研 究 論 文

グラフ理論に基づいた正則な係数行列の構築によるマトリックス法 LCI の一般化

福原　一朗・本藤　祐樹

A Generalization of Matrix-based Life Cycle Inventory Analysis by the Construction of a Regular Coefficient Matrix using the Graph Theory

Ichiro FUKUHARA and Hiroki HONDO

Synopsis:

Background, Aim, and Scope. Matrix-based life cycle inventory (LCI) analysis can be effectively used to evaluate the environmental impact of a product system including closed loops (e.g. reuse and recycling). However, the matrix-based method has not yet been widely used because most life cycle assessment (LCA) practitioners find it complicated and difficult to satisfactorily construct a regular coefficient matrix from a large amount of input/output data collected from all the processes that compose a product system. The authors aim to develop a method to construct a regular coefficient matrix, which would enable the widespread use of the matrix method. The method proposed in the authors' previous paper can be used to redescribe a product system as a geometrical figure and construct a regular coefficient matrix using graph theory. However, the method cannot be used when the geometrical figure corresponding to a product system is a non-planar graph. The objective of the present study is to improve and generalize the method so that it can be used with an arbitrary product system.

Methods. First, a product system is redescribed as a geometrical figure by using the five basic components as is done in the previous paper. The redescribed geometrical figure can be expressed as a graph that consists of abstract concepts of nodes and edges in the graph theory. Then, a coefficient matrix for the product system is constructed from the constraint conditions of the product system (e.g. the balance of energy and materials). Because the coefficient matrix must be a regular matrix, the constraint conditions are required to be linearly independent of each other. A combination of linearly independent equations can be found by using a spanning tree that is constructed from the graph corresponding to the product system. A spanning tree can be constructed from every connected graph irrespective of whether the connected graph is planar or not; therefore, a regular coefficient matrix can be constructed from an arbitrary product system.

Results and Discussion. A general method was developed to construct a regular coefficient matrix for an arbitrary product system based on graph theory. The validity of the developed method was demonstrated by using two simple numerical examples. The authors would like to emphasize that the developed method contributes to the generalization and the widespread use of the matrix-based LCI analysis. The generalized algorithm proposed in the present study enables us to construct a regular coefficient matrix and carry out LCI analysis using computer software. The authors plan to create such software, which would enable LCA practitioners to easily carry out the matrix-based LCI analysis without having to manually construct a coefficient matrix.

Keywords: Graph theory; life cycle assessment (LCA); matrix method; regular coefficient matrix; spanning tree

福原　一朗／横浜国立大学大学院環境情報学府／〒 240-8501 横浜市保土ヶ谷区常盤台 79-7
本藤　祐樹／横浜国立大学大学院環境情報研究院／〒 240-8501 横浜市保土ヶ谷区常盤台 79-7
Ichiro FUKUHARA, Hiroki HONDO / Graduate School of Environment and Information Sciences, Yokohama National University / 79-1 Tokiwadai, Hodogaya-ku, Yokohama 240-8501

1. はじめに

ライフサイクルインベントリ（LCI）分析手法のマトリックス法[1-3]は、プロセス間の財フローの入出力を行列の形で表現し、逆行列計算によって製品システム全体の環境負荷を整合的に求める手法である[4-7]。さらに、マトリックス法は、感度解析や信頼性分析の高度な解析手法への発展が期待できる[4, 8, 9]など、LCI分析ツールとして有用な点を多く持つが、その特性を熟知する研究者による利用に留まっている[4]。

LCI分析では、評価対象の製品システムの財フローを把握しなければならないが[10]、一般的に製品システムのプロセス数や財フロー数は相当数にのぼる[11]。そのため、多くの財フローからバランス式を抽出するマトリックス法では、適切な係数行列を構築することは難しく、半ば手作業で試行錯誤的に行われてきた。

適切な係数行列の構築が難しい最大の理由は、互いに依存関係にある制約条件（セレクション条件や境界条件）が存在することである。図1はリサイクルを単純化したフロー図であり、S_1には製造時の再生素材の使用率、S_2にはリサイクル率が与えられる。この2つの制約条件は互いに依存関係にあり、片方の制約条件が設定されると、もう一方は自ずと決定する。

この依存関係について、同様な逆行列計算を行う産業連関分析手法[12]では4種類の特殊な方式[脚注1)]によって対応している[13]。例えば、図1のフロー図にマイナス投入方式[脚注2)]を適用する場合、産業連関表上でS_2のリサイクル率をマイナス投入値とすることで、S_1の再生素材の使用率が可変となるような依存関係を説明できる[脚注3)]。

マトリックス法と産業連関分析手法の双方とも、リサイクルのような再帰的な財フローが多くなると、すべての依存関係にある制約条件を係数行列に適切に反映させることは困難になると予想される。

そこで、筆者らは、マトリックス法の係数行列を求めるための一般的手順を確立させる必要があると考えた。これまで着目されなかったフロー図の形状そのものをグラフ理論により数学的に取り扱うことで、互いに依存関係にある制約条件を抽出した。そして、LCI分析ツールとしてのマトリックス法の汎用性を高めるために、正則な係数行列の体系的な構築手順を提案した[15]。

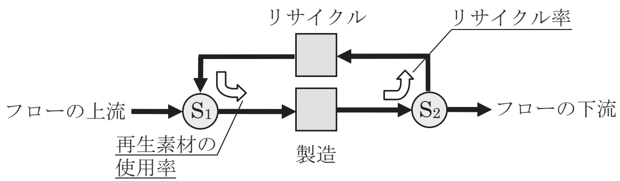


図1　リサイクルを単純化したフロー図

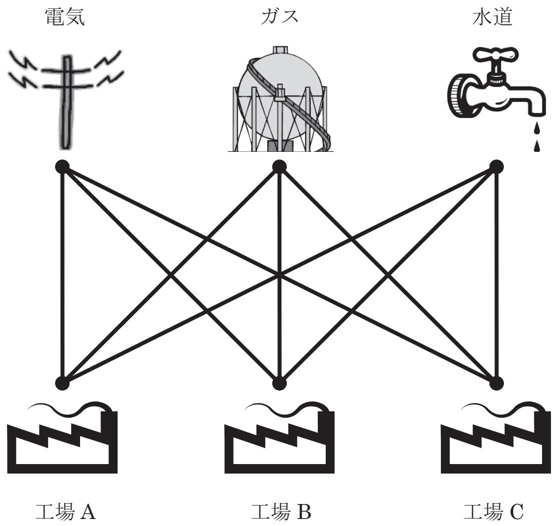


図2　前報[15]が適用できなかった製品システムの例

まず、新たに定義した基本要素を用いて、製品システムのフロー図をグラフとして表現する。そして、平面グラフが持つ数学的な法則を用いて、依存関係にある制約条件の組合せを明らかにし、正則な係数行列を構築した。しかし、この手順はフロー図の平面性に依存するため、任意の製品システムに対応していないという問題点があった。例えば、図2に例示する製品システムは完全二部グラフ$K_{3,3}$と同形であるため、フロー図の形状は非平面グラフとなる。前報[15]では、このわずか6つのプロセスで構成される製品システムに対応することができなかった。

本報ではグラフの平面性によらずに生成できる全域木を利用し、依存関係にある制約条件の組合せを求める。これにより、任意の製品システムから正則な係数行列を構築することができる。

この手法の有効性を確認するために2種類のケーススタディを実施する。

2. 既往研究のレビュー

2.1 マトリックス法

マトリックス法は、各プロセスで投入、生産される財フロー（原材料、生産物、副生産物などのマテリアルやエネルギー）と環境フロー（環境負荷物質、自然資源など）が、プロセス活動量に比例するという線型モデルを使用する。

