国	249号↓
21	1	2	2	271.1	41.7	白湯橋	国	249号↓
22	1	1	3	141.4	20.2	万寿跨線橋	国	249号↓
23	1	1	2	234.0	39.0	本田橋	国	249号↓
24	1	1	1	97.5	15.0	平野橋	主	穴水前線↓
25	1	1	2	67.5	15.0	甲大橋	主	能都穴水線↓
26	1	1	1	93.3	21.7	鹿波大橋	主	能都穴水線↓
27	2	1	2	156.8	19.0	神主田橋	主	宇出津野線↓
28	1	2	2	313.5	57.0	小間生橋	主	宇出津野線↓
29	1	3	1	102.5	20.5	年化橋	主	珠洲穴水線↓
30	1	2	2	276.0	46.0	明治橋	主	珠洲里線↓
31	1	1	2	99.0	22.0	渡詰橋	国	416号↓
32	1	1	3	252.6	42.1	江指大橋	国	416号↓
33	1	2	2	3105.0	172.5	小松中央立交差橋	主	小松越鶴来線↓
34	1	2	2	228.2	32.6	絵馬堂橋	主	小松山中線↓
35	1	2	3	2897.5	305.0	城南橋	主	金沢小松線↓
36	1	1	3	34.8	6.1	栢野橋	主	364号↓
37	2	1	3	244.8	27.2	中央大橋	主	橋立港線↓
38	1	1	2	92.4	22.0	栢野大橋	一	山中伊切線↓
39	1	1	3	53.0	6.8	神明橋	一	動橋山代線↓
40	2	2	3	105.4	17.0	柴山湯3号橋	一	新保矢田野線↓

図- 4.19 Tenken 実行例.dat

代替案の作成

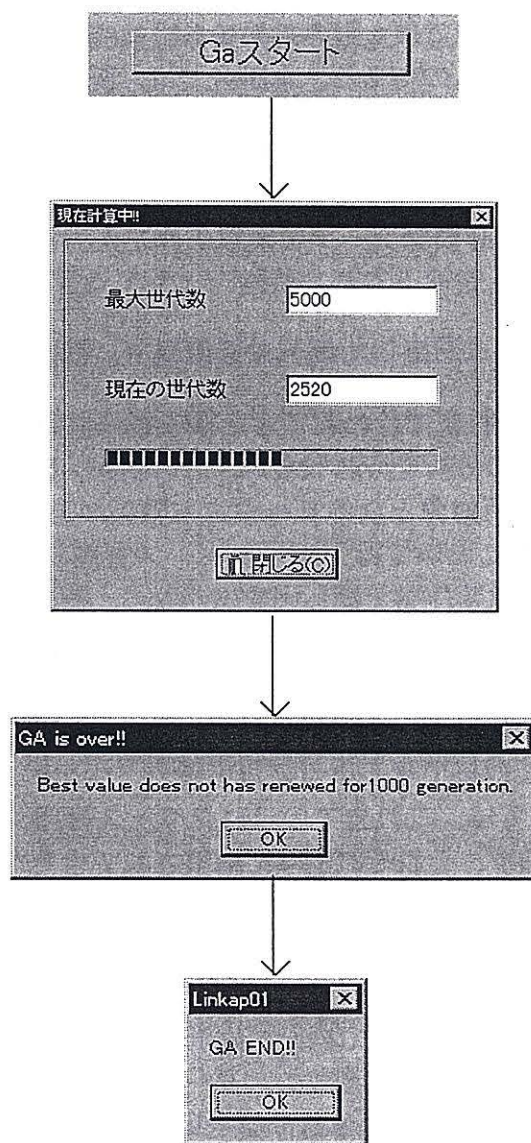


図- 4.20 計算過程

この実行例では、収束条件の「集団中の最大適応度が1000世代更新されなかったとき」にあてはまり、計算が5000世代に到達する前に終了された。

4.3 結論

橋梁補修計画作成システムは、パーソナルコンピュータ上において、実際の橋梁維持管理に用いられるデータが格納されたデータベースから補修・補強の対象となる橋梁群を選出し、限られた予算の中で最大限の効果をあげる橋梁補修計画代替案の作成を一貫して行うことができるシステムのである。本研究では、既存システムを見直し、そのシステムを最大限に利用して新しいシステムの構築を行った。

既存システムに新しく加えた機能を以下に列挙する。

- (1) 補修計画作成に必要なデータを検索条件を入力することによって、データベースから取り出す。
- (2) データベースから選出したデータを用いて、補修案の作成までを一貫して行う。

(1)に関しては、これまで行われてきた橋梁補修計画作成システムの研究では、様々な手法を用いてデータの分析が行われたが、その結果、課題として残された問題は分析手法そのものよりも、分析対象となるデータにも原因があると考えられるというものである。これまでのシステムにおいては、データベース管理システムのデータを用いていたものの、それらのデータは手作業で探し出すほかなく、その関係から橋梁補修計画作成システムでは、適切と思われる1つのデータ群をもとに分析が行われた。その結果、分析手法は改良を重ねられてきたが、データに関する問題はそのままにされてきた。そこで、この機能の追加により様々なパターンのデータ群を選出することができ、データの更新や新設橋梁の増加にも素早く対応できるようになった。

また、(2)に関しては、これまでのデータベース管理システムと橋梁補修計画作成システムは連帯しておらず、それぞれ別々に構築されてきたため、補修案の作成において読み取りように構成された1つのデータ群しか扱うことができなかった。しかし、この機能の追加により、様々な橋梁数を持ったデータ群を扱うことができ、かつ、データの検索から補修案の作成までを迅速に行うことができるようになった。

(1)、(2)の機能の追加によって橋梁補修計画作成システムは、実際に使われているデータの更新、増加にも対応でき、多数のパターンの対象橋梁群に対しての補修案を作成できるようになった。また、検索操作は簡略で理解しやすく、より実用的なものとなった。

今後の課題を述べると、本システムの検索システムでは橋梁診断の各項目の評価値を条件項目とし検索しているが、実際にシステムを扱う上で必要となってくる条件は、評価値の他に管理事務所別や橋種別などの条件である。さらに検索を行う際に、専門知識があれば、必要な検索条件を入力し検索することができるが、専門知識がない場合、どのように検索条件を設定し対象となる橋梁を検索するのかと言う問題が生じる。また、本システムは、元々2つのシステムからなっているため、今後それぞれ2つの方向で発展して行くものと考えられる。データベース管理システムに関しては、コンピュータ技術の進歩により

機能性, 信頼性, 簡便性の向上, さらにネットワークシステムなどによる発展性が考えられる。さらに, 本システムで利用している補修案作成システムは, 単年度の補修代替案の作成を目的としたものであるが, 実際問題としては, 5ヶ年, 10ヶ年のスパンでの補修計画が必要となってくる。そこで, 今後は適切なスパンを用いた補修計画作成システムの策定が望まれる。よって, これらの課題を踏まえてシステムを発展させて行かなければならない。

参考文献

- 1) 谷口 健:「橋梁データ管理システムの構築に関する研究」, 金沢大学卒業論文, 1997.
- 2) 西 雄一:「点検データに基づく橋梁補修計画への GA の援用に関する研究」, 金沢大学卒業論文, 1997.
- 3) 石川県:「昭和 57 年度～昭和 63 年度 橋梁診断調査総括表集」, 1990.
- 4) 橋 謙二:「遺伝的アルゴリズムを援用した橋梁維持管理支援システムに関する研究」, 金沢大学修士論文, 1994.
- 5) 木下 真二:「点検データに基づく GA を援用した橋梁補修計画に関する研究」, 金沢大学修士論文, 1995.
- 6) 福山 貴久:「GA を援用した橋梁補修計画に関する研究」, 金沢大学修士論文, 1997.
- 7) 辻岡 信也:「NN による橋梁損傷度判定システムの精度向上に関する研究」, 金沢大学修士論文, 1998.
- 8) 今野 浩, 鈴木 久敏:「整数計画法と組み合わせ最適化」, 日科技連, p. 149, 1982.
- 9) 藤田 喜久雄:「遺伝的アルゴリズムによる最適化問題へのアプローチ」, 日本機会学会関西支部設計 AI 研究懇話会第 4 回資料, pp.13, 1993.
- 10) 西川 緯一:「遺伝的アルゴリズムとその工学的意味」, システム/制御/情報, Vol.37, No.8, pp.447, 1993.
- 11) 北野宏明:「遺伝的アルゴリズム」, 産業図書, p.13, 1993.
- 12) 安居院猛, 長尾智晴:「ジェネティックアルゴリズム」, 昭晃堂, pp.10～11, 1993.
- 13) 前出), p.15

- 14) 近田康夫, 橋 謙二, 城戸隆良, 小堀為雄:「GAによる既存橋梁の補修計画支援の試み」, 土木学会論文集, No.513/I-31, pp.151~159, 1994.
- 15) 玉川 竹春 著, 三田 典玄 監: C++Builder3 データベースプログラミング, pp.7~11, プレンティスホール出版, 1998.
- 16) 谷尻 かおり 著, 谷尻 豊寿 監: C++Builder ファースト・プログラミング, pp.317~324, 技術評論社, 1997.

第5章 橋梁点検データの見直し

5.1 ニューラルネットワーク

5.1.1 ニューラルネットワークの概要

ニューラルネットワークの概念

ニューラルネットワーク¹とは人間の脳の神経回路網をモデルにしたコンピュータシステムのことである。人間に近い能力をもつ情報処理システムの構築を目的としている。

今日のコンピュータは、そのほとんど全てがノイマン型コンピュータの範疇にはいる。プログラムとして逐次的な処理手順を前もって与えておくことは、ノイマン型コンピュータの大きな特徴の一つである。ノイマン型コンピュータは、値計算やデータベース探索などのことを正確で曖昧性なく、かつ高速に処理できる。しかし、曖昧性を多く含むような情報の認識(パターン認識)等を行うことはなかなか困難である。逆に人間は数値計算の高速処理等は苦手であるが、パターン認識の能力には優れている。たとえば人間は、どんな表情をしていてもその人物を認識できる。ノイマン型コンピュータでパターン認識を行う場合には、可能なあらゆる解答を記憶しておき、与えられた入力データに対する最適な解を選びださねばならない。ただし、入力情報がほんの少し違うだけで全く違うものとして扱われる、意味のはっきりしない情報は扱いにくい、かつ入力情報の関係を示すための情報がさらに必要となる、といった問題点がある。また、入力データはランダムに変形したものが許される。これらのことから、入力データに対する解答の検索には相当の時間を要するだけでなく、解答そのものが記憶されていない可能性も考えられる。それに対し人間は、このようなパターン認識の諸問題を楽々と処理している。このことから人間の脳には、ノイマン型コンピュータとは別の処理機構が備わっていると考えられる。つまり並列処理型の機構である。

ノイマン型コンピュータの限界性も認められたため、人間の脳をモデルとした、ランダム性・あいまいさを含む問題に強い情報処理システム、ニューラルネットワークを構築しようとする動きが盛んになったのである。ニューラルネットワークはパターン認識にとど

¹モデルとしている生体のニューラルネットワークと明確に区別する場合には、人工ニューラルネットワーク(ANN:Artificial Neural Network)という。本論文では単にニューラルネットワークという場合にはANNのことを指す。

まらず、推論、学習、言語、記憶等の人間の認知機能に関する問題にも応用されている。

基本構造

生体のニューラルネットワークのモデル化 人間の脳の神経回路網を抽象化すると、多数の神経細胞が興奮性・抑制系のシナプス²結合で密に結合した、並列的な分散情報処理システムとしてモデル化できる。このようなネットワークの接点にあたる神経細胞(ニューロン)は、入力データを別の出力データに変換する関数をもつ処理ユニットと考えられる。またネットワークを結合するシナプス結合の伝達効率は、ネットワークの入出力データや状況によって変化できる可塑性をもっていると考えられる。さらにネットワーク全体の状態は、各ニューロンの状態やシナプス結合の伝達効率のネットワーク全体の数値的な分布パターンとして表現できる。

これらのことは、脳の神経回路網の特徴を基本的にとらえたものと考えられる。これがニューラルネットワークの基本的な考え方である。ニューラルネットワークでは、ニューロンの結合の伝達効率を変化させることで入出力関係を変えることができる。このことは、ニューラルネットワークの学習と呼ばれる。今後、人工的なニューロンのモデルのことを“ユニット”³という言葉で表現する。また、ニューロン間の結合の伝達効率のことを“重み”という言葉で表現する。

ユニットのイメージを図 5.1 に示す。

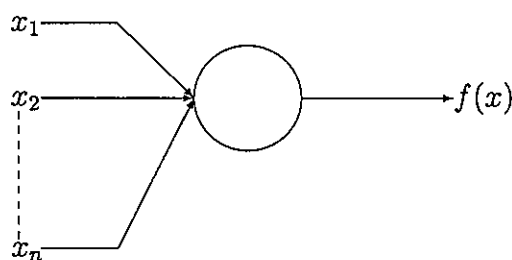


図- 5.1 ユニットのイメージ

つまりニューラルネットワークは、脳と同様に、多数のユニットを広範囲にわたり互いに接続したものである。

ユニット 各ユニットは、多数の他のユニットから入力される荷重和(重み付きの総和)があるしきい値を越えたときにのみ、出力を出すという単純なものである。先にも述べた

²一つのニューロンから他のニューロンや細胞に興奮を伝達するための接合部

³形式ニューロン、ノードといった言い方もある。

が、ユニットは入力データを別の出力データに変換する関数をもっている。その関数の代表的なものとして次の2式がある。

マカロック・ピッツモデル

$$f(x) = \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0) \end{cases} \quad (5.1)$$

