多様さに対応するユーザーインターフェースの研究
——Life Cycle Assessmentにおける最適化問題の一試み——

白　砂　洋志夫

A Study of User-interface for Multifariousness in the Information Domain.

Yoshio Shirasuna

An optimal evaluation method has been studied for life cycle assessment (LCA) or abridged life cycle assessment (ALCA). Many factors on environmental evaluation will be far more complicated. Then Hopfield's neural network model applied to matrix of environmental factors in ALCA. The european community ecolabel matrix is used in this optimal problem system. The results of evaluation used this system are indicated by numerical ranking. This optimal problem system will be used effectively on the environmental evaluation in ALCA.

1．緒言

現在、インターネットの発達により個人のコンピュータであってもネットワークに接続すれば世界中のさまざまなデータベースを利用し、必要とする情報を比較的簡単に入手できる環境ができている。それに比べ、手に入った情報をどのように加工し、意とする表現を提示するための広義のユーザーインターフェースの開発は遅れている。さまざまな情報ドメインに対応した、情報を有効に活用するためのユーザーインターフェースの開発［伊藤 94］が望まれている。

多様で具体的な問題の一つに地球環境問題があり、環境への負荷を評価するLife Cycle Assessment（以下ＬＣＡと略す）手法が注目されている。従来、製造業のビジネスにおいて、製品のライフサイクルでは主に発案・設計の第一段階から出荷までを扱ってきた。ここには研究、マーケッティング、サービスも製品を支えるために含まれている。ビジネスはコストと利益を考慮しつつ、需要に見合う質の製品を生み出すことに視点が置かれて策定されてきた。ここでは製品の廃棄、放出などに伴う環境への考慮はほとんどなされていない。これとは対照的に、環境イベントリー分析やインパクト分析は、製品に関わる「もの」の物理的、化学的なシステムに依存する。LCAでは「もの」やエネルギーの流れ、原材料の入手から残留物の最終廃棄までの変換を追跡する。これは本来、一般消費材のライフ

サイクルにおける環境への負荷を評価する手法として開発されものである。しかし、最近ではＬＣＡの概念は大型のプラントなどの建造物や生産プロセスにも適応されており［アメリカ環境保護庁96］、その概念は拡張されて多方面で利用されている。地球上の膨大な資源とエネルギーを使用した大量生産・大量消費・大量廃棄の産業社会の質的な変換、すなわち将来にわたって持続可能な社会を目指して、人類全体の環境への負荷活動をいかにトータルとして抑制していくかは地球的規模で急務な課題である。このように「もの」の製造システムの運用にＬＣＡの概念を適用して、種々の「もの」を設計し、作り上げ、維持管理し、リサイクルしていく中で、そうした製品の履歴が広義の環境に与える影響を最小限とすることを目的とし、ＬＣＡをシステムを根幹とした産業エコロジー［Graedel 94］、［T.E.グレーデル96］と称する新しい分野が開発されている。

この分野においても情報とその処理は重要で、さまざまな産業連関表などから有効なデータを取り込み、ＬＣＡシステムの最適な運用を示唆するための手法として「Sima-Pro」、「ブースッテド・モデル」［未踏科学技術協会 95］などが提案されている。しかし、環境に関わる情報は多様な影響因子により構成されているためにＬＣＡ環境影響評価を完全に行うには時間とコストがかかりすぎる。

そこで、一般的にはＬＣＡの概念を取り入れて、それぞれの固有なニーズや制約条件により適合する実際的な形式を編み出し実施している。このようなアプローチの形式をまとめて「簡便ＬＣＡ」（ＡＬＣＡ：abridged ＬＣＡ）［T.E.グレーデル 96］と呼び区別している。ＡＬＣＡの評価は基本的にはＬＣＡの評価と同じであり、ライフステージと物質などの入出力値のマトリックスが用いられる。本研究ではこのＡＬＣＡでの環境評価に用いられるマトリクスに注目して、ニューラルネットワークで用いる最適化問題の解法を適応し、パーソナルコンピュータを用いてＡＬＣＡにおける環境影響評価を簡便に行うためのユーザーインターフェースを提案する。

2．ＡＬＣＡ法におけるマトリックス

典型的なマトリクスの各要素についてすべての物質フローや環境影響量を決定する通常のLCAとは異なり、ＡＬＣＡでは、最も重要

表2.1：モデルとした欧州共同体エコラベル・マトリクス

重要な環境特性	製品のライフサイクルステージ				
	原料	生産	流通	利用	廃棄
廃棄物の発生量	*f1,1*				*fm,1*
土壌汚染及び劣化					
水質汚濁					
大気汚染					
騒音					
エネルギー消費量					
天然資源の使用量					
生態系への影響	*f1,n*				*fm,n*