脚注1）　一括方式、トランスファー方式、マイナス投入方式（ストーン方式）、分離方式がある。

脚注2）　副産物を、発生部門（列）にマイナス、消費部門（列）にプラスで計上する方法であり、副産物ごとに発生源と投入先を考慮したモデルを構築できる[13]。

脚注3）　投入係数に負値をとる要素があるため、任意の最終需要に対して正値の生産額が得られる保証はない。つまり、最終需要によっては生産量が負値となり、現実を反映していないモデルとなる問題点がある[14]。

そして、製品システム全体の制約条件を満足する各プロセス活動量を、式（2.1.1）に示す連立一次方程式を解くことによって算出する[2]。ここで、プロセス活動量ベクトルを$\mathbf{q}=(q_1, q_2, \cdots, q_v)^{\mathrm{T}}$で表す。ただし、Tは転置行列を示す。

$$A\mathbf{q}=\boldsymbol{\alpha} \qquad (2.1.1)$$

次に、後述する「システム境界」における財フローを境界ベクトル$\boldsymbol{\alpha}=(0, \cdots, 0, f)^{\mathrm{T}}$として与えると、活動量ベクトル$\mathbf{q}$は、次式（2.1.2）の逆行列計算により求められる。

$$\mathbf{q}=A^{-1}\boldsymbol{\alpha} \qquad (2.1.2)$$

式（2.1.2）が一意な解を持つための条件は、境界ベクトル$\boldsymbol{\alpha}$が非ゼロであり、さらに係数行列Aが正則行列であることである。

また、環境フローを表す行列をBとすれば、環境負荷ベクトル$\boldsymbol{\beta}$は次式（2.1.3）で求められる。

$$\boldsymbol{\beta}=B\mathbf{q}=BA^{-1}\boldsymbol{\alpha} \qquad (2.1.3)$$

このように、マトリックス法は環境負荷等を逆行列計算により整合的に求めることができる手法である。

2.2 フロー図の構築規則とグラフ理論的特徴

前報[15]で提案したフロー図の構築規則の概要を述べる。図3のプロセスと図4のセレクションの基本要素を用いて製品システムをグラフとして表現する。

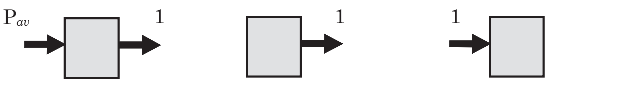
（1）通常プロセスとセレクション

図3 (a) の通常プロセスは、財iをP_{av}投入して、財jを1出力する1対1の入出力を持つプロセスであり、P_{av}をプロセスP_vの機能フローの係数とする。

図4のセレクションは、財フローの単純な分岐と合流を示しており、財の損失や環境負荷の発生はない。

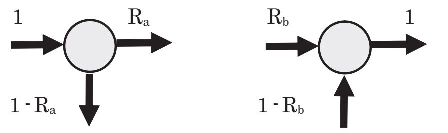
（2）境界プロセス

システム境界には、製品システム全体の財の入出力についての条件を設定した境界プロセスを少なくとも1つ配置する。この境界プロセスは、図3 (b) のインプット・プロセスと図3 (c) のアウトプット・プロセスである。インプット・プロセスは、財を製品システムに供給するだけのプロセスであり、アウトプット・プロセスは、製品システムから財が投入されるだけのプロセスである。



(a) 通常プロセス　(b) インプットプロセス　(c) アウトプットプロセス

図3　使用するプロセスの種類



(a) 分岐セレクション　(b) 合流セレクション

$\mathrm{R_a}$：分岐率，$\mathrm{R_b}$：合流率

図4　使用するセレクションの種類

（3）財フローの接続規則

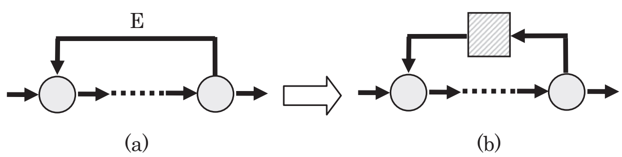
隣接する要素間の財フローは1つであり多重フローは無い。財フローは互いに異なる要素間を接続し、セレクションには3つの異なるプロセスが必ず接続する。つまり、財フローは、プロセス-プロセス（P-P）間と、プロセス-セレクション（P-S）間の2種類に限定される。実際には、プロセス内リサイクルのようなプロセス内部で閉じるタイプのサイクルがあるが、図5のように損失、負荷がゼロであるダミープロセスを辺Eに挿入することで、セレクション同士が直接的に接続することを回避して表現できる。このダミープロセスは、$\mathrm{P}_{av}=1$の通常プロセスである。

（4）多機能プロセス

実際の製品システムには、図6 (a) のような多機能プロセスが一般的に存在する。そこで、多機能プロセス内部に仮想的な1対1の入出力プロセスとセレクションがあると仮定する。つまり、投入した複数の財が合流して1つのフローとして仮想プロセスに入力され、出力された財フローが複数の財に分岐してプロセスの外部に生産される。従って、図6 (a) の多機能プロセスは、図6 (c) のように基本要素のみで構成される二分木グラフで表される。

製品システムのフロー図は、以上の（1）～（4）の構築規則によって次のようなグラフとして表現できる。

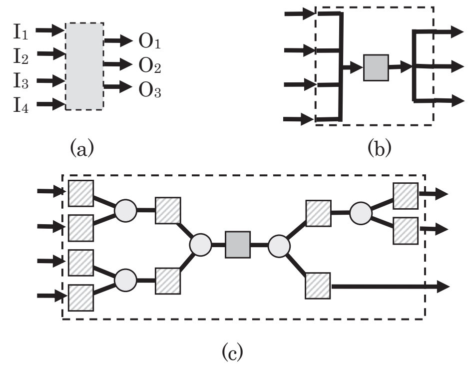
通常プロセスは次数2の頂点となり、セレクションは次数3の頂点となる。そして、境界プロセスは次数1の頂点として少なくとも1つ存在する。従って、グラフの頂点の次数は高々3となる。構築規則（3）より、次数3同士を結ぶ辺はなく、すべての頂点は少なくとも1つの辺に接続し、隣接する頂点の間に多重辺はない。さらに、頂点から出た辺が元の頂点に直接戻る自己ループは形成されない。つまり、このグラフは単純連結グラフである。



(a)　(b)

は、ダミープロセスを示す。

図5　セレクションが連結する場合の表記



は、ダミープロセスを示す。

は、多機能プロセスの境界を示す。

図6　多機能プロセスのグラフ表現の例

3. 一次従属ではない制約条件の選び方

マトリックス法の係数行列は、製品システムの制約条件から構築されるが、係数行列は式（2.1.2）の逆行列計算に用いるため正則でなければならない。つまり、係数行列の構築に用いられる制約条件が一次従属ではないことが求められる。

しかし、リサイクルやリユースなどが製品システムに含まれる場合、一次従属となる制約条件を選択することも可能となる。係数行列の正則性を保証するために、一次従属ではない制約条件を抽出する手順を以下に示す。

3.1 制約条件の一次従属性

3.1.1 制約条件の種類

2.2節の規則に従って表現されるフロー図に含まれるすべての制約条件は、次の3種類に分類される。

(1) 財フロー保存の条件

これは隣接する複数のプロセス間における財の入出力の総和に関する制約条件である。2.2節（3）の規則により、フロー図には以下の2種類の接続形態しか存在しない。それぞれの制約条件を次に示す。

1）2つのプロセスが1つの辺により直接接続する場合

プロセスP_Aの出力がプロセスP_Bに入力する場合、次式(3.1.1）の制約条件が与えられる。

$$P_A(\mathrm{output}) = P_B(\mathrm{input}) \qquad (3.1.1)$$

2）3つのプロセスが1つのセレクションを介して接続する場合

図7 (a）のようにプロセスP_Aの出力がプロセスP_BとP_Cに分岐する場合、セレクションには次式（3.1.2）の制約条件が与えられる。

$$P_A(\mathrm{output}) = P_B(\mathrm{input}) + P_C(\mathrm{input}) \qquad (3.1.2)$$