シグモイド関数モデル

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (5.2)$$

つまり、あるユニット i ($1 \leq i \leq n$) の出力は次式で示される。

$$x_i(t+1) = f\left(\sum_{j \neq i} w_{ij} x_j(t) - \theta_i\right) \quad (5.3)$$

ここで、 $x_i(t)$ は時刻 t におけるユニット i の出力値である。ユニット j の出力値がユニット i の入力値になっているとする。 w_{ij} はユニット j からユニット i への重み、 θ_i はユニット i のしきい値である。

ネットワークモデル ニューラルネットワークは幾つかの代表的なモデルに分類することができる。まずユニットの接続のしかた、つまりネットワークの形態で分類できる。以下に代表的なものを示す。

- **相互結合型ネットワーク**

相互結合型のネットワークでは、任意の二つのユニットが互いに接続している。入力データは、ネットワーク内を変化しながら何度も伝わる。そして、各ユニットの状態は最終的に一つの平行状態か、あるいは周期的な変動をするようになる。相互結合型ネットワークのモデル図を図 5.2 に示す。

- **階層型ネットワーク**

階層型ネットワークでは、ユニットが層状にグループ化されグループ間では信号は特定の方向にのみ伝わる。またフィードバック⁴を含むか、層内のユニット同士での結合があるか、ということによってさらに分類される。その例を図 5.3, 5.4, 5.5 に示す。

⁴あるユニットの出力が他のユニットを経て再び自信に入力信号として戻ってくるメカニズム

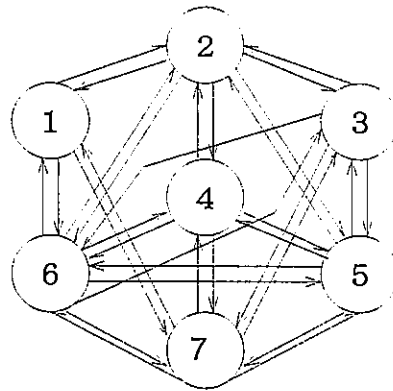


図- 5.2 相互結合型ネットワーク

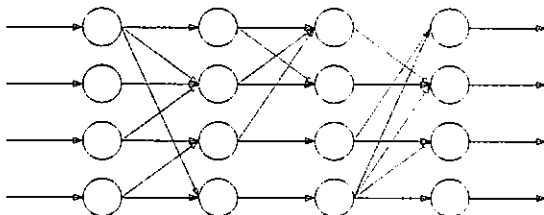


図- 5.3 単純階層型ネットワーク

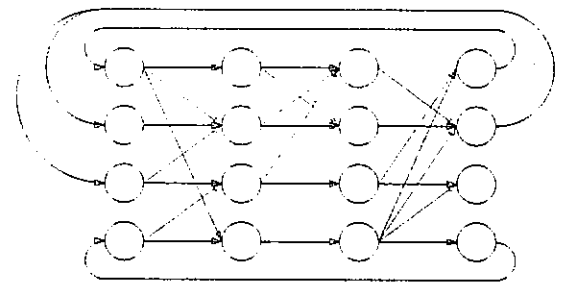


図- 5.4 フィードバックのある階層型ネットワーク

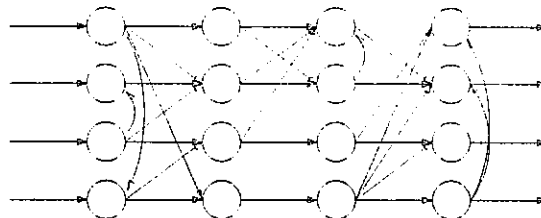


図- 5.5 層内結合がある階層型ネットワーク

ニューラルネットワークの学習とは、情報処理システムの目的にあうようにユニット間の重みを調整することである。学習を行うためには、評価の基準となるものが必要となる。その基準を教師信号と呼ぶが、教師信号を外部から与えるか、内蔵しているかによってニューラルネットワークは次の2つのモデルに分類される。

- **教師あり学習**

入力データとともに、その入力データに対する理想的な出力(教師信号)を与えておき、ニューラルネットワークのシステムを教師信号にあうように修正する方式。

- **教師なし学習**

外部からいちいち教師信号を与えない学習方式のことで、自己組織化ともいわれる。つまり、外界から内部構造に関する特別な信号(教師信号)がないにもかかわらず、その内部構造を外界に適応させることができるのである。

特徴

ニューラルネットワークの特徴として、次の点があげられる。

- 多数のユニットが同時に稼働する、並列処理が行われること。
- いくつかのユニットが不良であってもネットワーク全体の機能には影響されることなく、ネットワークが頑強であること。
- ユニットの各入力に対する重みを、処理目的の最適値へと自動的に調整する学習機能をもっていること。

特に学習機能は重要である。つまりニューラルネットワークを使用する場合、コンピュータに処理すべき問題の定義やプログラムを与える必要はない。なぜなら訓練すべき入力データを十分に与え学習を重ねることで、各ユニットの重みはシステムに最適なように調整されるからである。

5.1.2 SOM(Self-Organizing Map)

文献²⁾³⁾⁴⁾では、ニューラルネットワークにおけるユニット結合荷重(重み)のことをコードブックベクトルという言葉で表現している。SOM/LVQ に関しては文献²⁾³⁾⁴⁾に従った記述をすることにする。

基本構造

SOM(自己組織化マップ)は、教師信号を必要としない教師なし学習方式をとっているニューラルネットワークの一つである。つまり入力データ中の隠れた特徴を抽出し自動的に学習を行う。SOMは入力データの類似度を自動的に検出し、似た入力同士をネットワーク上の近くに配置する。データ間の類似は、マップ層での近さの関係に写される。ただしSOMは、複数データのクラスタ化(集団化)を行っているのであり、統計的なパターン認識を意味してはいない。SOMは入力層とマップ層からなる2層のネットワークである。マップ層は2次元で構成される。各入力ノードは、マップ層の全てのノードと結合しており、また同じ層での結合は無い。その概念を図5.6に示す。

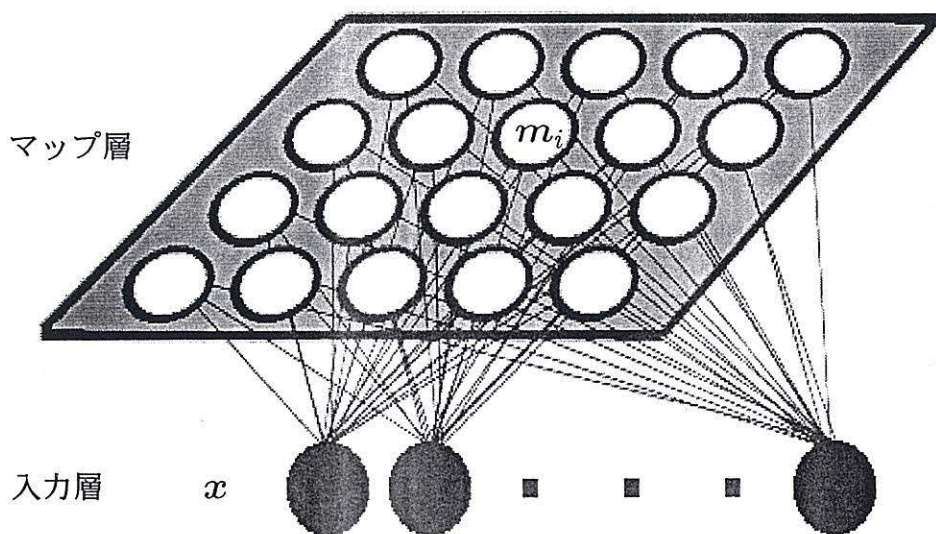


図- 5.6 SOM の概念図

つまりSOMは、入力データ空間 \mathcal{R}^n からノードの2次元配列上へのマップ化を定義している。つまり、多次元データの非線形関係を可視化する。入力データ空間を \mathcal{R}^n とすると、コードブックベクトル $m_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T \in \mathcal{R}^n$ は⁵⁾、すべてのノード i と結び付

⁵⁾ X^T は、行列 X の転置行列

けられている。最も簡単な場合において、入力ベクトル $x = [\xi_1, \xi_2, \dots, \xi_n]^T \in \mathbb{R}^n$ は可変のスカラ重み μ_{ij} を介して並列にすべてのニューロンと結合している。一般にそれらの重みは異なったニューロンに対してはすべて異なった値を持っている。

学習アルゴリズム

ここでは SOM の学習アルゴリズムについて説明する。

1. ネットワークの初期化

入力層とマップ層間のコードブックベクトルの初期値を、乱数を用いて小さな値に設定する。

2. 入力ベクトルの入力

入力層に入力ベクトル

$$x = [\xi_1, \xi_2, \dots, \xi_n]^T \in \mathbb{R}^n \quad (5.4)$$

を入力する。

3. コードブックベクトルと入力ベクトルの距離 (ユークリッド距離) の計算

上の入力ベクトルに対するマップ層のノード i のコードブックベクトルは

$$m_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T \in \mathbb{R}^n \quad (5.5)$$

で示される。

マップ層で各ノードのコードブックベクトルと入力ベクトルとのユークリッド距離を計算する。

$$\|x - m_i\| = \sqrt{\sum_{j=1}^n (\xi_j - \mu_{ij})^2} \quad (5.6)$$

4. ユークリッド距離が最小となるノードの選択

この競合によって選ばれたノードを勝者ノードとし、添え字 c によって表す。

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \quad (5.7)$$

5. コードブックベクトルの更新

勝者ノードとその近くのノードのコードブックベクトルが、次式により更新される。

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \quad (5.8)$$

ここで $t = 0, 1, 2, \dots$ は整数, すなわち離散時間座標である. $h_{ci}(t)$ は近傍関数である. 緩和過程では, 関数 $h_{ci}(t)$ は非常に中心的な役割をする. すなわち, 格子点上に定義される平滑カーネルとして振る舞う. 近傍関数は収束するために, $t \rightarrow \infty$ のとき, $h_{ci}(t) \rightarrow 0$ である必要がある. 普通 $h_{ci}(t) = h(\|r_c - r_i\|, t)$ であり, ここで $r_c \in \mathbb{R}^2$ と $r_i \in \mathbb{R}^2$ はそれぞれの配列の中でノード c と i の位置ベクトルである. $\|r_c - r_i\|$ が増加するにつれて $h \rightarrow 0$ になる. h_{ci} の平均幅と形は, データ点群にどのように“弾性表面”の“かたさ”を定義する.

近傍関数は次の二つの選択が可能である.

バブル関数

$$h_{ci} = \begin{cases} \alpha(t) & i \in N_c \\ 0 & i \notin N_c \end{cases} \quad (5.9)$$

ガウス関数

$$h_{ci} = \alpha(t) \cdot \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right) \quad (5.10)$$

ここで N_c は近傍集合, $\alpha(t)$ は学習率係数 ($0 < \alpha(t) < 1$) である. N_c の半径と $\alpha(t)$ は, 両方とも時間とともに普通単調減少する. $\sigma(t)$ は関数の幅のことであり, N_c の半径に相当する.

また $\alpha(t)$ は次の二つが選択可能である.

線形関数 (linear function)

$$\alpha(t) = \alpha(0) \left(1.0 - \frac{t}{rlen}\right) \quad (5.11)$$

時刻反比例型関数 (inverse-time type function)

$$\alpha(t) = \alpha(0) \frac{C}{C+t} C = \frac{rlen}{100.0} \quad (5.12)$$

$rlen$ は学習回数であり, $\alpha(0)$ は学習率係数の初期値である. 両者とも, それぞれパラメータ $rlen$, $alpha$ として設定する. $\alpha(t)$ は最初の約 1000 ステップは, 1 に近い値で開始しその後単調に減少させるべきである. すると初期期間で重みの順序づけが行われ残りのステップは, 微調整のために使われるからである. ただし最初の段階で, どのような学習率係数が最適であるかというのははっきりとしていない. なぜなら近傍集合の大きさもそれにより変化し, 状況を複雑にするからである. t は先ほども述べたが, 離散時間座標であり最終的には $rlen$ と等しくなる.

また N_c の半径, すなわちを $\sigma(t)$ は次式で示される.

$$\sigma(t) = 1.0 + (\sigma(0) - 1.0) \frac{rlen - t}{rlen} \quad (5.13)$$

$\sigma(0)$ はパラメータ *radius* として設定する.

近傍集合 $N_c = N_c(t)$ の大きさの選択は, $N_c(0)$ つまり最初の近傍集合をかなり広い範囲にとり, 時間とともにしだいに縮小させていく. このことにより, 学習の進行とともに粗調整から微調整へと変化させていくことができる. また $N_c(0)$ の値をあまりに小さい値にすると, マップの全体的な学習が行われなためである. N_c の初期半径, すなわち $\sigma(0)$ はマップ全体の直径の半分より大きくしてもよい.

6. 2に戻る.

以上のことを学習回数で指定した数だけ繰り返す.

式 (5.8) からわかるように, 勝者ノードとその近くのノードは全て, その時の入力ベクトルに近づく. このことにより, 類似したパターンが入力データが近い位置にマッピングされるようになるのである.