と考えられる要素だけを考慮し、さらに取り上げた要素で最も影響の大きいもののみを考えている。

プロセスは、数学的にはマトリックス操作の１演算として表すことができる。表2.1のマトリクスを考えてみる。今、マトリックス要素 fm,nがイベントリー分析で埋まったとすると、ALCAの第１段階のマトリクスができあがり、これを F と呼ぶことにする。また、別の表が sm,n なる環境影響データで埋まっているとすれば、S と呼ぶＬＣＡの第２段階マトリクスを得たことになる。ある単一の環境特性ｎに対するＬＣＡ評価は、次式で与えられる。

$$L_n = \sum_{1}^{m} f_{mn} \times s_{m,n} \qquad (2.1)$$

同様にして、ある単一のライフサイクルステージｍに対するＬＣＡ評価は、次式で与えられる。

$$L_m = \sum_{1}^{n} f_{m,n} \times s_{m,n} \qquad (2.2)$$

したがって、全体のＬＣＡ評価値は次のようになる。

$$L = \sum_{1}^{m}\sum_{1}^{n} f_{m,n} \times s_{m,n} \qquad (2.3)$$

ここでマトリクス F のいくつかの要素はゼロであることがある。これは次の二つの状況のいずれかのときに起る。つまり、ある製品の流通の段階では土壌汚染及び劣化は考えられないというような場合のように、イベントリー自身がゼロといった状況とか、ＡＬＣＡでしばしば経験するように、イベントリー値が重要でないと判断される状況である。また、マトリクス S でゼロを生じるのは、製品やプロセスについて何の環境影響も想定されないときである。エキスパート・システムのＬＣＡでは、マトリクス要素がすべて定量化されている場合が多い。各要素の数値は、問題があるなしといった判断システムの場合のようにバイナリーか、１から５の数値で影響の度合いをランク付けするシステムと同様にいくつかの数字の１つかである。この場合、マトリクス演算がそのまま適応できる。ＡＬＣＡのマトリクス演算の結果は設計案やその代替案に対して評点付けできることになる。

3．ＬＣＡにおけるユーザーインターフェースの意味

人間は視聴覚機能を持ち、これを通じて外界から入ってくる種々の情報(パターン情報といわれている)を識別して、その意味を理解して外界の状況を的確に把握することが出来る。また自分が望んでいる目的を達成するために、これらの外界の状況に照らし合わせ問題を解決するための思考すなわち推論を行う。今日では自然言語によってかなり自由な形式で表現されている情報の意味を理解することができる。さらに、これらの情報認識、問題解決、自然言語理解、パターン理解などの活動を通じて得た過去の経験や知識を記憶し、後にそれらの経験･知識を有効に活用して諸活動の最適化･経済化を計っていくという学習の能力を備えている。これら人間のもつ知的情報処理の本質的な機能は何か、現在のコンピュータのハードウエア、ソフトウエアの技術でどこまで実現できるかなどについて、ある程度の知見は得られている［江原 97 b］。たとえば、知識情報処理には知識をデー

タとして扱うなかで、前述した観点に立って、情報処理システムに数値データの処理回路のみでなく、ユーザーの経験や思考判断が取り入れることのできるサブシステム［伊藤 94］を附加することができれば、処理回路の軽減と有効な簡略が可能となると考えられる。そのなかでも対象とするデータ群が多様な因子を持つ問題処理システムに、このようなユーザーインターフェースは効果的であると考えられる。このような問題処理システムにおいて最適解問題としての扱いは重要なものの一つである。近年、脳の情報システムの解明から生まれたニューラルネット［江原 97］のダイナミックスに物理的な理論を導入して、最適解を求めることが提案されている。その一つにホップフィールドモデル［中野 95］、［廣田 96］がある。

4. ホップフィールド・モデルの概略と最適解問題への応用

このモデルは1982年に、Hopfield［中野 95］により提唱された。ホップフィールド・モデルにおけるネットワークはニューロンと呼ばれるユニット間の結合が対称関係の相互結合型ネットワークである。このモデルを図4.1に示す。各ユニットは、他のユニットから入力を受け、その入力の強さによってニューロン間の結合関係を固定すると、ニューロンの状態から定まるある関数が、時間が経つに連れて減少し、最終的には極小値に収束する。

その動作を N 個のユニットを持つネットワークで表現すると式(4.1) (4.2)となる。

ホップフィールドはこのネットワークにお

$$u_i(t) = w_{i1}(t) + w_{i2}(t) + \cdots\cdots + w_{iN}(t) + h_i \tag{4.1}$$

$$x(t+1) = \begin{cases} 1 & (u_1(t)) > 0 \\ x(t) & (u_i(t) = 0) \\ 0 & (u_i(t) < 0 \end{cases} \tag{4.2}$$