また、図7 (b）のようにプロセスP_AとP_Cの出力がプロセスP_Bに投入される場合、セレクションには次式（3.1.3）の制約条件が与えられる。

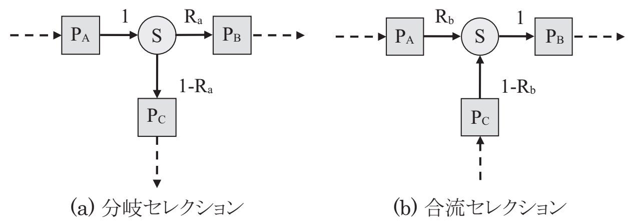


図7　セレクションの制約条件

$$P_A(\mathrm{output}) + P_C(\mathrm{output}) = P_B(\mathrm{input}) \qquad (3.1.3)$$

上記の1）と2）の制約条件は、フロー図から機械的に求めることができる。

(2) セレクション条件

これはセレクションのみに設定される、財フローの分岐率や合流率に関する制約条件である。セレクションでは財フローの分岐先や合流元が2方向あり、制約条件は1方向に1つずつ、計2つ作ることができる。

図7（a）の分岐セレクション（P_Bへの分岐率がR_a）の場合

$$\begin{cases} P_A(\mathrm{output}) \times R_a = P_B(\mathrm{input}) & (3.1.4) \\ P_A(\mathrm{output}) \times (1\text{-}R_a) = P_C(\mathrm{input}) & (3.1.5) \end{cases}$$

図7（b）の合流セレクション（P_Aからの合流率がR_b）の場合

$$\begin{cases} P_B(\mathrm{input}) \times R_b = P_A(\mathrm{output}) & (3.1.6) \\ P_B(\mathrm{input}) \times (1\text{-}R_b) = P_C(\mathrm{output}) & (3.1.7) \end{cases}$$

本報では、フロー図の形状をグラフとして扱い、グラフの辺には財フローが必ずあるものと仮定する。従って、分岐率と合流率については、100%の分岐や0%の合流を想定しない。つまり、R_aとR_bは次式（3.1.8）を満たすこととする。

$$0 < R_a < 1, \quad 0 < R_b < 1 \qquad (3.1.8)$$

(3) 境界条件

製品システム全体の財の入出力量を与えるための条件であり、システム境界に少なくとも1つ設定される。

3.1.2 サイクルと単純分岐の制約条件

図8では、セレクションS_2からプロセスP_3に分岐するサイクルによってリサイクルを表現している。製品への再生素材の使用比率は合流セレクションS_1に設定され、リサイクル率はS_2に設定される。S_1は図7(b）に対応し、式（3.1.3）の財フロー保存の条件と、式(3.1.6, 3.1.7)の2つのセレクション条件が与えられる。しかし、この3つの式は一次従属であり、制約条件が1つ過剰である。本報では、セレクション条件を1つのみとすることで一次従属問題を回避する。

同様に、S_2についても一次従属にならないように制約条件を2つ与え、図8全体の制約条件は次式（3.1.9）に示す4つの式となる。ただし、各プロセスの機能フローの係数を

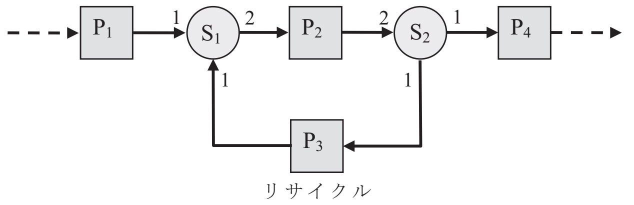


図8　サイクルの制約条件

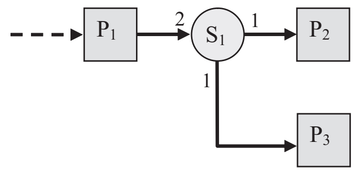


図9　単純分岐の制約条件

1とする。

$$\left.\begin{array}{ll} \text{財フロー保存の条件} & S_1: P_1 + P_3 = P_2,\quad S_2: P_2 = P_3 + P_4 \\ \text{セレクション条件} & S_1: P_1／0.5 = P_2,\quad S_2: P_2 \times 0.5 = P_3 \end{array}\right\} \quad (3.1.9)$$

留意すべきは、各セレクションに与える制約条件が一次従属ではないにもかかわらず、サイクル全体に関係する式(3.1.9)の制約条件は一次従属となることである。なぜなら、サイクルを構成する2つのセレクションには、一方のセレクション条件が他方のセレクション条件を決定させるという従属関係があるためである。従って、式（3.1.9）から過剰な条件式をさらに1つ除くことで、一次従属ではない制約条件の組合せが得られる。

このような従属関係は、サイクルを構成する2つのセレクション間以外にも存在する。図9は、サイクルを作らない単純分岐（分岐率50%）のフロー図であり、プロセスP_1からの財フローが、セレクションS_1で分岐して境界プロセスP_2とP_3に接続する。このフローには、財フローの保存条件、セレクション条件、プロセスP_2の境界条件、プロセスP_3の境界条件、合計4つの制約条件が式（3.1.10）のように与えられる。

$$\left.\begin{array}{ll} \text{財フロー保存の条件} & P_1 = P_2 + P_3 \\ \text{セレクション条件} & P_1 \times 0.5 = P_2 \\ \text{プロセス } P_2 \text{ の境界条件} & P_2 = 1 \\ \text{プロセス } P_3 \text{ の境界条件} & P_3 = 1 \end{array}\right\} \quad (3.1.10)$$

このうち、境界条件は式（2.1.2）の境界ベクトル$\boldsymbol{\alpha}$の実数要素を右辺に持つ式であり、製品システムに少なくとも1つ必要である。そこで、境界条件を1つ除いた3つの制約条件についてみると、これらは一次従属となっている。つまり、単純分岐の場合には、セレクション条件と境界条件が従属関係であり、どちらか1つの制約条件が過剰となる。

図8の例では、セレクションS_1とS_2の両方に制約条件が設定されていたが、逆に制約条件が1つも設定されない場合はどうだろうか。当然ながら、財フローのバランス式が不足し、係数行列を構築することは不可能となる。図8について言えば、リサイクル率か、再生素材の使用比率のどちらか1つが判明していることが必須となる。本報では、ISO14040s[脚注4)] で規定されている、LCI分析の「目的及び調査範囲の設定」[16] が正しく実施されることを想定して、必要最低限のセレクション条件や境界条件が与えられていると仮定する。

3.2 全域木を用いた制約条件の抽出法

一次従属ではない制約条件から正則な係数行列を求めることができれば、逆行列計算によって解を一意に求めることができる。しかし、製品システムの規模が大きくなると、従属関係にある制約条件を見つけ出すことは困難となる。それゆえに、ここで解くべき問題は、従属関係にある制約条件を自動的に抽出する手順である。これを、グラフ理論を用いて考察する。

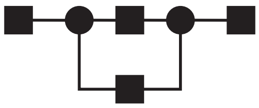
今、製品システムを表すグラフの頂点の次数は高々3となっているので、1つのサイクルは分岐の開始点と終了点に次数3の頂点を配置することで作られる。では、財フローが次々と分岐し複数のサイクルが共通の辺を持つように隣接する場合ではどうだろうか。これも同様に1つのサイクルが追加されるたびに、分岐の開始点と終了点に次数3の頂点が2つ必要になる。このように、1つのサイクルには最低2つの次数3の頂点が含まれる。このうち1つの次数3の頂点にのみセレクション条件を与えることで、すべての制約条件が一次従属となることを回避できる。