また学習終了時に, 平均量子化エラー (average quantization error) の算出が可能である. 平均量子化エラー q は次式で示される.

$$q = \sum_i^n h_{ci} \|x - m_i\|^2 \quad (5.14)$$

ただし, このとき $\alpha(t) = 1$ であり, 近傍関数がガウス関数の場合は $\sigma(t)$ は $\sigma(0)$ が使われる.

注意点

SOM の各パラメータは試行錯誤の後に最適なものを決定するしかない. 同じ入力データに対してはある最適なマップが存在するはずであり, そのとき平均量子化エラーは最小になると考えられる. そのため, 学習後にもう一度入力データを入力することによって定義される, 平均量子化エラーの算出が必要になるのである. つまり $m_i(0)$ の無作為な初期化と別のパラメータ設定による, かなりの数のマップが試されるべきである. そして, その中から最小量子化エラーをもつマップを選択することになるのである.

SOM の大きな特徴の一つとして, 高次元 (が許される) 入力データを 2次元のマップ上に写すことがあげられる. しかもこの写像は, 位相構造を保存する写像である. つまり入力データ空間 \mathcal{R}^n の密度分布が, 2次元マップ層上にうまく表現されているといえる. SOM は入力データ $x \in \mathcal{R}^n$ を確率データベクトルとすると, 高次元入力データベクトル x の確

率密度関数 $p(x)$ の 2 次元表示上への非線形射影とすることができる。これはコードブックベクトル m_i が、学習を重ねるごとに入力データ x の確率密度関数 $p(x)$ を近似するようになるからである。 $p(x)$ は、 x 空間で x が区間 dV_x に落ち込む確率を dV_x で割ったものである。また非線型写像というのは、マップ層への配置は学習の過程によって決められるものであり、その関係は線形的なものでは表せられないからである。SOM は、学習後のコードブックベクトルをなんらかの関数で 2 次元に射影しているわけではなく、SOM の学習そのものが 2 次元へのマッピングを意味している。なぜなら、マップ層は最初から 2 次元で定義されている。

SOMにより次元数が低減しているわけであり、直接人間の視覚による確認が可能になる。

5.1.3 LVQ(Learning Vector Quantization)

SOM との違い

LVQ(学習ベクトル量子化) は、SOM を教師有り学習に改良したものと考えることができる。SOM が入力データの 2 次元層への配置付けを行うのに対し、LVQ は入力データのクラス分類を行う。つまり SOM は入力データのクラスタ化(集団化)を行っているが、そのデータがどのクラスに属するかということは定義しない。それに対し LVQ では、明確に入力データのあるクラスへと分類するのである。LVQ のネットワーク構造は SOM と同様であるが、クラス分類を目的としているため入力層よりもマップ層のノード数が少ないのが普通である。また SOM と異なり勝者ノードの周りの近傍は、基本的な LVQ での学習中には定義されない。そのためコードブックベクトルの空間的な順序づけが、結果として起こるとは期待できない。また同じ理由から、SOM に比べ学習に必要な時間は短くなる。

LVQ のただ一つの目的は、入力データ空間内でクラス領域を定義することにある。つまりクラスの境界線を記述することにある。これは、マップ層のそれぞれのノードが自分の領土を持っていると考えることができる。入力ベクトルは入力層とマップ層間のコードブックベクトルにより、いずれかの領土に分類される。

学習アルゴリズム

ここでは LVQ の学習アルゴリズムについて説明する。

1. ネットワークの初期化

入力層とマップ層間のコードブックベクトルの初期値を、乱数を用いて小さな値に設定する。

2. 入力ベクトルの入力

入力層に入力ベクトル

$$\mathbf{x} = [\xi_1, \xi_2, \dots, \xi_n]^T \in \mathfrak{R}^n \quad (5.15)$$

を入力する。

3. コードブックベクトルと入力ベクトルの距離 (ユークリッド距離) の計算

上の入力ベクトルに対するマップ層のノード i のコードブックベクトルは

$$\mathbf{m}_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T \in \mathfrak{R}^n \quad (5.16)$$

で示される。

マップ層で各ノードのコードブックベクトルと入力ベクトルとのユークリッド距離を計算する。

$$\|\mathbf{x} - \mathbf{m}_i\| = \sqrt{\sum_{j=1}^n (\xi_j - \mu_{ij})^2} \quad (5.17)$$

4. ユークリッド距離が最小となるノードの選択

この競合によって選ばれたノードを勝者ノードとし、添え字 c によって表す。

$$\|\mathbf{x} - \mathbf{m}_c\| = \min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \quad (5.18)$$

5. コードブックベクトルの更新

LVQにはこの段階において、4つのアルゴリズムが存在する。以下順番に説明する。ただし、 $\alpha(t)$ ($0 < \alpha(t) < 1$) は学習率係数であり、SOMと同様に式 (5.11), (5.12) の二つがある。ただSOMと異なり初期値 $\alpha(0)$ は小さい値、たとえば0.1よりも小さい値にしたほうがよい。

LVQ1

もっとも基本となるものである。勝者ノードが正しく識別された場合は、そのノードのコードブックベクトルを入力ベクトルに近づくように修正する。誤って識別された場合は、逆に遠くなるように修正する。

$$\mathbf{m}_c(t+1) = \begin{cases} \mathbf{m}_c(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_c(t)] & \text{(勝者ノードが正しく分類されたとき)} \\ \mathbf{m}_c(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{m}_c(t)] & \text{(勝者ノードが誤って分類されたとき)} \end{cases} \quad (5.19)$$

$$m_i(t+1) = m_i(t) \quad i \neq c \quad (5.20)$$

勝者ノードが正しく分類されたとき、というのは x と m_c が同じクラスに属することを意味している。 x が正しく分類される、ということもできる。逆に勝者ノードが誤って分類されたとき、というのは x と m_c が異なるクラスに属することを意味している。 x が誤って分類される、ということもできる。

また式 (5.20) は、勝者ノード以外のコードブックベクトルは更新されないことを意味している。

OLVQ1

OLVQ1(Optimized-learning-rate LVQ1) は、LVQ1 の学習が早く収束するように学習率計数 α の値を最適に決定しようとしている。

$$m_c(t+1) = \begin{cases} m_c(t) + \alpha_c(t)[x(t) - m_c(t)] & \text{(勝者ノードが正しく分類されたとき)} \\ m_c(t) - \alpha_c(t)[x(t) - m_c(t)] & \text{(勝者ノードが誤って分類されたとき)} \end{cases} \quad (5.21)$$

$$m_i(t+1) = m_i(t) \quad i \neq c \quad (5.22)$$

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \quad (5.23)$$

$$s(t) = \begin{cases} +1 & \text{(勝者ノードが正しく分類されたとき)} \\ -1 & \text{(勝者ノードが誤って分類されたとき)} \end{cases} \quad (5.24)$$

また、 α の初期値は 0.3 としている。

LVQ2.1

LVQ2.1 は LVQ1 が誤認識したとき、つまり勝つはずでないノードがわずかの差で勝者ノードになってしまった場合に対して改良が加えられている。勝者ノードとその次に反応したノードのコードブックベクトルを同時に更新する。この2つのノードは、一方は正しいクラスに、他方は違うクラスに属していなければならない。正しいクラスに属するノードのコードブックベクトルは入力ベクトルに近づくように、間違ったクラスに属するノードのコードブックベクトルは遠くなるように修正する。

$$\begin{aligned} m_i(t+1) &= m_i(t) - \alpha(t)[x(t) - m_i(t)] \\ m_j(t+1) &= m_j(t) + \alpha(t)[x(t) - m_j(t)] \end{aligned} \quad (5.25)$$

ここで、 m_i と m_j は x に最近接な二つのコードブックベクトルである。このとき x と m_j は同じクラスに属するが、 x と m_i は異なるクラスに属する。ただし、以下の式を満たさねばならない。

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s, \quad \text{ただし } s = \frac{1-w}{1+w} \quad (5.26)$$

d_i, d_j は、それぞれ m_i, m_j からの x のユークリッド距離である。 w はウィンドウの相対幅といわれるパラメータであり、プログラム中では w_{in} として設定する。式(5.26)を満たすとき、 x が相対幅 w のウィンドウに落ちたという。また w は、 $0.2 < w < 0.3$ 程度がよい。

LVQ3

LVQ3ではLVQ2に加え、2つのノードが共に正しいクラスに属している場合にもコードブックベクトルを更新する。

$$\begin{aligned} m_i(t+1) &= m_i(t) - \alpha(t)[x(t) - m_i(t)] \\ m_j(t+1) &= m_j(t) + \alpha(t)[x(t) - m_j(t)] \end{aligned} \quad (5.27)$$

ここで、 m_i と m_j は x に最近接な二つのコードブックベクトルである。このとき x と m_j は同じクラスに属するが、 x と m_i は異なるクラスに属する。さらに、 x の値はウィンドウに落ちなければならない。もし x, m_i, m_j が同じクラスに属し、 $k \in \{i, j\}$ ならば次式を得る。

$$m_k(t+1) = m_k(t) + \varepsilon\alpha(t)[x(t) - m_k(t)] \quad (5.28)$$

ε はプログラム中ではパラメータ *epsilon* として設定する値である。 $w = 0.2$ または 0.3 に関して対応できる ε の値は 0.1 と 0.5 の間にある。

6. 2に戻る.

各LVQアルゴリズムについて

第5.1.3項で示したように、LVQアルゴリズムとしてLVQ1, OLVQ1, LVQ2.1, LVQ3が存在する。普通の統計的パターン認識では、これらのどの方法をとってもほぼ同じ精度

で答えが得られる。LVQ1, LVQ2.1は、より頑強な処理過程を行うようにアルゴリズムは定義されている。このためコードブックベクトルは、より長い学習回数の後でさえも一定値に留まると仮定している。LVQ1は、学習率を早く収束するように設定されており、概ね最適化できる。これがOLVQ1である。LVQ2.1は、クラス境界からコードブックベクトルの相対距離は最適化されている。ところが、コードブックベクトルがクラス分布の形を記述するように最適に配置されているという保証はない。よって、LVQ2.1では小さい学習率の値を使い、学習回数の数も制限して差分形の形でのみ使用されるべきである。

学習はまずOLVQ1で行うのがよい、他のアルゴリズムはOLVQ1での学習終了後に引き続き使用することもできる。つまりOLVQ1の学習アルゴリズムのみでも十分な場合も多いが、最終的に認識精度(容認正確さ)を改良を試みるためには、学習率の低い初期値を用いLVQ1, LVQ2.1, LVQ3を使い学習を続けることも可能である、ということである。

5.2 入力データの洗練化

5.2.1 概要

入力データには既に述べた、橋梁点検台帳のデータを用いる。鋼橋とコンクリート橋では入力データ項目に違いがあるので別々に扱う。また鋼橋のデータ中には橋梁の損傷度が4のものは存在していない。

入力データの洗練化の手順を以下に示す。

1. SOMによる入力データの分析

SOMは、類似した入力データを近くに配置したマップを生成する。

2. 疑わしいデータの抽出

生成されたマップを見てラベル付けが疑わしいと思われるデータを選び出す。つまり、近傍にマッピングされた他のデータとラベル⁶の違うものを選び出す。

3. LVQによる入力データの再評価

LVQを用い先ほど疑わしいと判断したデータの評価をする。LVQはSOMとは異なり、明確にそのデータがどのクラス⁷に属するかを判断できる。

4. 入力データの変更

LVQでの評価が最初の評価と違う場合には、そのラベルをLVQでの評価で書き直す。

5. 必要ならば以上のことを繰り返す。

⁶ここでは橋梁の損傷度

⁷これも橋梁の損傷度

5.2.2 鋼橋

SOMによる入力データの分析

まず、入力データにおける各パラメータの最適値を見つけ出さねばならない。ここでは量子化エラーと表示ラベル数の二つの値を参考にして最適値を見つけ出す。SOM.PAK version3.1⁸の場合、一つのノードには一つのラベルしかかけない。つまり同じ場所にマッピングされたデータは上書きされてしまう、という問題点がある。表示ラベル数とは、入力データ中いくつのデータがマップ上に表示されているかということである。

まずは、マップサイズを決めなければならない。これはノード数を決めることと同義である。量子化エラーと表示ラベル数はマップサイズが大きいほどよい結果が得られ、またマップサイズには限界はない。しかしマップサイズが大きすぎると入力データが分散してしまい、またディスプレイや紙面に入りきらないということになってしまう。つまり理論上あるいは演算上はマップサイズには制限は無いが、視覚的に判断するための媒体の制約からマップサイズには限界があるといえる。つまりできるだけ小さいマップサイズで、量子化エラーが小さく、表示ラベル数が多いものを選ぶことになる。

図 5.7, 表 5.1 に学習回数 30000 回と 50000 回時のマップサイズ ($X \times Y$) と量子化エラーの関係を、図 5.8, 表 5.2 に学習回数 50000 回時のマップサイズと表示ラベル数の関係を示す。ただしマップの初期化時には乱数を用いるため、パラメータ設定が同じでも量子化エラー、および、マップが同じものになるとは限らない⁹。他のパラメータは α (学習率係数の初期値) が 0.9, $radius$ (近傍集合の半径) はマップサイズの 1/2 を目安として用いている。

入力データの総数は 55 である。すなわち、表示ラベル数は最大で 55 である。これらの図と表を見るとマップサイズが 15×15 の時、量子化エラーの値はそれまでに比べて極めて小さくなり、またほぼ全てのラベルが表示されている。これらの図と表から最適なマップサイズは $15 \times 15 \sim 20 \times 20$ 程度であると推測できる。