ここでNはユニットの数であり、各記号は次のとうりである。

$x_i(t)$ ：ユニット i の出力

$u_i(t)$ ：ユニット i への出力

w_{ij} ：ユニット j からユニット i への結合の強さ

$-h_i$ ：ユニット i の閾値

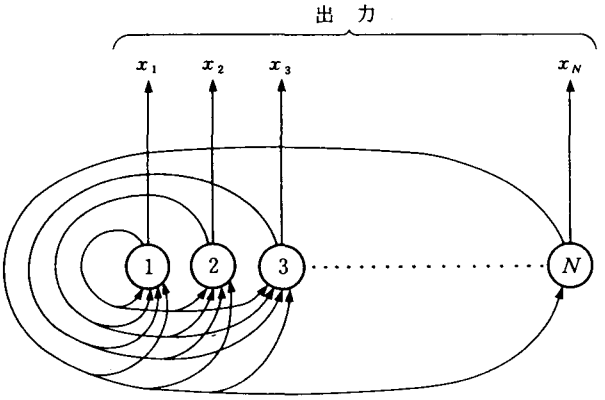

図4.1：ホップフィールド・モデル

いて力学との類推からエネルギーと呼ばれる量Eを次式のように定義し、この関数をエネルギー関数と呼んでいる。

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} x_i x_j - \sum_{i=1}^{N} h_i x_i \qquad (4.3)$$

ホップフィールド・モデルのネットワークでは、個々のユニットのミクロな動きによってその状態が変わっていくが、全体をマクロに見た場合、エネルギーという量が定義できて、その量がネットワークの動作に従って減少していくというものである。

ここでエネルギー量とネットワークの動作との関係に注目すると、このネットワークで同時刻にはN個のユニットのうち、一つのユニット出力しか変化しない。第 i 番目のユニットの出力 Xi を0の場合と1の場合としたときのエネルギーの差を計算すると（他のユニットの出力はすべて固定する）、次式が求まる。

$$\Delta E_i = E(x_i = 0) - E(x_i = 1)$$
$$= w_i x_1 + w_{i2} x_2 + \cdots\cdots + w_{iN} x_N + h_i \qquad (4.4)$$

これは前述した式（4.1）と同じである。したがって式（4.3）に従うユニットの出力変化は、以下のように理解できる。すなわち第 i 番目の出力を0としたときのエネルギー値と1とした時のエネルギー値が異なれば、エネルギーを小さくする方を出力として採用し、同じならば出力はもとのままとする。すなわち、ネットワークはエネルギーを小さくするように状態を遷移することになる。このような遷移状態を繰り返すうちに、どのユニットの出力を変えてもエネルギーは小さくならない平衡状態に到達し、定常となる。そこで、このモデルを最適化問題へ応用することを考える。最適化問題は、目的関数を設定してそれを最小化するような変数の組合わせを求めるものである。一方、エネルギー関数は、ネットワークの結合強度w ijを適宜設定すると、任意の2次関数を表すことができるので、目的関数を2次関数により表すことができれば、これと一致するエネルギーを持つネットワークが設計できる。このとき、最適化問題の変数は、ネットワークの出力 $X=(X1, X2, X3 \cdots Xi)$ に対応させる。これにより、ネットワークを動作させ、目的関数の最少化を計る。以上が最小化問題を解くホップフィールド・モデル

の概要である。

5．ユニット間の結合が対称関係の相互結合型ネットワーク

ホップフィールド・モデルを最適解アルゴリズムに応用して、ＬＣＡに関連する情報をデータベースから取り出し，その情報をもとに、多様な条件を考慮して、地球環境への負荷を軽減するリサイクル、リユース、廃棄などの「もの」のフロー行程の探査に役立つユーザーインターフェースの構築を試みる。この処理方法の基本に人工知能の領域で最適な解を探索する方法として開発された、ホップフィールド・モデルの利用を検討した。本論文では多様な環境影響因子に関係する情報を活用し、ＡＬＣＡ環境影響因子の評価にユーザーのもつ知識や経験を部分的に生かすシステムを構築することを目的とした。

前述の性質を逆に利用すると、最小値を求めたい関数をエネルギー関数として持つネットワークを構成して、適当な初期状態から始めて、収束した状態を知れば極小値を求めることが可能である。最適化問題は目的関数を設定してそれを最小化するような変数の組合わせを求めることになる。

ここで本研究において対象とした変数の組み合わせは表5.2に示したＡＬＣＡ法に用いられる欧州共同体エコラベル・マトリックスを使用した。表2.1の個々のイベントリーをユニットとしたネットワークをホップフィールド・モデルによる最適化問題として扱うと、式(2.1),(2.2),(2.3)のＬを最小とするような一般特性ｎのマトリクス出力 Ln とある単一のライフステージのマトリクス出力 Lm をホップフィールド・モデルのそれぞれのユニットごとに式(4.2)を対応させて式(2.1),(2.2)を最適化問題として計算する。全体の出力を環境影響評価値とする。このフローチャートを図5.1に示す。