本報では、図10に示す全域木を用いて、制約条件が一次独立となる組合せを自動的に抽出する。

まず、製品システムの制約条件が解を求めるために十分であるかどうかを調査する。それは、セレクション条件が与えられていない次数3の頂点に接続する辺のうち、サイクルを構成する任意の1辺を除去し、残されたグラフが連結か非連結かどうかにより判定できる。非連結の場合は、制約条件が不足し、解が一意に定まらないことを意味している。3.1.2項で述べたように、製品システムには必要最低限の制約条件が与えられていると仮定したため、残されたグラフは連結となるはずである。

次に、残されたグラフにサイクルがあるかを調べる。サイクルがある場合、制約条件が過剰であり、一次従属であることを示している。適切な係数行列を構築するためには、そのサイクルから制約条件を1つ落とす必要がある。1つのサイクルには分岐と合流のセレクションに対応する2つの次数3の頂点が含まれるので、その頂点のいずれかのセ

脚注4)　ISO（2006）: International Standard ISO 14040. Environmental management - life cycle assessment - principles and framework. ISO/TC207/SC5


(a) 元のフロー図

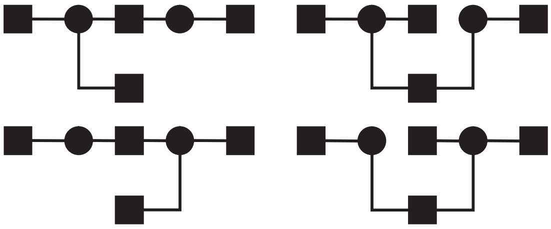

(b) 4通りの全域木

図10　全域木

レクション条件を削除する。そして、セレクション条件を削除した頂点に接続する辺のうち、サイクルを構成する任意の1辺を除去する。

上記の作業をグラフのすべてのサイクルに含まれる次数3の頂点に対して繰り返すと、最終的にグラフは全域木になる。グラフが全域木になったとき、制約条件は一次独立となり正則な係数行列が得られる。

全域木は、連結グラフの平面性に依存せずに作ることができるため、本報のマトリックス法は任意の製品システムに対応できる。

具体的な手順の詳細は5章のケーススタディに示す。

4. 係数行列の構築手法

3章では、全域木を用いることで、制約条件を一次従属ではないように整理した。この章では、制約条件式の数とプロセス数が一致することを示し、正方で正則な係数行列を構築する。

まず、前報[15]と同様に以下のように定義する。

グラフGの頂点集合$V(G)$のうち、プロセスを示す頂点集合を$P(G)$、属する元の数を$p(G)$とする。同様に、セレクションを示す頂点集合を$S(G)$、属する元の数を$s(G)$とする。

グラフGの辺集合を$E(G)$、属する元の数を$\varepsilon(G)$とする。$E(G)$は、一端が$S(G)$の元の頂点に接続する辺集合$E_{PS}(G)$（P-S間の財フローに対応）と、両端が$P(G)$の元の頂点に接続する辺集合$E_{PP}(G)$（P-P間の財フローに対応）の和である。

（定義終）

4.1 財フロー保存の条件

前報[15]で示したように、製品システム内には、次数3の頂点に接続する3辺の辺集合と、次数2の頂点同士を接続する辺に対応する、次に示す2種類の財フロー保存の条件が含まれる。

(1) 次数3の頂点に接続する3辺に対応する財フロー保存の条件

3つのプロセスがセレクションを中心に、$E_{PS}(G)$の元である辺によって接続する。この財フロー保存の条件はセレクションごとに作ることができる。セレクションの数は$s(G)$個あるため、3つのプロセス間にある財フロー保存の条件の集合を$Ms(G)$と定義すれば、$Ms(G)$に属する元の数も$s(G)$である。

(2) 次数2の頂点同士を接続する辺に対応する財フロー保存の条件

2つのプロセスが$E_{PP}(G)$の元の辺によって互いに接続する。辺集合$E_{PP}(G)$は、辺集合$E(G)$のうち、$E_{PS}(G)$を除いたものである。さらに、上記（1）から、$E_{PS}(G)$の元の数は$s(G)$の元の数の3倍であるので、2つのプロセス間の、財フロー保存の条件の集合を$Me(G)$と定義すれば、$Me(G)$に属する元の数は、$\varepsilon(G)–3s(G)$となる。

ここで、プロセス間の財フローバランス条件集合$Ms(G)$と$Me(G)$の和集合を$M(G)$とし、属する元の数を$m(G)$と定義すれば、これらの間には次式（4.1.1）が成り立つ。

$$\left.\begin{array}{l} Ms(G) \subset M(G),\ Me(G) \subset M(G), \\ Ms(G) \cap Me(G) = \varnothing,\ Ms(G) \cup Me(G) = M(G), \\ m(G) = s(G) + \{\varepsilon(G) - 3s(G)\} = \varepsilon(G) - 2s(G) \end{array}\right\} \quad (4.1.1)$$

一方、$P(G)$と$S(G)$は、それぞれグラフGの頂点集合$V(G)$の真部分集合であり、$P(G)$と$S(G)$の元は互いに素である。従って、次式（4.1.2）が成り立つので、式（4.1.3）も成り立つ。

$$\left.\begin{array}{l} P(G) \subset V(G),\ S(G) \subset V(G), \\ P(G) \cap S(G) = \varnothing,\ P(G) \cup S(G) = V(G) \end{array}\right\} \quad (4.1.2)$$

$$\upsilon(G) = p(G) + s(G) \quad (4.1.3)$$

4.2 セレクション条件と境界条件

セレクションを示す頂点集合$S(G)$のうち、全域木を作るときに次数を1減らす頂点集合を$S_cut(G)$とし、属する元の数を$s_cut(G)$とする。また、全域木に残る次数3の頂点集合を$S_remain(G)$とし、属する元の数を$s_remain(G)$とする。頂点集合$S_cut(G)$と$S_remain(G)$の元は互いに素であるので、次式（4.2.1）が成り立つ。

$$s(G) = s\_remain(G) + s\_cut(G) \quad (4.2.1)$$

除去される辺の本数は閉路階数$\gamma(G)$と等しく、$s_cut(G)$とも等しいので、次式（4.2.2）が成り立つ。

$$\gamma(G) = s\_cut(G) \quad (4.2.2)$$

また、頂点数nと辺の数mについて式（4.2.3, 4.2.4）が成り立つ。

$$\begin{aligned} n = \upsilon(G) &= p(G) + s(G) \\ &= p(G) + s\_remain(G) + s\_cut(G) \end{aligned} \quad (4.2.3)$$

$$m = \varepsilon(G) \qquad (4.2.4)$$

式（4.2.2）～（4.2.4）を閉路階数の式$\gamma(G) = m - n + 1$に代入する。

$$s\_cut(G) = \varepsilon(G) - \{p(G) + s\_remain(G) + s\_cut(G)\} + 1 \qquad (4.2.5)$$

式（4.2.5）を整理し、次式（4.2.6）を得る。

$$\varepsilon(G) = p(G) + s\_remain(G) + 2 \times s\_cut(G) - 1 \qquad (4.2.6)$$

式（4.1.1）と式（4.2.1）から次式（4.2.7）が得られ、

$$m(G) = \varepsilon(G) - 2 \times \{s\_remain(G) + s\_cut(G)\} \qquad (4.2.7)$$

式(4.2.6)を式(4.2.7)に代入して$\varepsilon(G)$を消去し、次式(4.2.8)を得る。

$$p(G) = m(G) + s\_remain(G) + 1 \qquad (4.2.8)$$

式（4.2.8）から、財フロー保存の条件、セレクション条件、システムの境界条件と、プロセス数との関係が導かれる。まず、プロセス間の財フロー保存の条件式は集合$M(G)$で求められ、その要素数は$m(G)$である。次に、3.2節で述べたように、グラフの全域木に含まれる次数3の頂点だけにセレクション条件を与えるので、$s_remain(G)$本のセレクション条件式が作られる。最後にシステムの境界条件式が1本ある。これらの制約条件の数は式（4.2.8）の右辺に示され、左辺のプロセス数と等しくなる。