⁸SOM.PAK については Kohonen らのホームページを参照。

⁹パラメータ $rand$ を同じにした場合は同じになる。

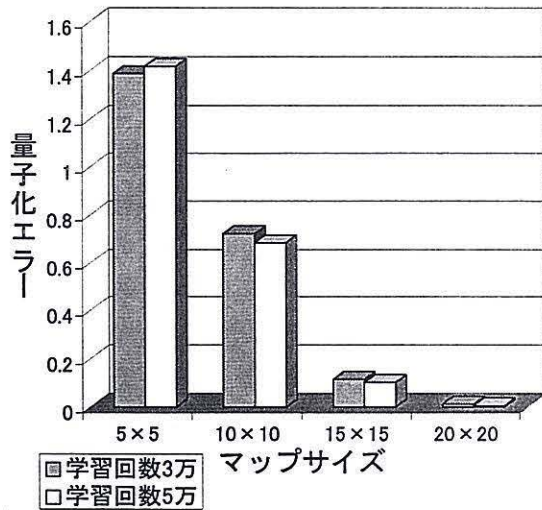


図- 5.7 マップサイズと量子化エラー

表- 5.1 マップサイズと量子化エラー

マップサイズ	学習回数	
	30000	50000
5 × 5	1.387806	1.417839
10 × 10	0.721793	0.683966
15 × 15	0.115413	0.102622
20 × 20	0.011373	0.005309

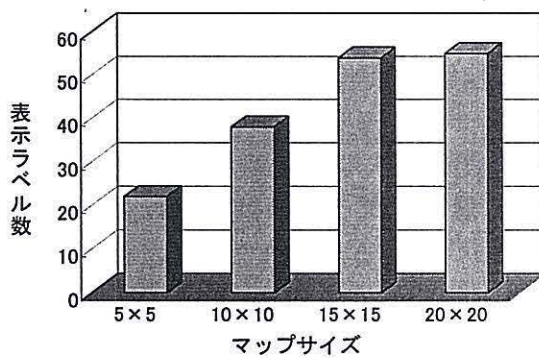


図- 5.8 マップサイズと表示ラベル数

表- 5.2 マップサイズと表示ラベル数

マップサイズ	表示ラベル数
5 × 5	22
10 × 10	38
15 × 15	54
20 × 20	55

次に学習回数を決めねばならない。図 5.9 にマップサイズが 15×15 時の学習回数と量子化エラーの関係を、図 5.10 にマップサイズが 20×20 時の学習回数と量子化エラーの関係を示す。これらの図は、ある一つの学習で途中の学習回数ごとに結果を抜き出したものではない。つまり、プロットごとにコードブックベクトルの初期化からやり直したものである。なぜなら乱数で初期化した多くのマップを造ることが、最適なマップを造る一番の近道だからである。図 5.9, 図 5.10 で Y 軸の大きさのとりかたが異なっているので注意が必要である。また *radius* は 15×15 の時 8, 20×20 の時 10 とした。

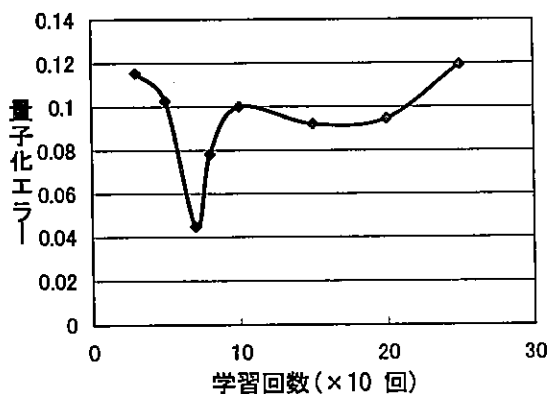


図- 5.9 マップサイズ 15×15

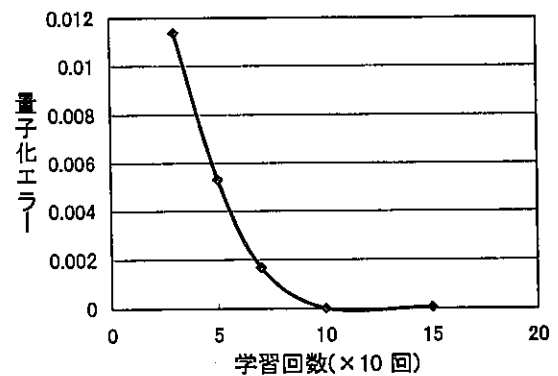


図- 5.10 マップサイズ 20×20

図 5.9 を見ると、学習回数 70000 回で量子化エラーが最も小さくなっている。グラフの形からして偶然量子化エラーが小さいところに陥ったものと考えられる。乱数でマップを初期化するためか、時々量子化エラーが周囲に比べ極めて小さくなることもある。

図 5.10 においては、学習回数 100000 回で最も量子化エラーが小さくなり、その後増加している。これはそれを境に過学習におちいったためと考えられる。よってマップサイズ 20×20 においては学習回数 100000 回を採用する。

マップサイズ 15×15 の時、図 5.9 の範囲以降にまた局所的に量子化エラーが小さくなる点が存在する可能性がある。しかしマップサイズ 20×20 の時に、学習回数 100000 回で量子化エラーは最小になっている。よって、マップサイズが 20×20 よりも小さい時に 100000 回以上の学習回数を必要にするとは考えにくいので、マップサイズ 15×15 においては学習回数 70000 回を採用する。

またマップサイズ 18×15 で学習回数 100000 回、*radius* が 9 のときに量子化エラー 0.000364 という結果を得たのでこれも採用する。

表 5.3 に採用した SOM の各パラメータ, および, 結果を示す.

表- 5.3 採用した SOM

マップサイズ	学習回数	量子化エラー	表示ラベル数
15 × 15	70000	0.044583	53
18 × 15	100000	0.000364	55
20 × 20	100000	0.000007	55

SOM の結果から疑わしいと思われるデータを抽出するわけであるが, その作業は人間の目で行うため 1 つのマップではなく 3 つのマップを用意した.

疑わしいと思われるデータは, 周りのラベルと異なるラベルが表示されているものである. ラベルは橋梁の損傷度と橋梁を識別するためのアルファベット 2 文字で表示しているため, 正確には最初の数字だけで判断する.

図 5.11 にマップサイズ 15 × 15, 図 5.12 にマップサイズ 18 × 15, 図 5.13 にマップサイズ 20 × 20 の SOM の結果を示す. 黒で囲まれているのが疑わしいと思われるデータである.

抽出した疑わしいデータは, 1AB, 1BA, 1BB, 1BE, 1BH, 1BN, 1CC, 3AC, 3AJ, 3BG, 3BS, 3BX, 3CB である.