ホップフィールド・モデルを利用したＡＬＣＡ環境影響因子の評価プログラムリストを付録として添付した。

従来の徹底した数値解析一辺倒では膨大な数値処理が必要となり誰でもが簡単にパソコンで扱うというのは困難である。そこでＡＬＣＡにおけるユーザーインターフェースでは、人間の思考をモデルとした最適解を求める手法をユーザーインターフェースに取り入れることを検討した。これらのユーザーインターフェースのプログラミングは汎用性のあるＣ言語を用いた。多様な環境影響因子に関係する情報を活用し、ＡＬＣＡ環境影響因子の評価にコンピュータが専門でないものにとっても、手を出しやすいとすることが普及のために必要である。さらに最近パーソナルコンピュータのＯＳとしてWindowsが多く使用されている。VisualCなどのＣ言語系のソフトも普及していることも考慮した。

6．計算結果と考察

本研究ではLife Cycle Assessment評価システムにニューラルネットワークにおける最適化問題の解法を応用したユーザーインターフェースを付加し、欧州共同体で用いられているエコラベルを対象とし、そのマトリックスをニューラルネットワークで使われているユニットの概念を取込んだＡＬＣＡ環境影響評価のためのプログラムを用いて計算を行った。その結果を表6.1に示す。ホップフィールド・モデル最適解アルゴリズムの問題点［中

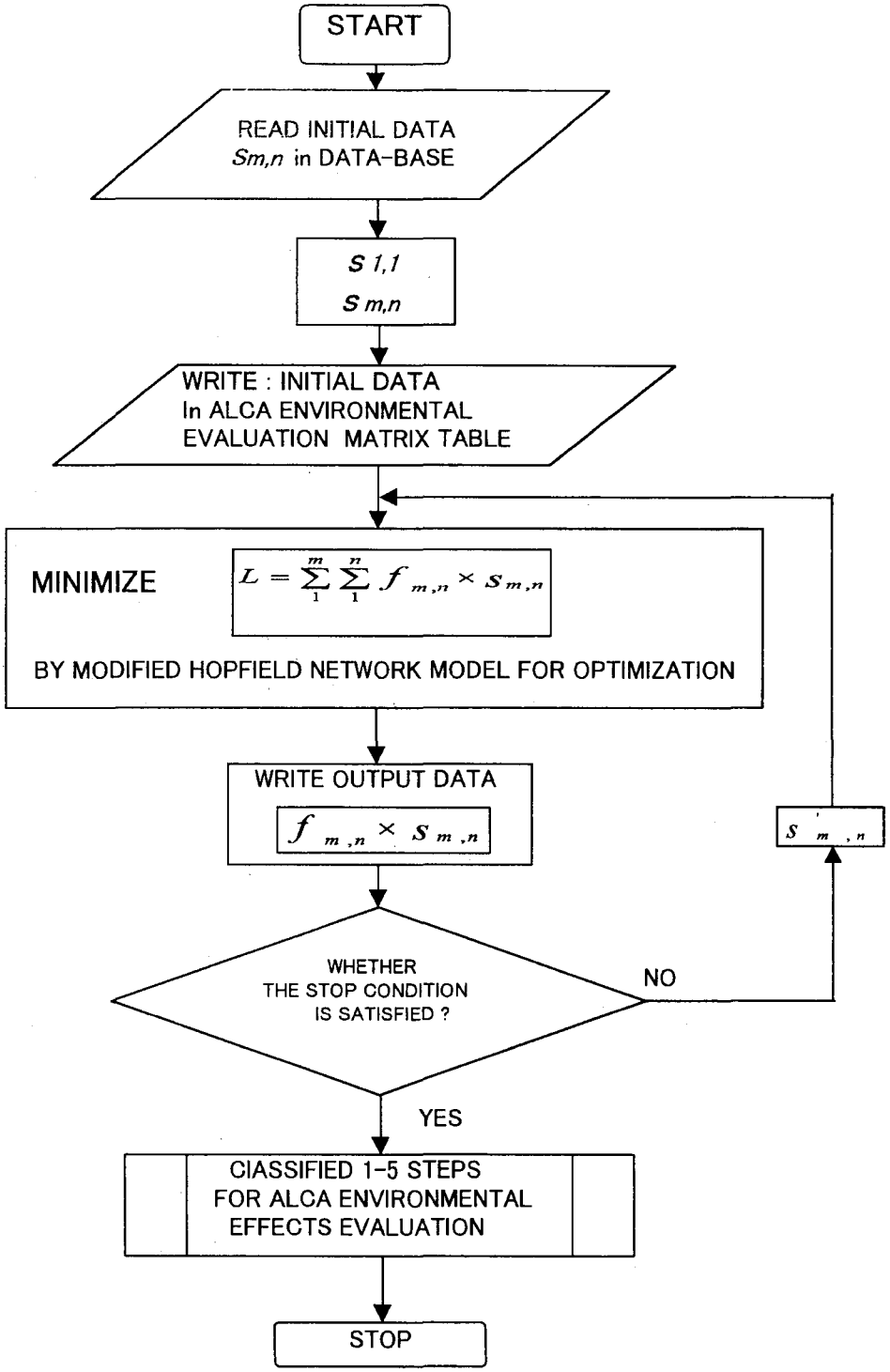

図5.1：ＡＬＣＡ環境影響因子評価システム・フローチャート

表6.1：ＡＬＣＡ環境評価プログラム出力例

重要な環境特性	製品のライフサイクルステージ				
	原料	生産	流通	利用	廃棄
廃棄物の発生量	3	2	1	1	4
土壌汚染及び劣化	2				
水質汚濁	3				
大気汚染	3				
騒音	2				
エネルギー消費量	4				
天然資源の使用量	4				
生態系への影響	3				