前報[15]と同様に、係数行列Aを（$m(G) + s_remain(G)$行$\times p(G)$列）とすると、Aは、次の2つの部分行列が結合した正方行列とみなせ、次式（4.2.9）で表される。

$$A = \begin{bmatrix} A_{\mathrm{M}} \\ \hline A_{\mathrm{RF}} \end{bmatrix} \left. \begin{matrix} m(G)\text{行} \\ s\_remain(G) + 1\text{行} \end{matrix} \right\} \times p(G)\text{列} \qquad (4.2.9)$$

A_{M}（$m(G)$行$\times p(G)$列）：プロセス間にある財フロー保存の条件の集合を示す。

A_{RF}（$s_remain(G) + 1$行$\times p(G)$列）：セレクション条件と境界条件を示す。

プロセス間の財フローバランス条件集合$M(G)$の要素を$M_i(G); i = 1, \cdots, m(G)$とすれば、行列$A_{\mathrm{M}}$の$ij$要素$a_{m_{ij}}$は、次式（4.2.10）で表される。

$$a_{mij} = \begin{cases} -\mathrm{P}_{aj} & ,M_i(G)\text{に対応する財フローから、プロセス}\mathrm{P}_j\text{に財を}\mathrm{P}_{aj}\text{入力} \\ 1 & ,\text{プロセス}\mathrm{P}_j\text{から}M_i(G)\text{に対応する財フローに1出力} \\ 0 & ,\text{プロセス}\mathrm{P}_j\text{は}M_i(G)\text{に対応する財フローに含まれない} \end{cases} \qquad (4.2.10)$$

全域木の、i番目の次数3の頂点において、プロセスP_uからプロセスP_vへ、$R_a (0 < R_a < 1)$の比率で分岐するときは、部分行列A_{RF}のij要素$a_{rf_{ij}}$は次式（4.2.11）で示される。ただし、プロセスP_vの機能フローの係数をP_{av}とする。

$$a_{rfij} = \begin{cases} R_a & ,j = u \\ -\mathrm{P}_{av} & ,j = v \\ 0 & ,j \neq u \vee j \neq v \end{cases} \qquad (4.2.11)$$

また、全域木の、i番目の次数3の頂点において、プロセスP_vへ投入される財のうち、プロセスP_uからの財が$R_b (0 < R_b < 1)$の比率を占めている場合に、部分行列A_{RF}のij要素$a_{rf_{ij}}$は次式（4.2.12）で示される。

$$a_{rfij} = \begin{cases} 1/R_b & ,j = u \\ -\mathrm{P}_{av} & ,j = v \\ 0 & ,j \neq u \vee j \neq v \end{cases} \qquad (4.2.12)$$

プロセスP_uの活動量を境界条件iとする場合に、部分行列A_{RF}のij要素$a_{rf_{ij}}$は次式（4.2.13）で示される。

$$a_{rfij} = \begin{cases} 1 & ,j = u \\ 0 & ,j \neq u \end{cases} \qquad (4.2.13)$$

5. ケーススタディ

本報で提案したマトリックス法のケーススタディを、特徴的な2つの製品システムによって行う。

第1に、3.2節で述べた、全域木を用いた制約条件の抽出手順を数値例で説明することを目的としてケーススタディを行う。既往文献[17]において計算が難しいとされたリサイクルを含む製品システムを例として取り上げる。この製品システムからは、複数の全域木が得られ、各々に対応して異なる係数行列が生成される。5.1節では、そのうち2種の全域木を取り出し、それらに対応する2種の係数行列から同一解が得られることを示す。

第2に、本報で提案する手法の有効性を示すことを目的として、前報[15]の方法では解くことができなかった製品システムを例としてケーススタディを実施する。この製品システムは、わずか5つのプロセスでありながら非平面グラフとなっており、非平面グラフとなる製品システムが現実にも十分に存在し得ることを示している。5.2節では、仮想的ではあるが現実にも十分にあり得る製品システムを例にして、非平面グラフとなる場合にも、本報の手順によって解が一意に求められることを示す。

5.1 サイクルを含む製品システム

(1) 製品システム例の説明

サイクルを含む製品システムの例として、図11に示す製品システムを取り上げる。このシステムはLCAソフト「LCAサポート」の適用例の解説において、計算が難しいとされたリサイクル型モデル[17]を参考に作成した。この製品システムは、10個のプロセスと、2つのリサイクル工程AとBを持つ。そして、P_1とP_2から原料が投入され、P_7で製品が、P_8でリサイクル財が生産される。また、P_6は別の製品システムに中間財を供給する境界プロセスである。

この製品システムから得られるフロー図を図12に示す。図中の点線は多機能プロセスの境界であり、その内部にセレクションを付加することで、多機能プロセスを表現する。

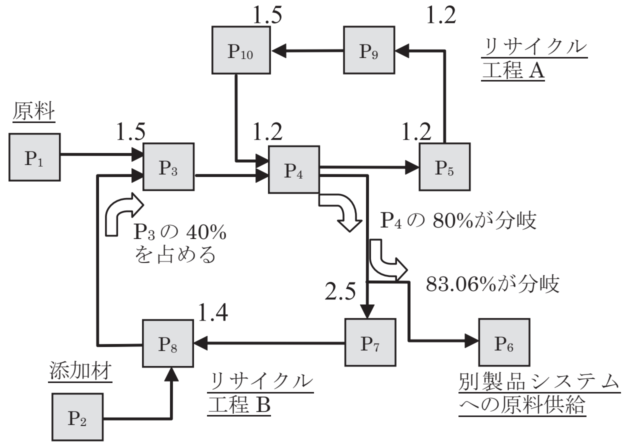


図11　サイクルを含む製品システム

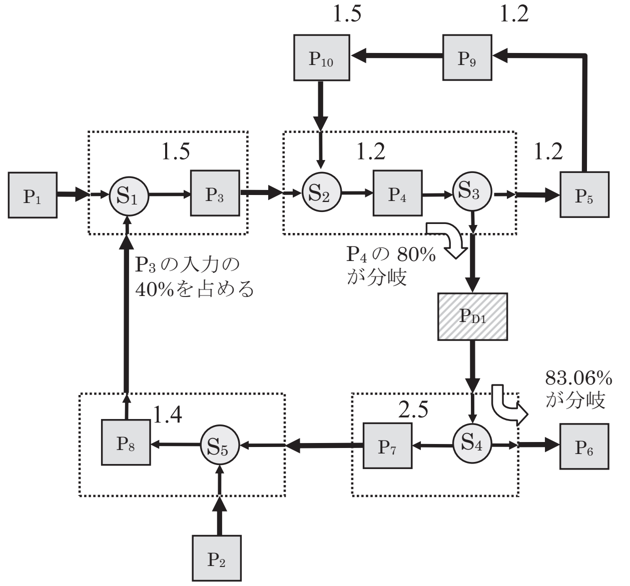


図12　サイクルを含む製品システムのフロー図

また1個のダミープロセスを付加するため、プロセス数は合計11個に増加する。各プロセスの肩の数字は機能フローの係数である。また、境界条件として、P_8でリサイクル財が1生産され、P_6で中間財が1生産されるものとする。

(2) プロセス間の財フロー保存の条件

プロセス間の財フローバランス条件集合$M(G)$を示す部分行列A_{M}を作成する。A_{M}は「3つのプロセス間にある財フロー保存の条件」もしくは「2つのプロセス間にある財フロー保存の条件」を表している。