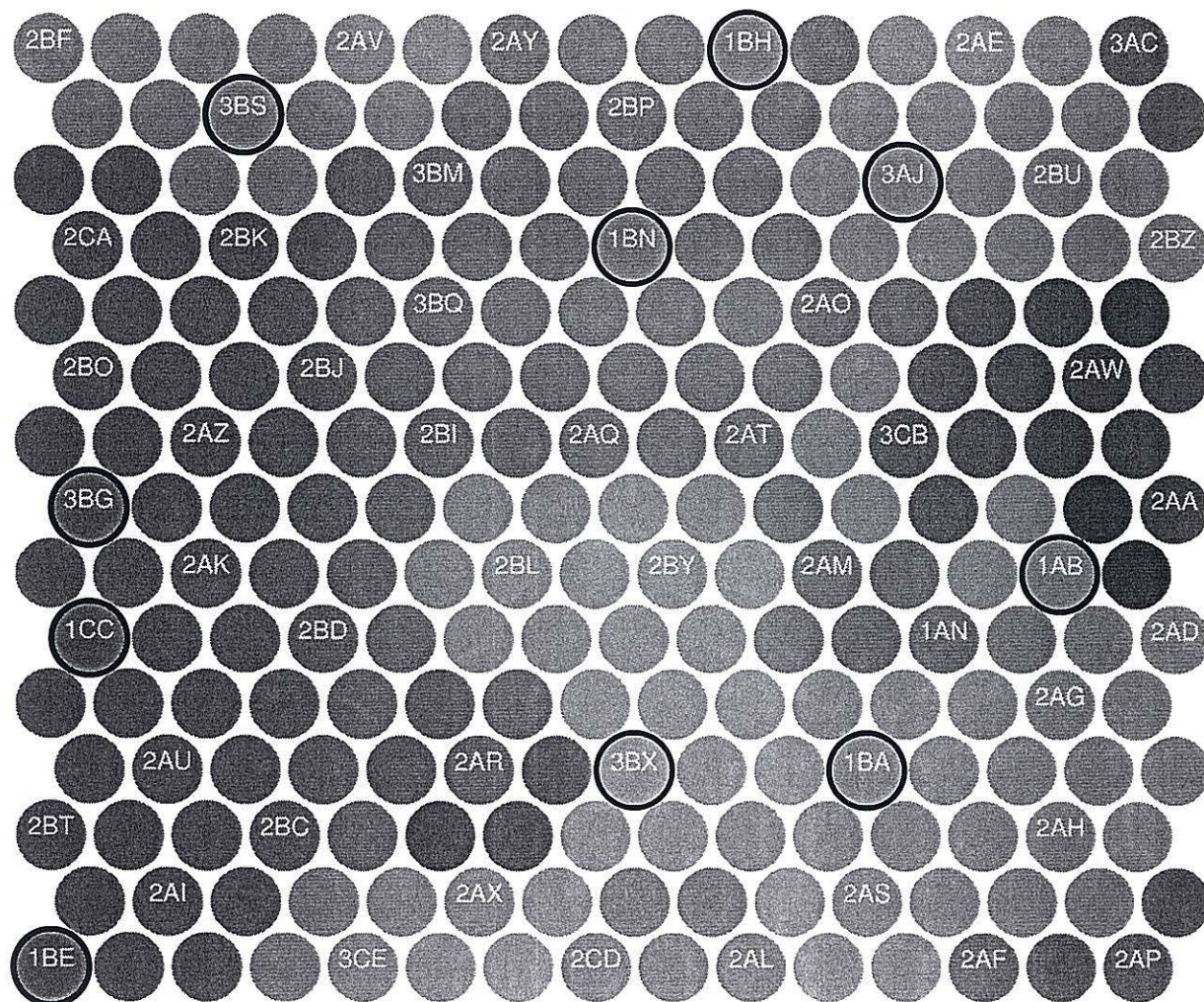


図- 5.11 マップサイズ 15 × 15, 学習回数 70000

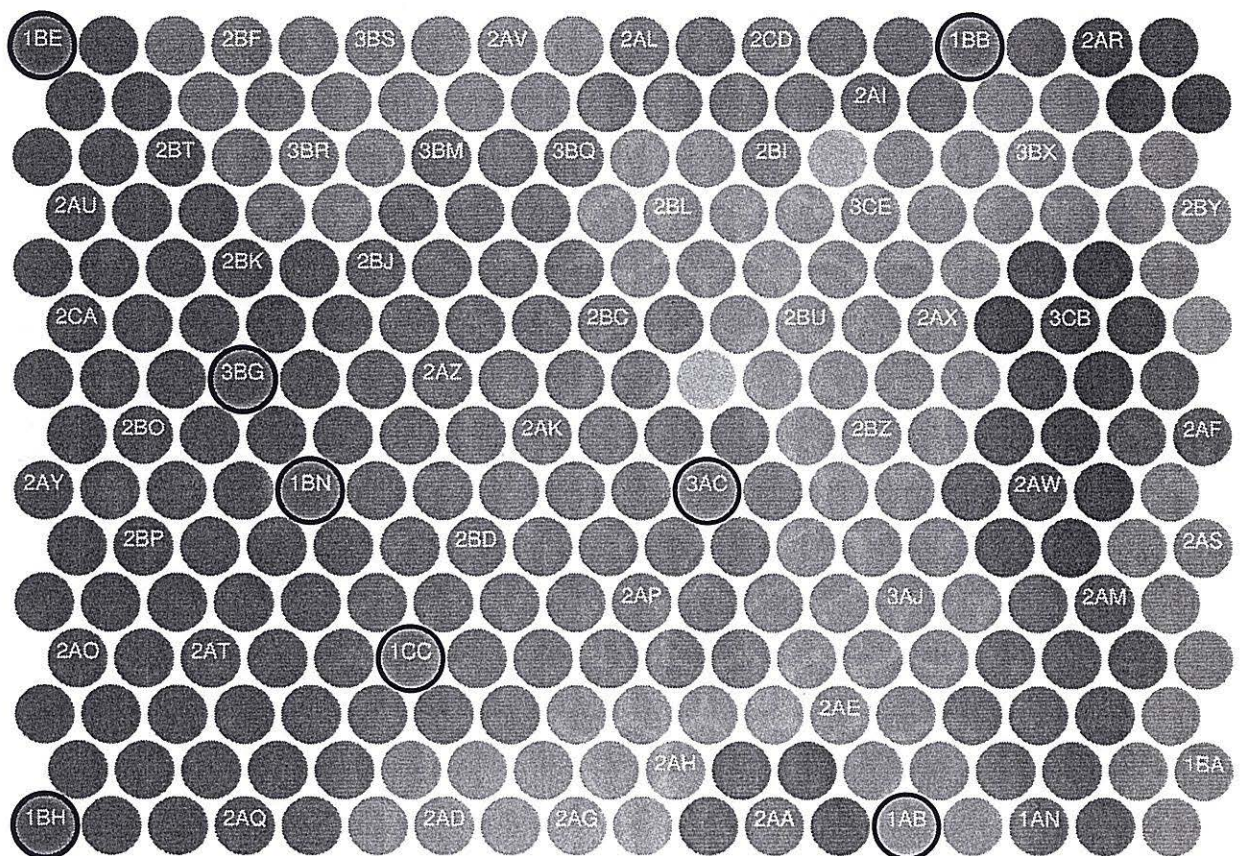


図- 5.12 マップサイズ 18 × 15, 学習回数 100000

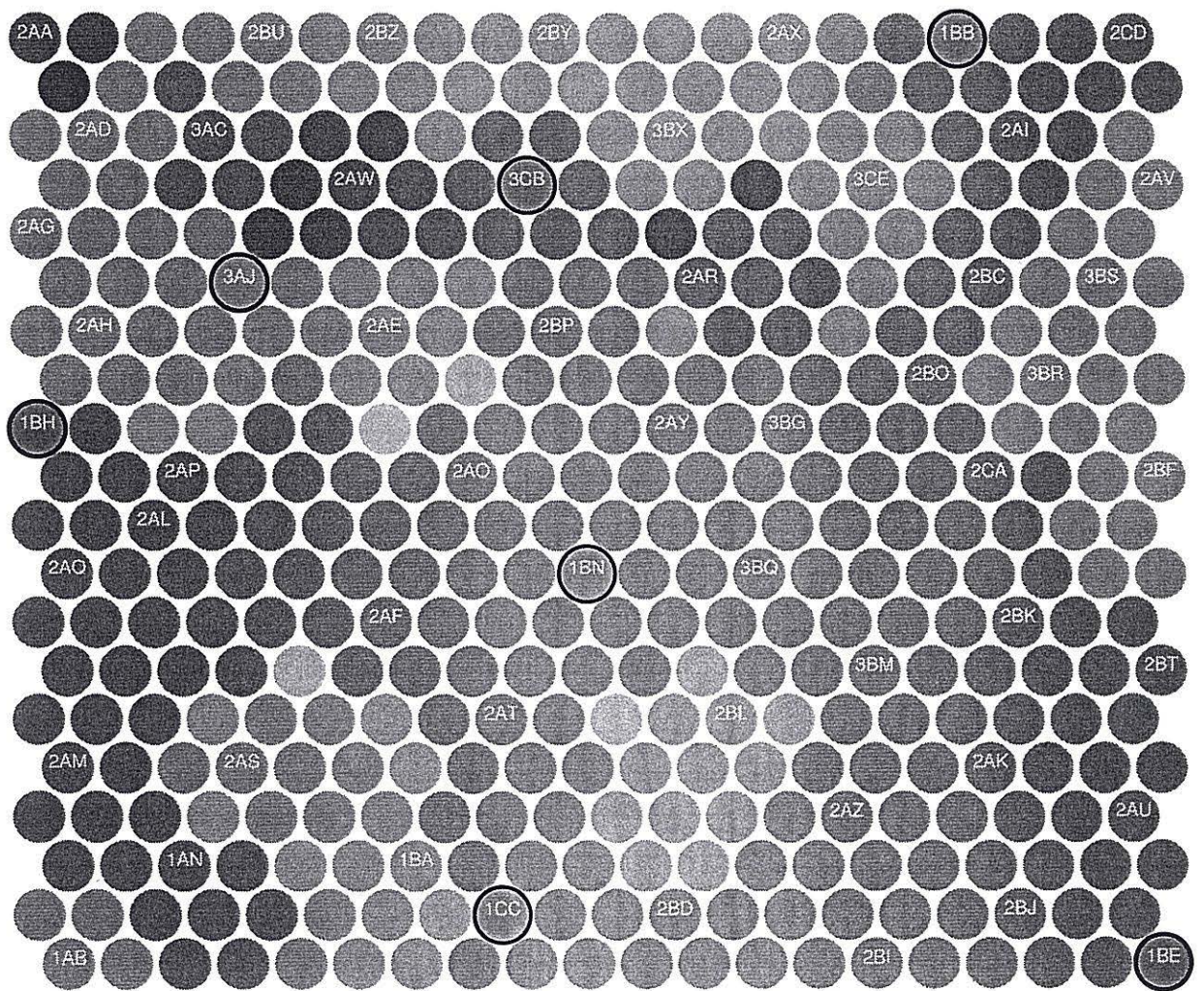


図- 5.13 マップサイズ 20 × 20, 学習回数 100000

LVQ による入力データの再評価

SOM で抽出したデータは、あくまで疑わしいと思われるデータである。これを LVQ により明確に判断をください。

LVQ で学習に用いるデータは、先ほど SOM で疑わしいと判断したデータを削除しないでそのまま用いた。その理由を以下に示す。

1. SOM で抽出したデータが本当に間違っているかどうかはまだわからない。
2. 入力データは多いほうがよい。
3. 間違っているデータは LVQ の学習で淘汰される。

LVQ の各パラメータも、trial and error で決定する必要がある。まず SOM のマップサイズに当たる NOC^{10} (Number Of Codebook vector) を決定しなければならない¹¹。またそれは、LVQ の構造上 SOM よりも数は小さくなる。NOC は学習に用いた入力データをプログラム *accuracy* で処理して算出した“総合的な認識精度 (Total accuracy)” から判断する。図 5.14 に NOC と認識精度の関係を示す。学習には *olvq1* を使用した。また学習回数は 2200 回とした。

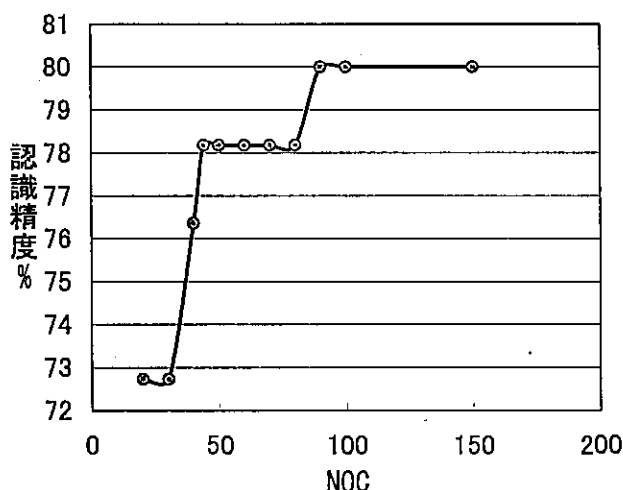


図- 5.14 NOC と認識精度の関係

これを見ると NOC が 30 で認識精度は上昇をはじめ、90 で一定の値に落ち着いている。よって適切な NOC 値はこの間にあると考えられる。最終的に、 NOC が 44 でプログラム *balance* を通さなかったときの認識精度が 85.45% と高かったため、これを採用するこ

¹⁰プログラム上では小文字の *noc* で指定する。

¹¹コードブックベクトルの数とノードの数は等しい

とにした。この時の学習回数は2200回である。認識精度が高いので学習回数はこの程度でよいと判断した。このときの *accuracy* の結果を以下に示す。

Recognition accuracy:

```
2: 37 entries 100.00 %
3: 10 entries 70.00 %
1: 8 entries 37.50 %
```

Total accuracy: 55 entries 85.45 %

この学習したコードブックベクトルを使い、先ほど SOM で抽出したデータの再評価をする。つまり、クラス分類プログラム *classify* で処理する。また先ほど SOM 抽出したデータをデータ項目の値はそのままクラスを1~3でつけ、プログラム *accuracy* で処理するという手法もとった。書き直したデータの例を示す。

1ABについて

14
2 2 3 2 3 2 3 3 1 2 2 2 2 3 1
2 2 3 2 3 2 3 3 1 2 2 2 2 3 2
2 2 3 2 3 2 3 3 1 2 2 2 2 3 3

最初の14はデータ項目が14であることを示している。最後の15番目の数字がクラスを示している。この入力データと先ほど採用した学習済みのコードブックベクトルを用い、プログラム *accuracy* で処理した結果を以下に示す。

Recognition accuracy:

```
1: 1 entries 100.00 %
2: 1 entries 0.00 %
3: 1 entries 0.00 %
```

Total accuracy: 3 entries 33.33 %

プログラム *classify* だけでなく *accuracy* も使用したのは *classify* はどのクラスに分類されるかのみを示すが、*accuracy* ではその精度が%単位で数値的に表示されるからである。

これは、1ABのクラスは1であるということを示していると考えられる。つまり、SOMでの疑わしいとの判断は間違っていたことになる。SOMでの判断は疑わしいと思われる、という程度までしかできないのでこの結果になんら問題はない。

表 5.4 に、*classify* による解析結果をまとめて示す。*accuracy* による解析結果も全く同じである。*accuracy* による解析結果は全てのデータが、あるクラス (*classify* の出力したクラス) で認識精度が 100% という結果になった。

表- 5.4 *olvq1* による解析結果

データ名	元の評価	<i>olvq1</i> による評価
1AB	1	1
1BA	1	1
1BB	1	3
1BE	1	2
1BH	1	2
1BN	1	2
1CC	1	2
3AC	3	2
3AJ	3	2
3BG	3	2
3BS	3	3
3BX	3	3
3CB	3	3

元の評価と *olvq1* による評価が異なった場合にはその入力データを書き換え、先ほどのパラメータ設定 (*NOC*=44, 学習回数=2200) で再度学習をした。この学習したコードブックベクトルと書き換えた入力データ (すなわち学習に用いた入力データ) を用いてプログラム *accuracy* での処理結果を示す。

```

Recognition accuracy:

      2:   44 entries 100.00 %
      3:    8 entries 100.00 %
      1:    3 entries 100.00 %

Total accuracy:   55 entries 100.00 %

```

全てのクラスの認識精度が 100% となり、当然総合的な認識精度も 100% となった。

最終的な入力データの決定

ここで第 5.2.2 項での結果を反映した入力データを用い、再び SOM で学習した結果を図 5.15 に示す。各パラメータはマップサイズが 20×20 、学習回数が 10000 回、 α が 0.9、 $radius$ が 10 である。その結果量子化エラーは 0.000010 であり、全てのデータが表示されている。

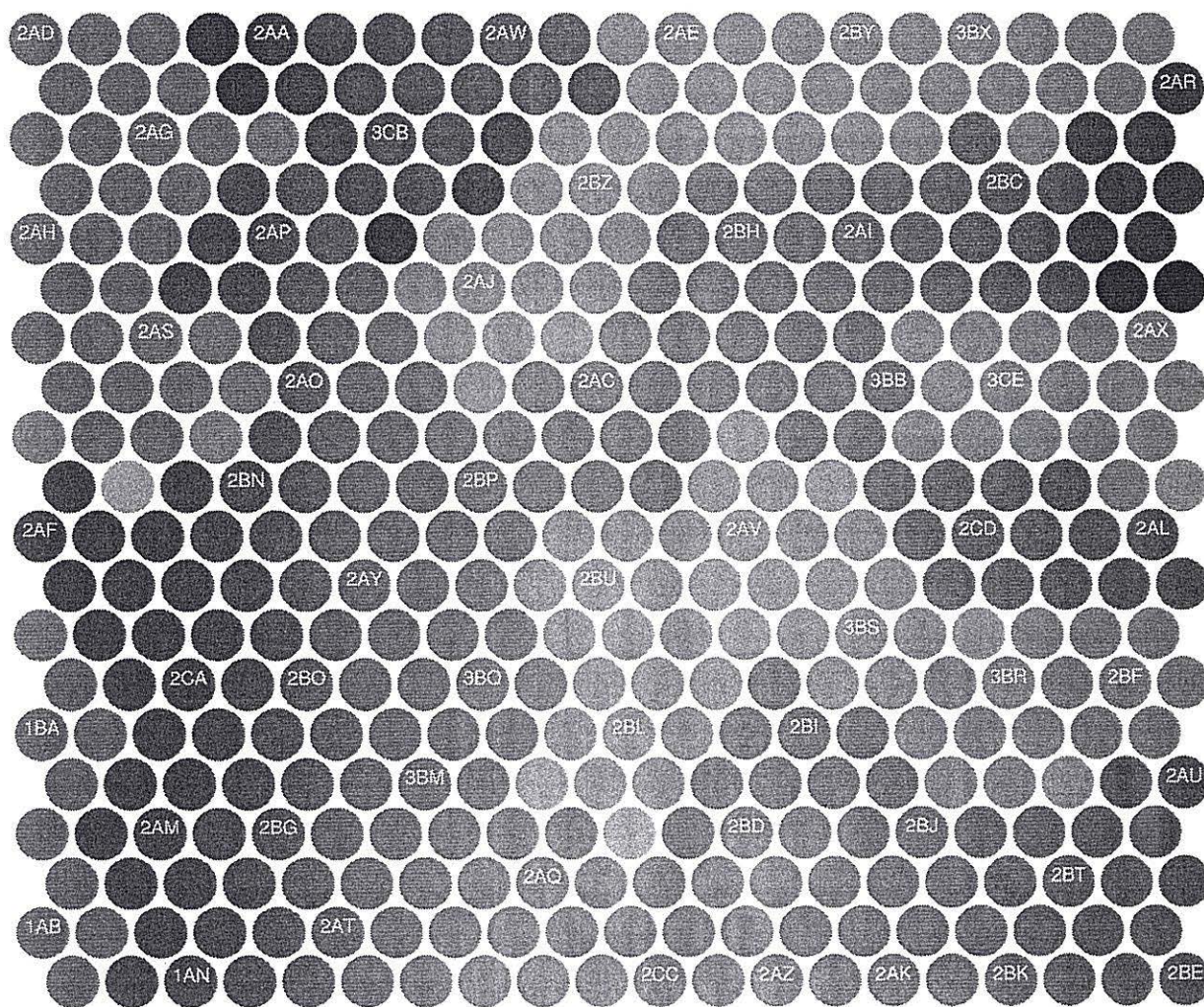


図- 5.15 入力データ修正後の SOM の結果

図 5.15 を見ると、近傍がほぼ正しく表現されている。また、第 5.2.2 項で LVQ による各クラスの認識精度は全て 100%であった。よって、鋼橋の橋梁点検データの見直しはこれで終了する。