影響度：5：決定的、4：主要、3：中程度、2：多少、1：微

野 95]、[廣田 96] である最小値探索の能力が不十分なことはこの結果にはほとんど影響していないと判断できる。

7．多様な要因を考慮した視覚的表現法の利用

さらにユーザーインターフェースで処理した多様な条件での最適解の情報を視覚的に表現するためレーダチャート用いた。その結果の一例を図7.1、図7.2に示す。この表示は産業エコロジーを具体的に広めるためにＩＳＯ１４０００シリーズなどの施行により産業界はもとより自治体などの現場に関わる多くの人たちが具体的な数値をもとにして環境影響をより具体的に評価をすることに利用されている。このような観点から、評価がレーダチャートのような視覚で表現されることは有益である。

8．結言

有効なＡＬＣＡ環境影響評価法を構築するには、各ユニットの出力を現実的なＡＬＣＡ環境影響評価に適合するレベルに設定する必要がある。たとえば、ISO140001の評価基準に対応するレベル、または、具体的なプラントにおける制御系に直接フィードバックできる数値など、出力を利用する側に適応した出力レベルの選択が必要となる。本研究において検討したホップフィールド・モデル最適解アルゴリズムを取込んだＡＬＣＡ環境影響評価のためのシステムでは、ブラックボックス的な部分が多分にあり、人間が最終的に責任をもって評価を下し、意思決定をする評価システムとするには以下のことが満たされる必要がある。

① 結果にいたるプロセスとルールがある程度具体的に理解できること。
② プロセスやルールをユーザーがメンナンスできること。
③ 実行可能な案が業務の言葉で表現されていること。
④ 関連するすべての部署の情報が使われていること。