まず、プロセスP_1とプロセスP_8で生産された財が、セレクションS_1で合流してプロセスP_3に投入されている。プロセスP_3に投入される財は、プロセス活動量q_3に機能フロー係数P_{a3}を乗じた値であり、式（5.1.1）のバランス式が成り立つ。同様に、セレクションS_2からS_5までの条件式を作る。

$$q_1 \times 1 + q_8 \times 1 - q_3 \times P_{a3} = 0 \tag{5.1.1}$$

（プロセスP_3の機能フローの係数$P_{a3} = 1.5$）

次に、プロセスP_5-P_9間の財フロー$E_{5\text{-}9}$では、プロセスP_5で生産された財がプロセスP_9に投入されている。プロセスP_9に投入される財は、プロセス活動量q_9に機能フロー係数P_{a9}を乗じた値であり、式（5.1.2）のバランス式が成り立つ。同様に、プロセスP_9-P_{10}間の財フロー$E_{9\text{-}10}$の条件式を作る。

$$q_5 \times 1 - q_9 \times P_{a9} = 0 \tag{5.1.2}$$

（プロセスP_9の機能フローの係数$P_{a9} = 1.2$）

これらをまとめた部分行列A_{M}を式（5.1.3）に示す。

(3) 全域木を用いたセレクション条件と境界条件の抽出

図12では、セレクション条件が、分岐、合流率としてセレクションS_1、S_3及びS_4に与えられている。3.2節の手順で全域木を作成し、制約条件を設定しなくてもよい頂点を見つける。その頂点に、既に制約条件が設定されている場合は、その条件を削除する。その2例を図13(b) と図13(c) に示す。

図13 (b) では、サイクルを構成するセレクションの組合せとしてS_2とS_3の組と、S_1とS_4の組を全域木から抽出する。この2組のうち、前者については、制約条件を設定する頂点と設定しない頂点の組であるため、S_2の1辺を除去する。また、後者は両方に制約条件が設定されている。ここではS_4のセレクション条件を削除し、1辺を除去する。

また、図13 (c) では、全域木からS_2とS_3の組と、S_4とS_5の組を抽出する。この2組は制約条件を設定する頂点と設定しない頂点の組であるため、セレクション条件を削除する必要は無く、1辺を除去するだけでよい。一方、サイ

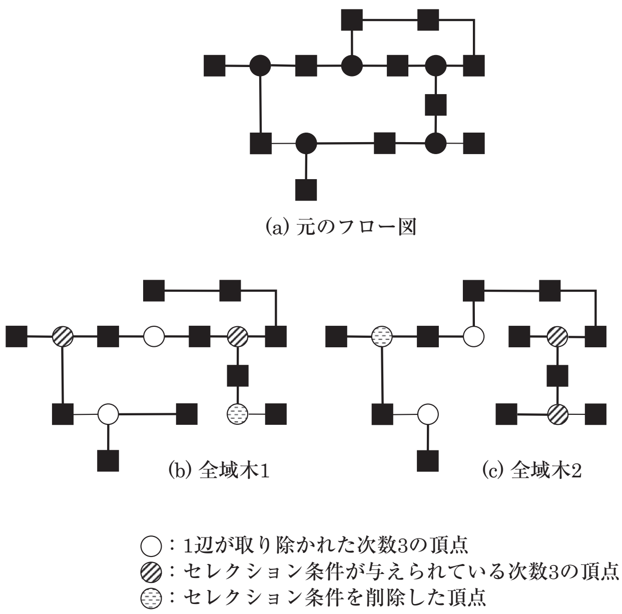


図13　サイクルを含む製品システムの全域木

クルを構成しない次数3の頂点S_1は2つ目の境界条件と従属関係にあり、どちらも制約条件が設定されている。ここでは、S_1のセレクション条件を削除する。

セレクションの分岐条件と境界条件を示す部分行列A_{RF}は、図13 (b) では式 (5.1.4)、図13 (c) では式 (5.1.5) に示される。また、境界ベクトルは、$\alpha = (0,\cdots,0,1,1)^{T}$となる。ただし、Tは転置行列を示す。

部分行列A_MとA_{RF}を式 (4.2.9) に従って結合し、プロセス活動量ベクトル$\mathbf{q}$と境界ベクトル$\boldsymbol{\alpha}$を含めたシステム全体のバランス式を作る。図13(b) では式 (5.1.6) となり、図13(c) では式 (5.1.7) となる。

式 (5.1.6) と式 (5.1.7) をそれぞれ解くと、プロセスの活動量ベクトル$\mathbf{q}$は式 (5.1.8) に示す同一解が得られる。

$$A_M = \begin{array}{cc} & \begin{array}{ccccccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & P_{10} & P_{D1} \end{array} \\ \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ E_{5\text{-}9} \\ E_{9\text{-}10} \end{array} & \begin{pmatrix} 1 & 0 & -1.5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1.2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1.2 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -2.5 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1.5 & 0 \end{pmatrix} \end{array} \tag{5.1.3}$$

$$A_{RF} = \begin{array}{cc} & \begin{array}{ccccccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & P_{10} & P_{D1} \end{array} \\ \begin{array}{c} R\text{-}S_1 \\ R\text{-}S_3 \\ F_{P6} \\ F_{P8} \end{array} & \begin{pmatrix} 0 & 0 & -1.5 & 0 & 0 & 0 & 0 & 2.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{5.1.4}$$

$$A_{RF} = \begin{array}{cc} & \begin{array}{ccccccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & P_{10} & P_{D1} \end{array} \\ \begin{array}{c} R\text{-}S_3 \\ R\text{-}S_4 \\ F_{P6} \\ F_{P8} \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0.8306 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{5.1.5}$$

$$\begin{array}{cc} & \begin{array}{ccccccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & P_{10} & P_{D1} \end{array} \\ \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ E_{5\text{-}9} \\ E_{9\text{-}10} \\ R\text{-}S_1 \\ R\text{-}S_3 \\ F_{P6} \\ F_{P8} \end{array} & \left(\begin{array}{ccccccccccc} 1 & 0 & -1.5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1.2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1.2 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -2.5 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1.5 & 0 \\ \hline 0 & 0 & -1.5 & 0 & 0 & 0 & 0 & 2.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right) \end{array} \begin{array}{c} \\ \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \\ q_{10} \\ q_{D1} \end{bmatrix} \end{array} \begin{array}{c} \\ = \end{array} \begin{array}{c} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \end{array} \tag{5.1.6}$$

$$\begin{array}{cc} & \begin{array}{ccccccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 & P_9 & P_{10} & P_{D1} \end{array} \\ \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ E_{5\text{-}9} \\ E_{9\text{-}10} \\ R\text{-}S_3 \\ R\text{-}S_4 \\ F_{P6} \\ F_{P8} \end{array} & \left(\begin{array}{ccccccccccc} 1 & 0 & -1.5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1.2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1.2 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -2.5 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1.5 & 0 \\ \hline 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0.8306 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right) \end{array} \begin{array}{c} \\ \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \\ q_{10} \\ q_{D1} \end{bmatrix} \end{array} \begin{array}{c} \\ = \end{array} \begin{array}{c} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \end{array} \tag{5.1.7}$$

$$\mathbf{q} = \left(1.5, 1.3, 1.7, 1.5, 0.25, 1.0, 0.082, 1.0, 0.21, 0.14, 1.2\right)^T \tag{5.1.8}$$

既往のLCAソフトで計算が難しいとされたリサイクル型の製品システム[17]に本報の手法を適用し、全域木を用いた制約条件の抽出手順を数値例で説明した。

任意の2つの全域木から、それぞれ異なる係数行列が生成され、いずれの係数行列からも同一解が得られることが示された。つまり、全域木を用いて制約条件を抽出する際に、どの全域木を使用すればよいのかを考慮せずに解を得ることができることが分かる。

5.2 非平面グラフとなる製品システム

(1) 製品システム例の説明

グラフが非平面となる例として、図14に示す製品システムを取り上げる。このシステムは、3つのプロセス②、③、④で素材と中間部品が製造される。また、各プロセスからの廃棄物は回収後リサイクルされ、再生素材がプロセス原料として戻される。フロー図の形状を見ると、5つのプロセス②、③、④、⑥、⑦が完全グラフK_5を形成しているため非平面グラフとなり、前報[15]の手法では解を求めることができない。