5.2.3 コンクリート橋

コンクリート橋も鋼橋と同様にして，入力データの見直しを行う。

SOM による入力データの分析

まず，各パラメータの最適値を探し出す。図 5.16，表 5.5 に学習回数 30000 回時のマップサイズ ($X \times Y$) と量子化エラーの関係を，図 5.17，表 5.6 に学習回数 30000 回時のマップサイズと表示ラベル数の関係を示す。他のパラメータは α が 0.9， $radius$ はマップサイズの 1/2 程度を用いている。

量子化エラーは鋼橋に比べ大きい，それは量子化エラーの定義，式 5.14 から当然のことと考えられる。入力データの総数は 119 であるが，データ項目が全く同じでラベルの異なるデータが二組 (3dj と 2eb，3ck と 2cr) 存在するため，表示ラベル数は最大で 117 である。これらの図と表を見るとマップサイズが 20×20 で総入力データの 9 割以上が， 25×25 でその全てが表示されている。量子化エラーは若干高めではあるが，これは学習回数が少ないためであると考えられる。よって，最適なマップサイズは $20 \times 20 \sim 25 \times 25$ 程度であると推測できる。

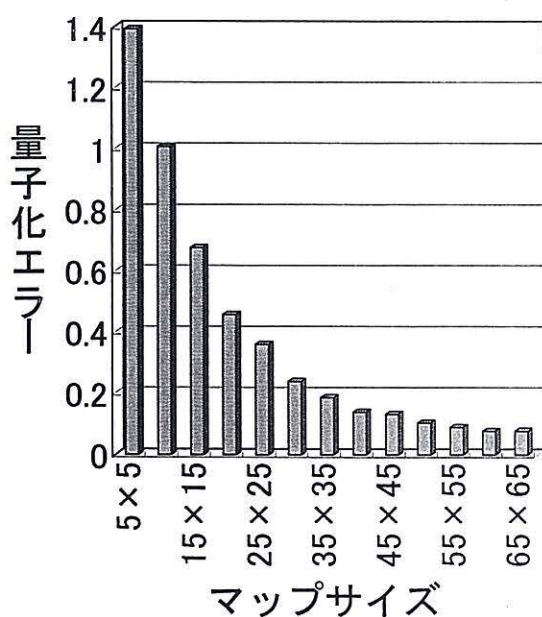


図- 5.16 マップサイズと量子化エラー

表- 5.5 マップサイズと量子化エラー

マップサイズ	量子化エラー
5 × 5	1.392529
10 × 10	1.006540
15 × 15	0.676666
20 × 20	0.456550
25 × 25	0.360308
30 × 30	0.239102
35 × 35	0.184920
40 × 40	0.138012
45 × 45	0.131188
50 × 50	0.103794
55 × 55	0.089323
60 × 60	0.076457
65 × 65	0.076896

表- 5.6 マップサイズと表示ラベル数

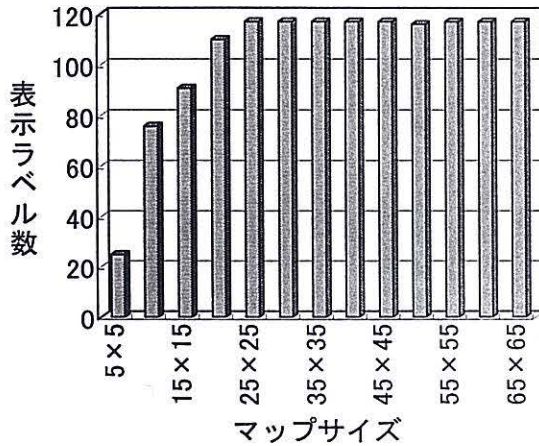


図- 5.17 マップサイズと表示ラベル数

マップサイズ	表示ラベル数
5 × 5	25
10 × 10	76
15 × 15	91
20 × 20	110
25 × 25	117
30 × 30	117
35 × 35	117
40 × 40	117
45 × 45	117
50 × 50	116
55 × 55	117
60 × 60	117
65 × 65	117

そこでマップサイズ 20×20 、および、 25×25 を採用することにした。また、正方形以外のマップをみるためマップサイズ 20×30 も採用することにした。 $15 \times 15 = 225$ 、 $25 \times 25 = 625$ であり、 $20 \times 30 = 600$ であるので 20×30 は妥当であると考えられる。 $radius$ は 20×20 で 10、 20×30 で 18、 25×25 で 13 とした。

次に学習回数を決めねばならない。図 5.18 にマップサイズが 20×20 時の学習回数と量子化エラーの関係を、図 5.19 にマップサイズが 20×30 時の学習回数と量子化エラーの関係を、図 5.20 にマップサイズが 25×25 時の学習回数と量子化エラーの関係を示す。

これらの図も鋼橋と同様、プロットごとにコードブックベクトルの初期化からやり直した結果である。

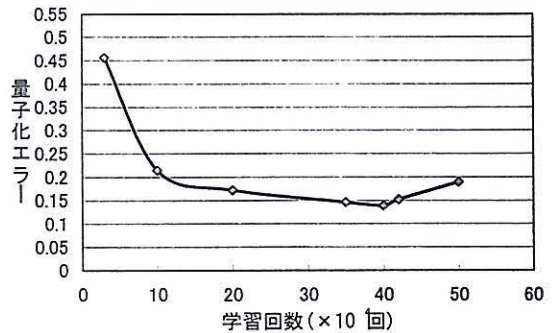


図- 5.18 マップサイズ 20×20

これらの図では、3 つとも学習回数 400000 回で量子化エラーは最小になっている。それ以降は過学習におちいったと考えられる。

鋼橋に比べコンクリート橋は入力データが多いのだが、その差はマップサイズよりも学習回数に現れているといえる。ただし各パラメータは入力データの内容にも左右されるので、一概にそう断言することはできない。

表 5.7 に採用した SOM の各パラメータ、および、結果を示す。

図 5.21 にマップサイズ 20×20 、学習回数 400000 回、図 5.22 にマップサイズ 20×30 、学習回数 400000 回、図 5.23 にマップサイズ 25×25 、学習回数 400000 回の SOM の結果を示す。黒で囲まれているのが疑わしいと思われるデータである。データ項目の値が同じでありながらラベルの違う二組のデータ (3dj と 2eb, 3ck と 2cr) と、SOM で疑わしいと思われるデータを抽出した。抽出したデータは 1bd, 1cc, 1ct, 1cv, 1dm, 1dp, 1el, 2bl, 2cr, 2dd, 2eb, 2ef, 3aa, 3ap, 3ax, 3be, 3bf, 3ce, 3ck, 3cp, 3cs, 3dj, 3dq, 3dt, 3du, 3ej, 3ek, 4ac, 4ag, 4an である。

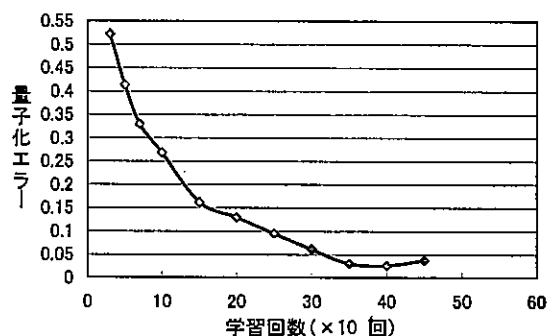


図- 5.19 マップサイズ 20×30

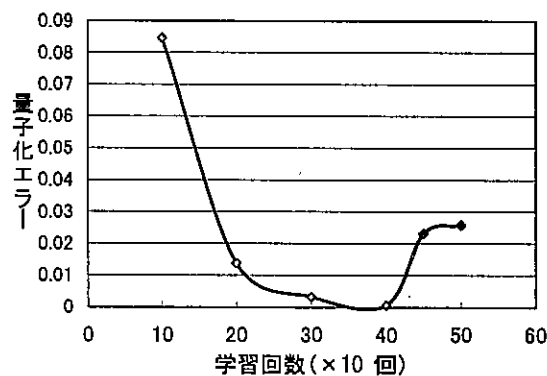


図- 5.20 マップサイズ 25×25

表- 5.7 採用した SOM

マップサイズ	学習回数	量子化エラー	表示ラベル数
20×20	400000	0.138857	112
20×30	400000	0.026337	117
25×25	400000	0.000645	117

最終的な入力データの決定

ここで第 5.2.2 項での結果を反映した入力データを用い、再び SOM で学習した結果を図 5.15 に示す。各パラメータはマップサイズが 20×20 、学習回数が 10000 回、 α が 0.9、 $radius$ が 10 である。その結果量子化エラーは 0.000010 であり、全てのデータが表示されている。

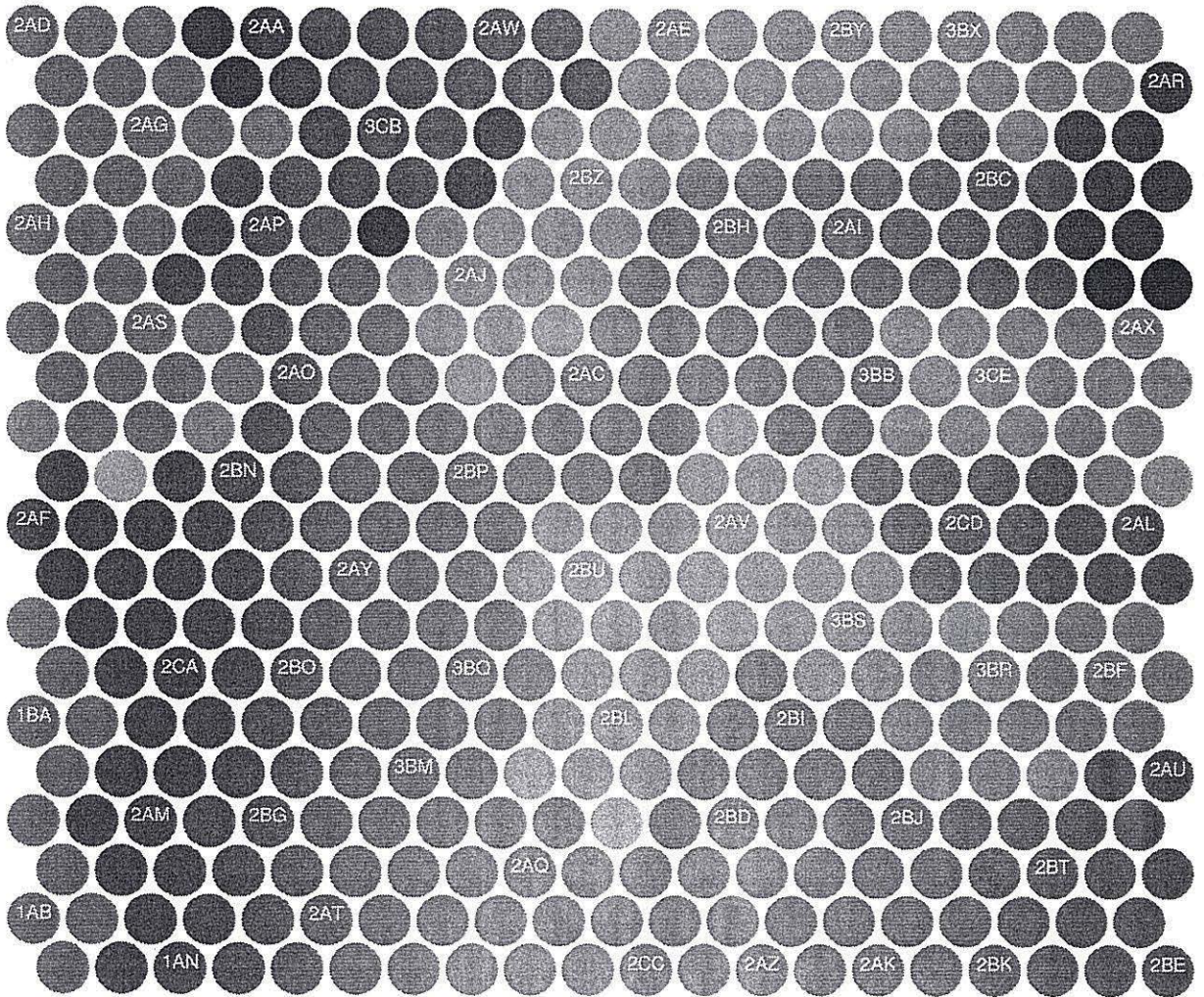


図- 5.15 入力データ修正後の SOM の結果

図 5.15 を見ると、近傍がほぼ正しく表現されている。また、第 5.2.2 項で LVQ による各クラスの認識精度は全て 100%であった。よって、鋼橋の橋梁点検データの見直しはこれで終了する。