このような意味では、ニューラルネットワークの中でも知識、経験、感覚を具体的に評価に取込みやすいニューロ・ファジーの応用

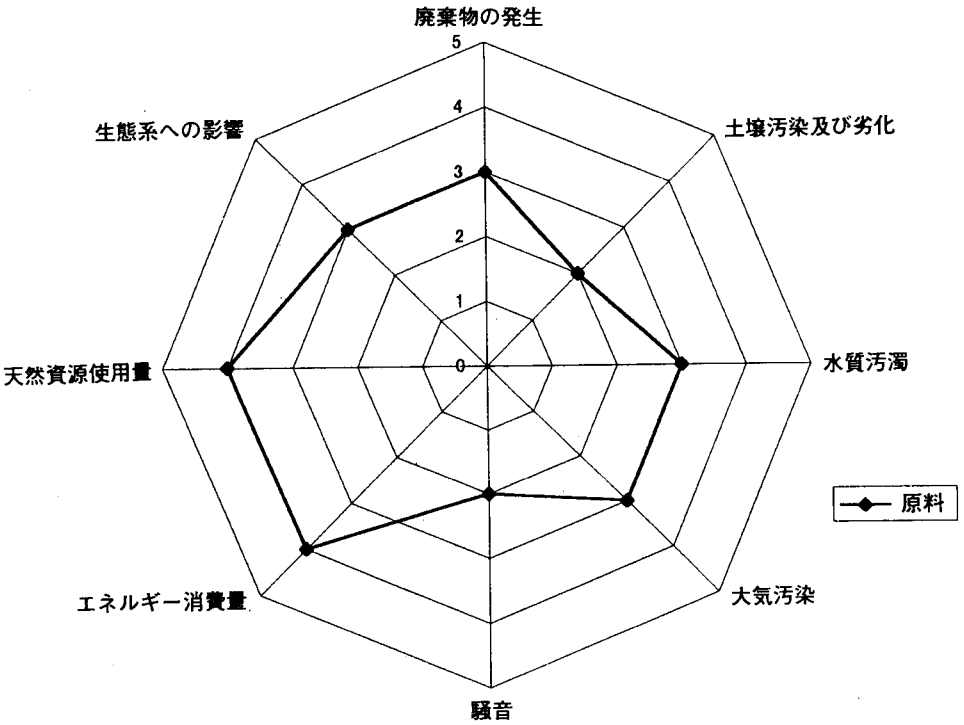

図 7.1：原料イベントリの環境影響因子による評価

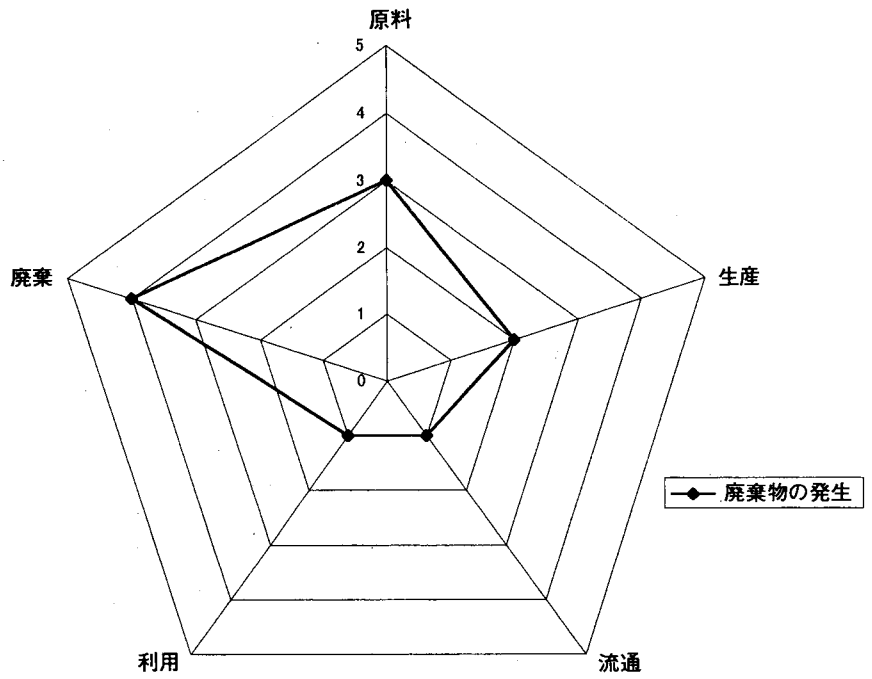

図 7.2：各イベントリにおける廃棄物発生評価

が有効と考えられ、次の研究課題としたい。

参考文献

[Graedel 94] Graedel, T. E. : Industrial ecology : Definition and implementation, in Industrial Ecology and Global Change, R. Socolow, C. Andrews, F. Berkhout and V. Thomas, eds. Cambridge, UK : Cambridge University Press, 1994.

[伊藤 94] 伊藤潔、田村恭久、杵島修三：Triadic Domain Modelに基づくシステムの分析・設計、日本ソフトウェア科学会「ソフトウェア工学の基礎」論文集、Vol. 1, December 1994, pp. 81-88

[中野 95] 中野馨編著知能システム研究会：「Cでつくる脳の情報システム」、近代科学社、1995、pp. 97-124

[未踏科学技術協会 95](社) 未踏科学技術協会・エコマテリアル研究編：「ＬＣＡのすべて」、工業調査会、1995, pp. 8-13

[T. E. グレーデル 96] T. E. グレーデル / B. R. アレンビー著 後藤典弘訳 「産業エコロジー」、プレンティスホール・トッパン、1996, pp. 181-185