この製品システムから得られるフロー図を図15に示す。7個のプロセスに13個のダミープロセスを付加するため、プロセス数は合計20個に増加する。境界条件として、P_5で製品が1生産されるものとする。

(2) プロセス間の財フロー保存の条件

プロセス間の財フローバランス条件集合$M(G)$を示す部分行列A_Mを、5.1節（2）と同様に作成し、式（5.2.1）に示す。

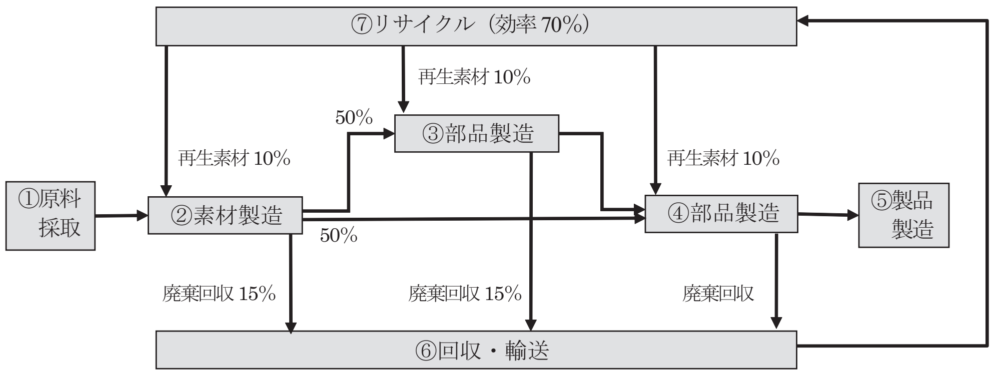


図14　非平面グラフとなる製品システム

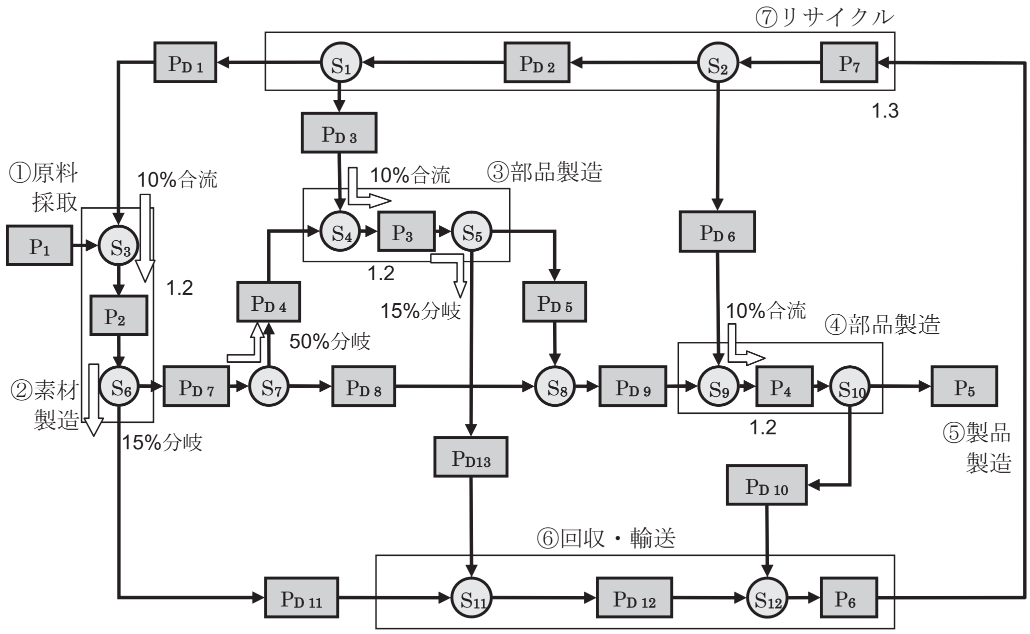


図15　非平面グラフとなる製品システムのフロー図

(3) 全域木を用いたセレクション条件と境界条件の抽出

3.2節の手順で図15から全域木を1つ作成し、図16 (b) に示す。図16 (b) では、⊘で示すセレクションS_3、S_4、S_5、S_6、S_7、S_9にセレクション条件が与えられる。

セレクションの分岐条件と境界条件を示す部分行列A_{RF}は、式 (5.2.2) のように生成される。境界ベクトルは、$\alpha=(0,\cdots,0,1)^T$となる。ただし、Tは転置行列を示す。

部分行列A_MとA_{RF}を式 (4.2.9) に従って結合し、プロセス活動量ベクトル$\mathbf{q}$と境界ベクトル$\boldsymbol{\alpha}$を含めたシステム全体のバランス式を式 (5.2.3) に示す。

式 (5.2.3) を解いてプロセスの活動量ベクトル$\mathbf{q}$を式 (5.2.4) に得る。

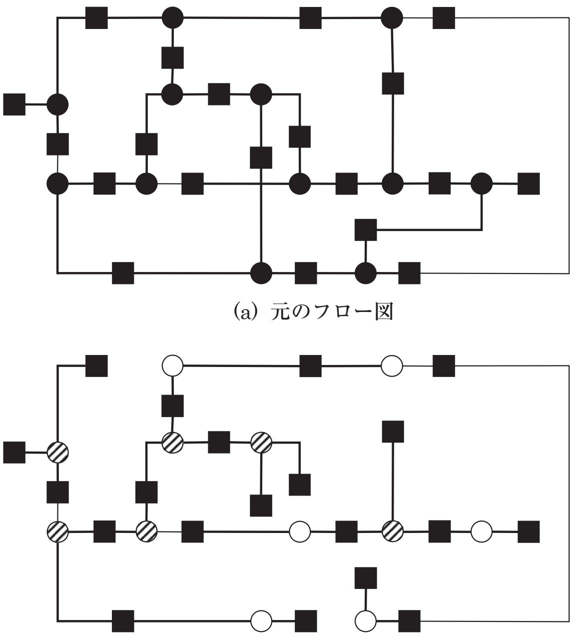
$$
A_M = \begin{array}{cc}
 & \begin{array}{cccccccccccccccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_{D1} & P_{D2} & P_{D3} & P_{D4} & P_{D5} & P_{D6} & P_{D7} & P_{D8} & P_{D9} & P_{D10} & P_{D11} & P_{D12} & P_{D13} \end{array} \\
\begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \\ S_{11} \\ S_{12} \\ E_{6\text{-}7} \end{array} &
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1.2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
\qquad (5.2.1)
$$

$$
A_{RF} = \begin{array}{cc}
 & \begin{array}{cccccccccccccccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_{D1} & P_{D2} & P_{D3} & P_{D4} & P_{D5} & P_{D6} & P_{D7} & P_{D8} & P_{D9} & P_{D10} & P_{D11} & P_{D12} & P_{D13} \end{array} \\
\begin{array}{c} R\text{-}S_3 \\ R\text{-}S_4 \\ R\text{-}S_5 \\ R\text{-}S_6 \\ R\text{-}S_7 \\ R\text{-}S_9 \\ F_{P5} \end{array} &
\begin{pmatrix}
0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
\qquad (5.2.2)
$$

$$
\begin{array}{cc}
 & \begin{array}{cccccccccccccccccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_{D1} & P_{D2} & P_{D3} & P_{D4} & P_{D5} & P_{D6} & P_{D7} & P_{D8} & P_{D9} & P_{D10} & P_{D11} & P_{D12} & P_{D13} \end{array} \\
\begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \\ S_{11} \\ S_{12} \\ E_{6\text{-}7} \\ R\text{-}S_3 \\ R\text{-}S_4 \\ R\text{-}S_5 \\ R\text{-}S_6 \\ R\text{-}S_7 \\ R\text{-}S_9 \\ F_{P5} \end{array} &
\left[\begin{array}{cccccccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1.2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}
\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_{D1} \\ q_{D2} \\ q_{D3} \\ q_{D4} \\ q_{D5} \\ q_{D6} \\ q_{D7} \\ q_{D8} \\ q_{D9} \\ q_{D10} \\ q_{D11} \\ q_{D12} \\ q_{D13} \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