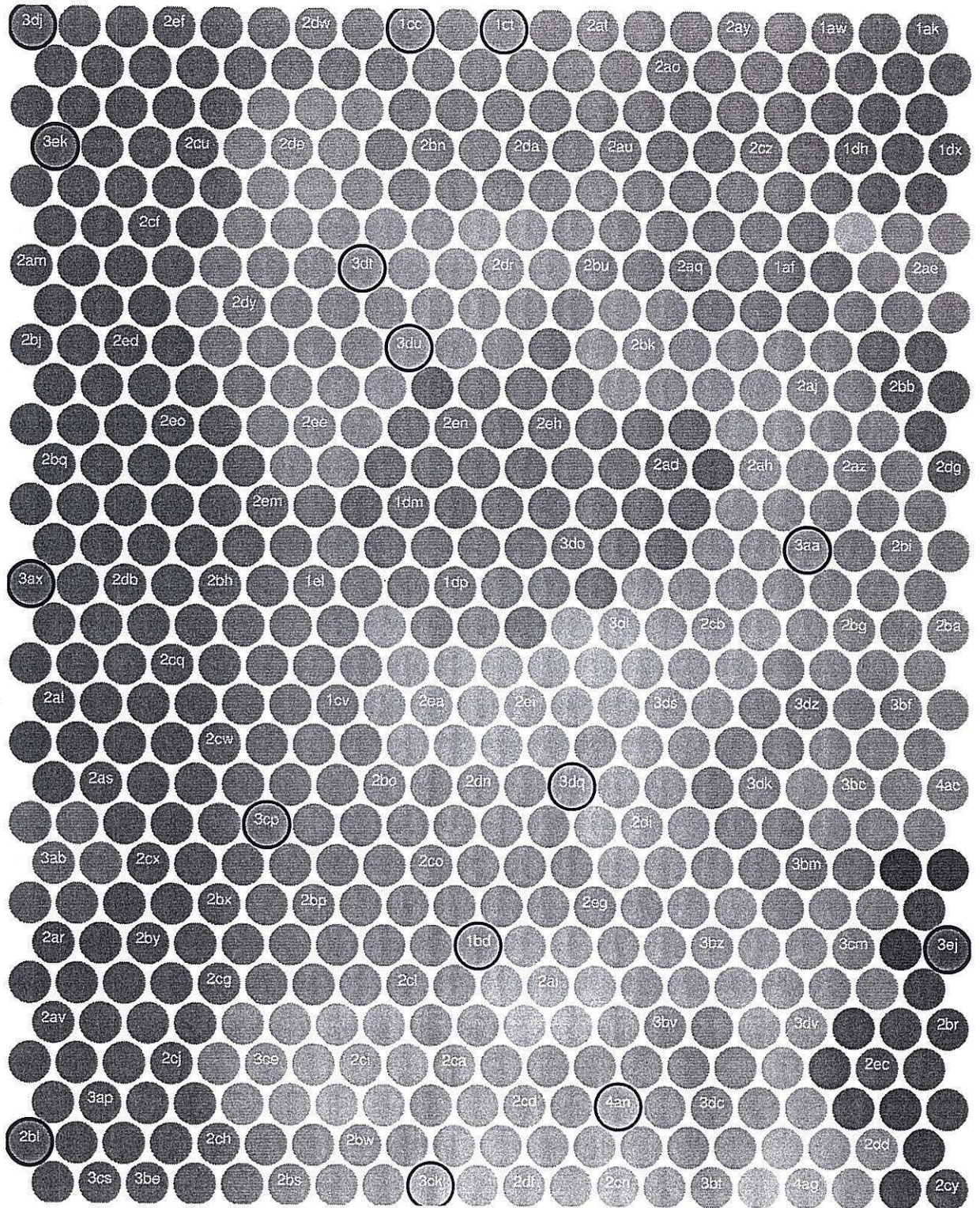


図- 5.22 マップサイズ 20 × 30, 学習回数 400000

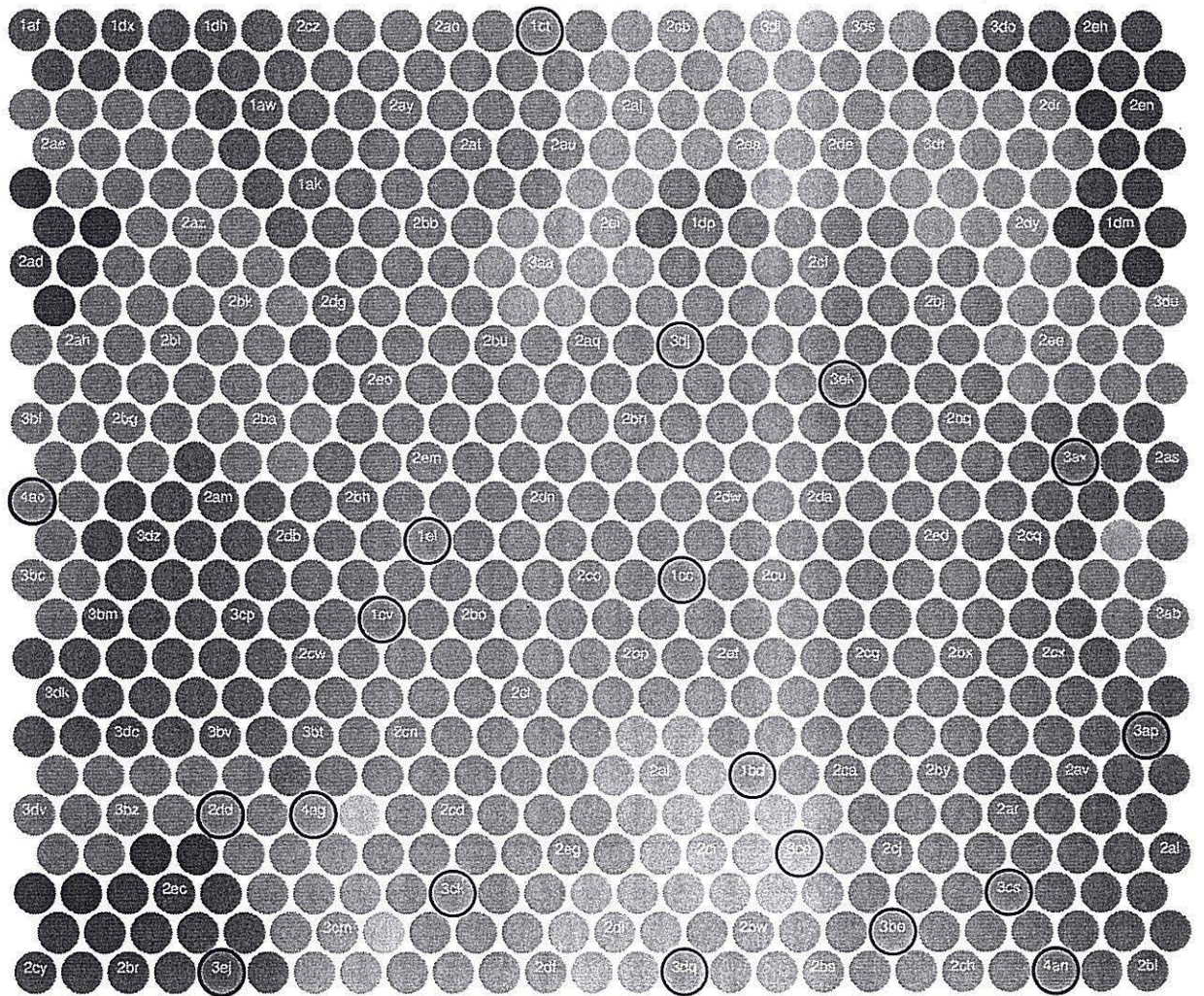


図- 5.23 マップサイズ 25 × 25, 学習回数 400000

LVQによる入力データの再評価

SOMで抽出した疑わしいと思われるデータを、LVQにより再評価する。鋼橋と同様にLVQで学習に用いるデータは、先ほどSOMで疑わしいと判断したデータを削除しないでそのまま用いた。

まずパラメータ *NOC* を決定する。*NOC* は学習に用いた入力データをプログラム *accuracy* で処理して算出した、全体の認識精度 (Total accuracy) から判断する。図 5.24 に *NOC* と認識精度の関係を示す。学習には *olvq1* を使用した。また学習回数は 4000 回とした。

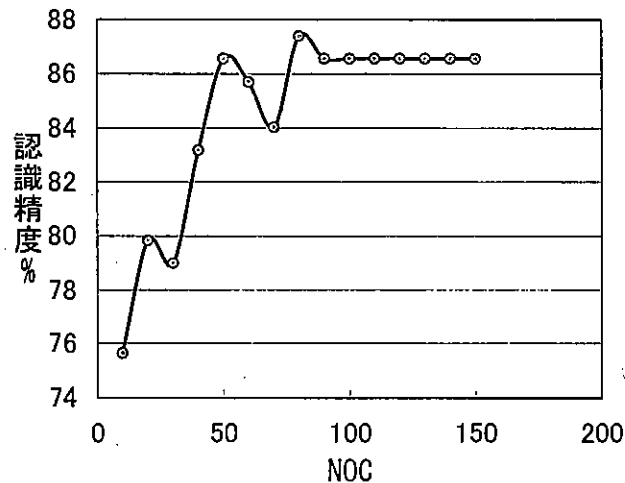


図- 5.24 *NOC* と認識精度の関係

NOC が 80 で認識精度が最も高く 87.39%であった。よって、これを採用することにした。認識精度が十分高いため、学習回数は 4000 回をそのまま採用した。その時の *accuracy* の結果を以下に示す。

Recognition accuracy:

2:	75 entries	98.67 %
3:	29 entries	86.21 %
1:	12 entries	41.67 %
4:	3 entries	0.00 %

Total accuracy: 119 entries 87.39 %

クラス 4 の認識精度が 0.00% になってしまったのは、入力データにクラス 4 のものが 3 つしか含まれていなかったためと考えられる。

この学習で得られたコードブックベクトルを用いて、先程抽出したデータの再評価をする。クラス分類プログラム *classify* と *accuracy* を使用した手法の二つをとった。プログラム *accuracy* を使用する手法は、抽出したデータをデータ項目の値はそのままクラスを1~4 でつけプログラム *accuracy* で処理する、というものである。

1bd をプログラム *accuracy* で処理した結果を示す。

Recognition accuracy:

1:	1 entries	0.00 %
2:	1 entries	100.00 %
3:	1 entries	0.00 %
4:	1 entries	0.00 %

Total accuracy: 4 entries 25.00 %

これは、1bd のクラスは2であることを示していると考えられる。つまり SOM では2の近傍に表示されたただけだが、LVQ では明確に2と判断している。

プログラム *accuracy* はこのように精度を%単位で数値的に表示する。SOM でのラベル付けが疑わしいとして再検討した全てのデータが、*accuracy* で処理するために付した4つのクラスのいずれか1つにおいて100%の認識精度で評価された。また、プログラム *classify* による解析結果は *accuracy* の結果と同じであった。

表 5.8 に *olvq1* による解析結果をまとめて示す。

表- 5.8 *olvq1* による解析結果

データ名	元の評価	<i>olvq1</i> による評価
1bd	1	2
1cc	1	2
1ct	1	2
1cv	1	2
1dm	1	2
1dp	1	2
1el	1	2
2bl	2	2
2cr	2	2
2dd	2	2
2eb	2	3
2ef	2	2
3aa	3	3
3ap	3	2
3ax	3	2
3be	3	3
3bf	3	3
3ce	3	3
3ck	3	2
3cp	3	3
3cs	3	3
3dj	3	3
3dq	3	3
3dt	3	3
3du	3	3
3ej	3	3
3ek	3	2
4ac	4	3
4ag	4	3
4an	4	2

入力データにはクラスは違うが、データ項目が全く同じものが二組あった。3dj と 2eb, 3ck と 2cr である。表 5.8 をみると 3dj は 3 であり 2eb は 3, 3ck は 2 であり 2cr は 2 という評価がくだされている。よって、入力データのうち 2eb と 3ck は削除した。なぜなら、全く同じ入力データが存在する場合、そのデータを他のデータより強調した学習をしてしまうからである。

元の評価と *olvg1* による評価が異なった場合にはその入力データを書き換え、先ほどのパラメータ設定 ($NOC=80$, 学習回数=4000) で最学習した。この学習で得られたコードブックベクトルと書き換えた入力データ (すなわち学習に用いた入力データ) を用いてプログラム *accuracy* の処理した結果を以下に示す。

Recognition accuracy:

2:	85 entries	97.65 %
3:	27 entries	96.30 %
1:	5 entries	100.00 %

Total accuracy: 117 entries 97.44 %

鋼橋のように全てのクラスで認識精度が 100% とまではいかなかったものの、高い精度であるといえる。

ただし 3 つあったクラス 4 の入力データは LVQ により全て評価が変わったため、クラス 4 は消滅してしまった。クラス 4 は安全な橋梁を意味しているのでこのデータを用い橋梁損傷度を自動判定するエキスパートシステムを構築した場合、安全な橋梁を危険と判断することはあっても、危険な橋梁を安全と判断することはないといえる。

また $NOC=80$, 学習回数を 3500 回としたときの出力結果を下に示す。

Recognition accuracy:

2:	85 entries	100.00 %
3:	27 entries	96.30 %
1:	5 entries	100.00 %

Total accuracy: 117 entries 99.15 %

学習回数が 40000 回の時に比べクラス 2 の精度が向上している。学習回数が 40000 回よりも小さい 3500 回で認識精度が向上したのは、2eb と 3ck を削除したため入力データ数が減少したためと思われる。