[アメリカ環境保護庁 96] アメリカ環境保護庁編、梅田富雄訳：「環境にやさしい設計ガイド」、工業調査会、1996, pp. 25-44

[廣田 96] 廣田薫編著：「知能工学概論」、昭晃堂、1996, pp. 84-95

[江原 97] 江原淳、戸田慎一、土田昭司、梅本亨：「やさしい情報処理」、有斐閣、1997, pp. 224-230

[江原 97b] 江原淳、戸田慎一、土田昭司、梅本亨：「やさしい情報処理」、有斐閣、1997, pp. 250-254

付録：最適解アルゴリズムを組み入れたユーザーインターフェース・プログラム

```
/*************************************************/
/*                                               */
/*  ホップフィールド最適解を適応した ALCA        */
/*                                               */
/*************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <graphics.h>
#include "tsp.h"

#define N       7
#define M       N*N
#define DT      0.01

double x[N][N];
double u[N][N];
double w[N][N][N][N],h[N][N];
double d[N][N];
double T;
double tau=2.0,u0=0.2,umax=1.0;
double alpha=3.0,beta=6.0;
int    tx[N],ty[N];
int    schedule = 0;

void dis_init1()
{
    int  graphdriver = DETECT,graphmode;

    clrscr();
    initgraph(&graphdriver,&graphmode,getenv("TC"));
    cleardevice();
    rectangle(420,20,620,390);
}

void dis_init2()
{
    clrscr();
    setfillstyle(SOLID_FILL,BLACK);
    bar(19,19,401,391);
    setcolor(WHITE);
    rectangle(20,20,400,390);
}
```

```
void dis_init3()
{
    clrscr();
    setfillstyle(SOLID_FILL, BLACK);
    bar(19, 19, 401, 391);
    setcolor(WHITE);
    rectangle(20, 20, 400, 300);
    rectangle(20, 310, 400, 390);
}

void menu_disp(x, y)
  int  x, y;
{
    if(y == 1) {
        setcolor(BLACK);
        setfillstyle(SOLID_FILL, LIGHTGREEN);
        bar(50, 37+x*30, 380, 57+x*30);
    }
    else if(y == 0) {
        setcolor(WHITE);
        setfillstyle(SOLID_FILL, BLACK);
        bar(50, 37+x*30, 380, 57+x*30);
    }
    switch(x) {
        case 1:
            outtextxy(50, 70, "1. ホップフィールド離散モデルの実行");
            break;
        case 2:
            outtextxy(50, 100, "2. ホップフィールド連続モデルの実行");
            break;
        case 4:
            outtextxy(50, 160, "4. LCA ユニットの再配置");
            break;
        case 5:
            outtextxy(50, 190, "5. パラメータ(alpha, beta, tau, u0)の変更");
            break;
        case 6:
            outtextxy(50, 220, "6. 終了");
            break;
    }

}

void menu()
{
```

```
    int  i;

    dis_init2();
    setfillstyle(SOLID_FILL, BLACK);
    bar(21, 21, 399, 299);

    outtextxy(100, 35, "<メニュー>");
    for(i=1;i<=6;i++) menu_disp(i, 0);

    menu_disp(1, 1);
}

void position()
{
    int  i, j;
    int  x, y;
    int  check = 0;
    int  distance;

    for(i=0;i<N;i++) {
        while(1) {
            x = rand()%180;
            y = rand()%300;
            for(j=0;j<i;j++) {
                distance = abs(tx[j]-x)+abs(ty[j]-y);
                if(distance < 75) {
                    check = 1;
                    break;
                }
            }
            if(check == 1) {
                check = 0;
                continue;
            }
            else break;
        }
        tx[i] = x;
        ty[i] = y;
    }
    towns_put();
}

void towns_put()
{
    int i;
```

```
    setfillstyle(SOLID_FILL, BLACK);
    bar(421, 21, 619, 349);
    setcolor(CYAN);
    setfillstyle(SOLID_FILL, LIGHTBLUE);
    for(i=0;i<N;i++) fillellipse(tx[i]+430, ty[i]+35, 3, 3);
}

void distance_calc()
{
    int i, j;
    double dx, dy;

    for(i=0;i<N;i++)
        for(j=i;j<N;j++) {
            dx = ((double)(tx[i]-tx[j]))/100.0;
            dy = ((double)(ty[i]-ty[j]))/100.0;
            d[i][j] = sqrt(dx*dx+dy*dy);
            d[j][i] = d[i][j];
        }
}

void w_calc()
{
    int k, i, l, j;

    for(k=0;k<N;k++)
        for(i=0;i<N;i++)
            for(l=0;l<N;l++)
                for(j=0;j<N;j++) w[k][i][l][j] = 0.0;

    for(k=0;k<N;k++)
        for(i=0;i<N;i++)
            for(l=0;l<N;l++)
                for(j=0;j<N;j++) {
                    if( (k*N+i) > (l*N+j) ) continue;
                    if( (i == j+1) || ((i == 0)&&(j == 6)) ) {
                        w[k][i][l][j] -= alpha*d[k][l];
                        w[l][j][k][i] -= alpha*d[k][l];
                    }
                    if(((k!=l)&&(i==j))||((k==l)&&(i!=j))) {
                        w[k][i][l][j] -= 2*beta;
                        w[l][j][k][i] -= 2*beta;
                    }
                }

    for(k=0;k<N;k++)
```

```
        for(i=0;i<N;i++) h[k][i] = 2*beta;
}

void path_draw()
{
    int k,l,i,j,s;
    int x1,y1,x2,y2;

    for(s=1;s<=N;s++) {
        i = s%N;
        j = s-1;
        for(k=0;k<N;k++) {
            if(x[k][i] > 0.5) {
                x1 = tx[k]+430;
                y1 = ty[k]+35;
                for(l=0;l<N;l++) {
                    if(x[l][j] > 0.5) {
                        x2 = tx[l]+430;
                        y2 = ty[l]+35;
                        setcolor(RED);
                        line(x1,y1,x2,y2);
                    }
                }
            }
        }
    }
}

void schedule_disp(x,y)
  int  x,y;
{
    int basey = 280;