最終的な入力データの決定

ここで第 5.2.3 項での結果を反映した入力データを用い、再び SOM で学習した結果を図 5.25 に示す。各パラメータはマップサイズが 25×25 、学習回数が 400000 回、 α が 0.9、 radius が 13 である。その結果量子化エラーは 0.000694 であり、全てのデータが表示されている。

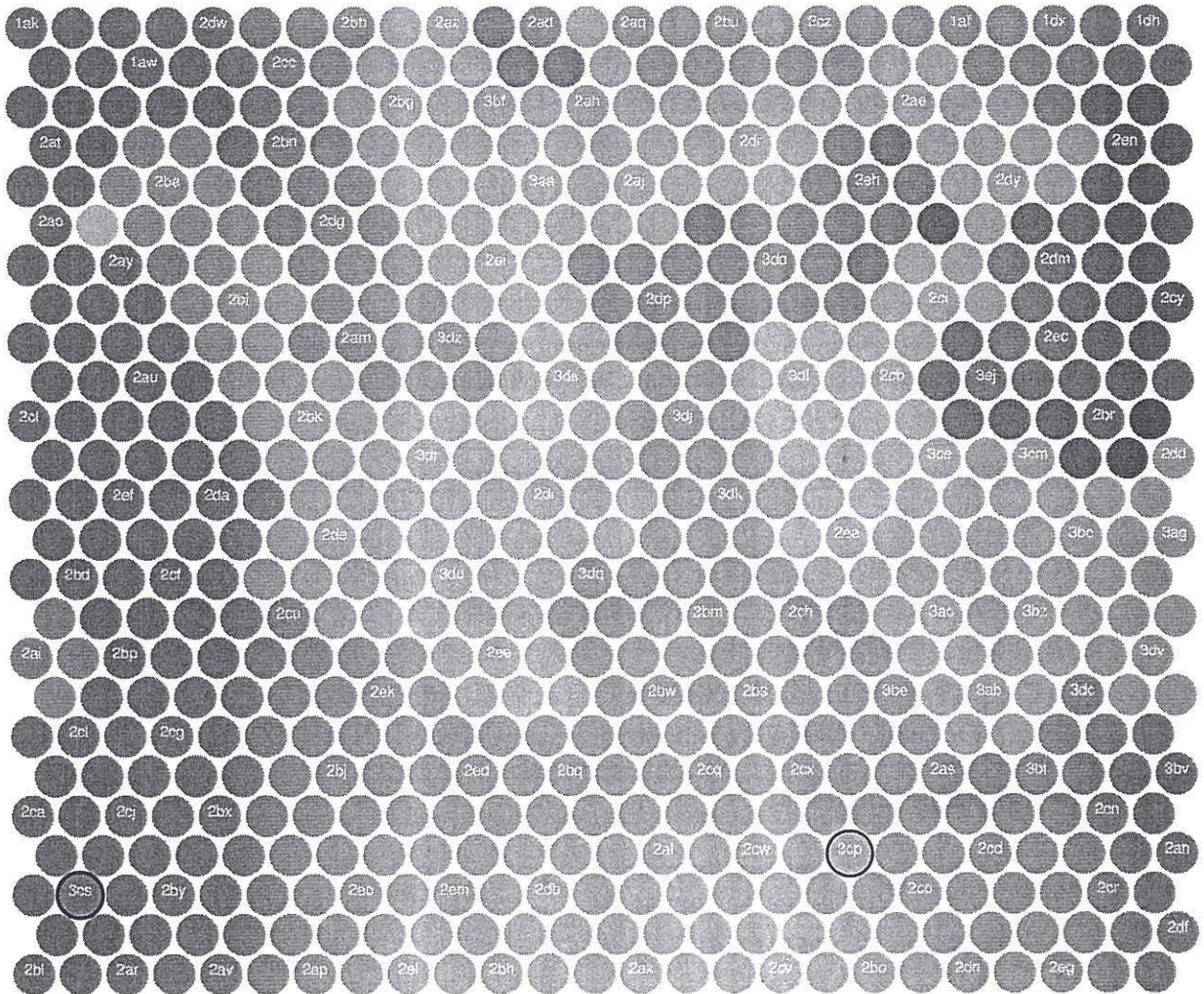


図- 5.25 入力データ修正後の SOM の結果

図 5.15 を見ると 3cp と 3cs が疑わしく思える。しかしこの二つは第 5.2.3 項で既に再評価済みのデータであり、再評価の結果として評価の変わらなかったものである。また第 5.2.3 項で総合的な認識精度は 99.15 % とかなり高かった。よって、コンクリート橋の橋梁点検データの見直しはこれで終える。

5.2.4 考察

SOM/LVQ において見直されたデータに関する考察

ここで、SOM/LVQ で評価が変わったデータに関して検証する。図 5.23 から、データ 1cc とその近傍に配置されているデータ 2dw, 2cu, 2ef, 2bp, 2co を抜き出す。1cc は SOM において周囲のデータのラベルが “2” であったデータで、LVQ による解析の結果 “2” と評価されたものである。そのデータ項目の内容を表 5.9 に示す。

表- 5.9 評価が変わったデータとその近傍のデータ

1	1	1	1	2	2	2	1	1	1	2	2	2	1cc
1	1	1	1	2	2	3	1	1	1	2	2	2	2dw
1	1	1	1	2	2	2	1	1	1	1	1	2	2cu
1	1	1	1	2	3	2	1	1	1	1	1	3	2ef
1	1	1	1	2	2	1	1	1	1	1	2	2	2bp
1	1	1	1	1	1	1	1	1	1	1	2	2	2co

表 5.9 を見ると、確かに 1cc は周囲のデータと類似していることがみてとれる。

入力データに関する考察

入力データを見直す前に、そのデータをプログラム *accuracy* にかけた内容を再び示す。

鋼橋

Recognition accuracy:

2:	37 entries	100.00 %
3:	10 entries	70.00 %
1:	8 entries	37.50 %

Total accuracy: 55 entries 85.45 %

コンクリート橋

Recognition accuracy:

2:	75 entries	98.67 %
3:	29 entries	86.21 %
1:	12 entries	41.67 %
4:	3 entries	0.00 %

Total accuracy: 119 entries 87.39 %

ここで注目する点は各クラスの認識精度ではなく、各クラスにエントリーされているデータ数である。鋼橋、コンクリート橋ともにクラス2のデータ数が他のクラスに比べて多い。クラス2は、“できるだけ早期に対策を講じたほうがよい” 橋梁を意味している。

クラス2の数が多いことはLVQの学習にも影響を及ぼしていると考えられ、表 5.4,5.8を見るとLVQで評価の変わったデータの多くがクラス2への変更となっている。

本研究では入力データを見直すことで知識の洗練化を行ったが、知識をさらに洗練させ知識ベースの信頼性をあげるためには、クラス2の数を他のクラスの数とバランスをとらせる必要があると考えられる。または、クラス2をさらにいくつかのクラスに分ける処理が必要であると考えられる。つまりクラス2の意味する、“できるだけ早期に対策を講じたほうがよい” の“できるだけ” という表現にいくつかの段階があるように見える。

未知データの解析

ここでLVQを使い、未知データの解析ができることを示す。つまりLVQが入力データの洗練化だけではなく、エキスパートシステムの推論機構として使用できる可能性を示す。

未知データを含んだデータをSOMで処理し、その結果得られたマップの近傍から未知データのクラスを推測することも可能である。しかしその場合は“可能性が強い” という程度であり、断言することはできない。

鋼橋を例にして話を進めることにする。まず以下の点に注意して、鋼橋のデータからいくつかのデータを抽出する。

- SOM/LVQ で評価の変わったデータは除外する (第 5.2.2 項参照)。
- 各クラスのデータ数に大きな差がないようにする。

評価の変わったデータを削除すると、

クラス		データ数
1	⇒	3
2	⇒	37
3	⇒	7

となる。クラス 1 はその全てを抽出する。各クラスのデータ数に大きな差がないように、クラス 2、クラス 3 のデータからはそれぞれ 5 つのデータを抽出する。このとき図 5.13 の SOM の結果を見て、近傍集合の中から 1 つのデータを選び出す。この方法により、類似したパターンの中から 1 つのデータを抽出することができる。このようにして抽出したデータは、各クラスを代表しているデータと考えることができる。

次にこの抽出したデータ群をコードブックベクトルにする。LVQ_PAK のプログラム *eveninit* によってコードブックベクトルを初期化すると、入力データでデータ数の多いクラスのコードブックベクトルがどうしても多くなってしまう。今回は未知データの解析に LVQ を用いるため、ある一つのクラスへ偏りが出ないようにこのような手法を用いる。また LVQ の学習はあるクラスを代表するコードブックベクトルを作り出すことであり、先に SOM で各クラスを代表していると考えられるデータを抽出することは良い戦略であるといえる。

第 5.2.2 項での LVQ による解析結果を反映したデータ (すなわち、LVQ によって橋梁の全体的な損傷度が変わった場合はそのデータを書き換えたもの) を入力データにして学習をさせる。このことにより抽出しなかったデータも考慮に入れることができる。またデータ抽出時の人間の曖昧さを解消できる。

後はこの学習の結果得られたコードブックベクトルを使い、解析したいデータをプログラム *classify* で処理すればよい。

全ての部材の損傷が大きいと判定されたデータを用意し *classify* で処理した。その結果はクラス 1 であり、十分うなずけるものとなった¹²。

¹²クラス 1 は、“放置すると近い将来危険な状態になることが予想されるので、直ちに対策を講ずる必要がある。”ということの意味している。

入力データに関する検討

コンクリート橋には、鋼橋にはある塗装の項目が存在しない。そのため本研究では別々のデータとして扱ったが、この2種類のデータを同様に扱えるかどうかについて検討する。

コンクリート橋のデータと鋼橋のデータを同等に扱うためには、データの次元数(項目数)を同じにする必要がある。つまり、コンクリート橋のデータに塗装データ項目を追加する必要がある。ここでは次の2種類の方法でデータ項目を追加してみる。

- 塗装項目については問題がないとする。
- 塗装項目のデータを欠損データとして扱う。

問題がないということは、そのデータ項目を“1”にするということである。またSOM/LVQは“x”とすることで、そのデータを欠損データとして取り扱う。このようにしたコンクリート橋のデータを第5.2.4項で作成した、簡易的な未知データ判定システムで処理する。その結果を表5.10に示す。この表は見直し前の入力データと、第5.2.3項でSOM/LVQによって見直しを行った後のデータの二つとの一致率を%単位で表わしている。

表- 5.10 コンクリート橋に塗装項目を付け足したときの一致率

データ	塗装項目	
	問題無し(1)	欠損(x)
見直し前	68.07%	72.27%
見直し後	82.05%	82.91%

表5.10を見ると、まず見直し後のデータのほうが一致率が高いことが分かる。これはSOM/LVQによる見直しが、有効であったことを示していると考えられる。次に塗装項目を欠損扱いしたほうが一致率が高い。これは次の2点を示している。

- 塗装項目を欠損扱いしたほうがよい。
- LVQは欠損項目を含むデータの処理にも有効である。

第5.2.4項で作成した簡易的な未知データ判定システムにより処理したわけであるが、この一致率の高さからLVQが未知データの解析にも有効であると考えられる。

最後に(これが当初の目的なのだが)、一致率が高いので鋼橋・コンクリート橋のデータを同等に扱うことは可能であると考えられる。

参考文献

- 1) 辻岡信也：橋梁維持管理支援に関する研究，金沢大学学士学位論文，1996.2.
- 2) T.Kohonen 著，高德・岸田・藤村 訳：自己組織化マップ，シュプリンガー・フェアラーク東京，1996.6.
- 3) T.Kohonen, J.Hynninen, J.Kangas, and J.Laaksonen , *SOM_PAK : The Self-Organizing Map Program Package*, Lab.of Computer and Information Science, Helsink Univ. of Tech., Report A31, 1996.1.
- 4) T.Kohonen, J.Hynninen, J.Kangas, J.Laaksonen, and K.Torkkola , *LVQ_PAK : The Learning Vector Quantization Program Package*, Lab.of Computer and Information Science, Helsink Univ. of Tech., Report A30, 1996.1.
- 5) T.Kohonen 著，中谷和夫 監訳：自己組織化と連想記憶，シュプリンガー・フェアラーク東京，1995.10.
- 6) 石川県:昭和 57 年度～昭和 63 年度 橋梁診断調査総括表集, 1990.
- 7) 桐谷滋, ニューロンネットグループ 著, 飯沼一元 編, 中野馨 監修：入門と実習 ニューロコンピュータ, 技術評論社, 1989.9.
- 8) 萩原将文：ニューロ・ファジィ・遺伝的アルゴリズム，産業図書，1994.9.
- 9) J.Dayhoff 著，桂井浩 訳：ニューラルネットワークアーキテクチャ入門，森北出版，1992.4.
- 10) 松岡清利 編：ニューロコンピューティング，朝倉書店，1992.7.
- 11) 船橋誠壽：ニューロコンピューティング入門，オーム社，1992.6.
- 12) 鈴木力 編：情報・知識 imidas1998, pp.222,888-899, 集英社，1998.1.
- 13) 赤坂繁 編：情報・知識 imidas1995, pp.972-983,993-1001,1025-1026, 集英社，1995.1.