    int width = 25;

    if(y == 1) {
        setcolor(BLACK);
        setfillstyle(SOLID_FILL,LIGHTGREEN);
        bar(230,basey+x*width,345,basey+20+x*width);
    }
    else if(y == 0) {
        setcolor(WHITE);
        setfillstyle(SOLID_FILL,BLACK);
        bar(230,basey+x*width,345,basey+20+x*width);
    }

    switch(x) {
```

```
        case 0:
            outtextxy(230,basey/*+width*0*/,"1. T=k/t");
            break;
        case 1:
            outtextxy(230,basey+width*1,"2. T=k/sqrt(t)");
            break;
        case 2:
            outtextxy(230,basey+width*2,"3. T=k/log(t)");
            break;
    }
}

void schedule_select()
{
    char c = 0;
    int  i;

    setcolor(WHITE);
    rectangle(220,240,380,370);
    outtextxy(230,250,"スケジュール");
    for(i=0;i<3;i++) schedule_disp(i,0);

    schedule_disp(schedule,1);
    while(1) {
        c = getch();
        if(c == 13) break;
        if(c == ' ') {
            i = schedule;
            schedule++;
            if(schedule>2) schedule = 0;
        }
        schedule_disp(i,0);
        schedule_disp(schedule,1);
    }
    setfillstyle(SOLID_FILL,BLACK);
    bar(219,239,381,371);
}

void parm_disp(x,y)
  int  x,y;
{
    int basey = 256;
    int width = 16;

    _setcursortype(_NOCURSOR);
```

```
if(x == 4) {
    if(y == 1) {
        setcolor(BLACK);
        setfillstyle(SOLID_FILL, LIGHTGREEN);
        bar(230, basey+width*4, 266, basey+width*5);
    }
    else if(y == 0) {
        setcolor(WHITE);
        setfillstyle(SOLID_FILL, BLACK);
        bar(230, basey+width*4, 360, basey+width*5);
    }
}
else {
    if(y == 1) {
        textcolor(T_BLACK);
        setfillstyle(SOLID_FILL, LIGHTGREEN);
        bar(290, basey+x*width, 360, basey+width+x*width);
    }
    else if(y == 0) {
        textcolor(T_WHITE);
        setfillstyle(SOLID_FILL, BLACK);
        bar(290, basey+x*width, 360, basey+width+x*width);
    }
}

switch(x) {
    case 0:
        gotoxy(38, 17);
        cprintf("%4.2lf", alpha);
        break;
    case 1:
        gotoxy(38, 18);
        cprintf("%4.2lf", beta);
        break;
    case 2:
        gotoxy(38, 19);
        cprintf("%4.2lf", tau);
        break;
    case 3:
        gotoxy(38, 20);
        cprintf("%4.2lf", u0);
        break;
    case 4:
        outtextxy(230, basey+width*4, "Quit");
        break;
}
```

```
}

void parm_change()
{
    char c = 0;
    int  i,j;
    int  basey = 256;
    int width = 16;

    setcolor(WHITE);
    rectangle(220,240,380,370);
    outtextxy(230,basey,         "alpha =");
    outtextxy(230,basey+width,   "beta  =");
    outtextxy(230,basey+width*2,"tau   =");
    outtextxy(230,basey+width*3, "u0    =");
    for(i=0;i<5;i++) parm_disp(i,0);

    parm_disp(0,1);
    i = 0;
    while(1) {
        while(1) {
            c = getch();
            if(c == 13) break;
            if(c == ' ') {
                j = i;
                i++;
                if(i>4) i = 0;
            }
            parm_disp(j,0);
            parm_disp(i,1);
        }
        if(i == 4) break;
        textcolor(T_WHITE);
        _setcursortype(_NORMALCURSOR);
        gotoxy(30,23);
        cprintf("?");
        switch(i) {
            case 0:
                scanf("%lf",&alpha);
                break;
            case 1:
                scanf("%lf",&beta);
                break;
            case 2:
                scanf("%lf",&tau);
                break;
```

```
            case 3:
                scanf("%lf",&u0);
                break;
        }
        /*_setcursortype(_NOCURSOR);*/
        parm_disp(i,1);
        gotoxy(30,23);
        cprintf("          ");
    }
    setfillstyle(SOLID_FILL,BLACK);
    bar(219,239,381,371);
    clrscr();
}

double rn()
{
    double ret;

    ret = ((double)rand())/RAND_MAX;
    return(ret);
}

double f(u)
  double u;
{
    return(1/(1+exp(-1*u)));
}

double g(x)
  double x;
{
    if(x != 0.0) return(x*log(x));
    else         return(0.0);
}

void hop1(index)
  int    index;
{
    int    i,j,k,l;
    double u;

    k = index/N;
    i = index%N;
    u = 0.0;
    for(l=0;l<N;l++)
        for(j=0;j<N;j++)
```

```
            u += w[k][i][l][j]*x[l][j];
        u += h[k][i];

        if(u > 0.0)       x[k][i] = 1.0;
        else if(u < 0.0) x[k][i] = 0.0;
    }

    void hop2()
    {
        int    i, j, k, l;
        double net[N][N];

        for(k=0;k<N;k++)
            for(i=0;i<N;i++)
                net[k][i] = 0.0;

        for(k=0;k<N;k++)
            for(i=0;i<N;i++) {
                for(l=0;l<N;l++)
                    for(j=0;j<N;j++)
                        net[k][i] += w[k][i][l][j]*x[l][j];
                net[k][i] += h[k][i];
            }

        for(k=0;k<N;k++)
            for(i=0;i<N;i++) {
                u[k][i] = (1-DT/tau)*u[k][i]+DT*net[k][i];
                x[k][i] = f(u[k][i]/u0);
            }
    }
```