(5.2.3)

$$
\mathbf{q} = (2.1, 1.9, 0.75, 1.3, 1.0, 0.62, 0.48, 0.23, 0.32, 0.090, 0.81, 0.64, 0.16, 1.6, 0.81, 1.4, 0.34, 0.29, 0.29, 0.11)^T \qquad (5.2.4)
$$


(a) 元のフロー図

(b) 全域木の１つ

○：1辺が取り除かれた次数3の頂点
◍：セレクション条件が与えられている次数3の頂点

図16　非平面グラフとなる製品システムの全域木

そして、この得られた活動量ベクトル**q**と、環境フローを表す行列Bを式（2.1.3）に代入することで、システム全体の環境負荷ベクトル**β**を算出できる。

以上より、前報[15]の手法では解を求めることができなかった製品システムに本報の手法を適用することで、P_1からP_{D13}までの全プロセスの活動量を一意に求め、ライフサイクル全体にわたる環境負荷を算出できることが示された。

6. まとめ

筆者らの研究は、マトリックス法によるLCI分析の実施を容易にすることを目指し、グラフ理論に基づき係数行列を構築する手順を確立することを目的としている。前報[15]で提案された手法では、まず、新たに定義した5種類の基本要素を用いて、分析対象とする製品システムをグラフとして表現した。そして、プロセスと財フローとの関係性をグラフ理論によって定式化し、正則な係数行列を作成した。しかし、この手法は、分析対象の製品システムを抽象化して得られるグラフの平面性に依存し、任意の製品システムに完全に対応していない問題点があった。

そこで本報では前報[15]で提案された手法を次のように改良した。まず、全域木を用いて、製品システムのすべてのバランス式を一次従属にならないようにフロー図から求める手法を提示した。全域木は平面性によらずすべての連結グラフから作ることができる。これにより、製品システムの平面性に依存することなく任意の製品システムからでも正則な係数行列を作成することが可能になった。

今後、提案した手法を用いて分析対象の製品システムからのフロー図の構築、係数行列生成を自動的に行い、システム全体の環境負荷を推計できるソフトウェアを開発する予定である。係数行列の構築が自動的に行われることで、マトリックス法を用いたインベントリ分析が容易に実施できることが期待できる。

（平成22年5月5日受付、平成22年10月19日採択）

参照文献

1) R. Heijungs (1994): Ecological Economics, 10, 69-81
2) R. Heijungs, S. Suh (2002): The Computational Structure of Life Cycle Assessment, Published by Kluwer Academic Publishers, The Netherlands, 241
3) K. Halada (1996): Proceedings of 2nd International Conference on EcoBalance, Tokyo, 188-193
4) 盧偉哲, 泉聡志, 酒井信介 (2006): 日本LCA学会誌, 2 (3), 266-272
5) J.M.Gwak, M.R. Kim, T. Hur (2003): Journal of Cleaner Production, 11, 787-795
6) R. Heijungs, S. Suh (2006): Journal of Cleaner Production, 14(1), 47-51
7) S. Suh, G. Huppes (2005): Journal of Cleaner Production, 13(7), 687-697
8) R. Heijungs (2002): Proceedings of 5th International Conference on EcoBalance, Tsukuba, 77-80
9) Shinsuke Sakai, Koji Yokoyama (2002): Clean Technologies and Environmental Policy, 4, 72-78
10) 中島謙一, 長坂徹也, 原田幸明, 井島清 (2005): 日本金属学会誌, 69(2), 221-224
11) 酒井信介, 盧偉哲 (2005): LCA日本フォーラムニュース, 38, 10-13
12) 本藤祐樹 (2005): Journal of the Japan Institute of Energy, 84(11), 935-941
13) 総務省編 (2009): 産業連関表 総合解説編 平成17年, 経済産業調査会, 東京, 110-111
14) 中村愼一郎 (2000): Excelで学ぶ産業連関分析, エコノミスト社, 東京, 226
15) 福原一朗, 本藤祐樹 (2009): 日本LCA学会誌, 5 (1), 89-101
16) 伊坪徳宏, 田原聖隆, 成田暢彦 (2007): LCA概論, 社団法人産業環境管理協会, 東京, 12

17) 宮本重幸, 天川雅文 (1998): 日本エネルギー学会誌, 77 (10), 981-986
18) J. A. Bondy, U.S.R. Murty (2007): Graph Theory (Graduate Texts in Mathematics), Springer-Verlag, London, 651
19) F. Harary (1994): Graph Theory, Westview Press, Boulder, CO, USA, 284

付録

本報で用いたグラフ理論に特有の用語、定理について[18, 19]、以下に解説する。

1. 頂点、辺の定義

一般的にグラフGは頂点集合$V(G)$、辺集合$E(G)$及び、接続関数ψ_Gの組$(V(G), E(G), \psi_G)$で表される。頂点集合$V(G)=\{v_1, v_2, \cdots, v_\upsilon\}$は非空な有限集合であり、その元を$G$の頂点と呼ぶ。辺集合$E(G)=\{e_1, e_2, \cdots, e_\varepsilon\}$は$V(G)$と素な集合で、その元を$G$の辺と呼ぶ。頂点の数、辺の数を$V(G)$、$\varepsilon(G)$とする。$\psi_G$は、各辺に頂点の非順序対を対応させる接続関数である。グラフGの辺eが$e \in E(G)$であるとき、各辺はその両端の頂点対を用いて$\psi_G(e)=v_1v_2$（ただし、$v_1, v_2 \in V(G)$）のように表すことができる。このとき、v_1とv_2を辺eの端点であるという。

2. グラフの平面性

平面グラフとは、グラフのどの2つの辺も、それらが接続する頂点以外の場所では幾何学的に交差しないように描かれたグラフである。また平面グラフと同形であるグラフを平面的グラフという。

グラフが常に平面的であるための必要十分条件は、完全グラフK_5または完全二部グラフ$K_{3,3}$に縮約可能で位相同形な部分グラフを含まないことであり、*Kuratowski*の定理として知られている。

3. 道、閉路

隣接している頂点同士を辿る$\{v_1, e_1, v_2, e_2, \cdots, e_{n-1}, v_n\}$の頂点と辺の系列を歩道といい、辺と頂点の重複も許さない場合に道という。特に、始点と終点が同じ道$(v_1 = v_n)$を閉路という。この閉路によってグラフGの無限面と呼ばれる非有界な面が、新たな連結領域に分割されるとき、その連結領域の閉包をGの面と呼ぶ。その非有界な面を含むグラフの面集合を$F(G)$とし、面の数を$f(G)$とする。

辺の両端の頂点が等しい場合、その辺を自己ループという。また、2頂点間に複数の辺がある場合、これらの辺を多重辺という。特に、自己ループも多重辺も含まないグラフのことを単純グラフという。

4. 連結グラフと非連結グラフ

グラフGの頂点v_1とv_2に道があれば、v_1はv_2に連結される。そこで、任意の2頂点v_1、v_2に対し、「v_1はv_2に連結される」ことを「v_1はv_2と同じV_iに属する」と読み替えると、グラフGは$V(G)$の部分集合$V_1, V_2, \cdots, V_i, \cdots, V_n$で分割できる。この各集合からなる部分グラフ$G(V_1), G(V_2), \cdots, G(V_n)$をそれぞれグラフ$G$の成分と呼ぶ。成分を1つしか持たないグラフを連結グラフといい、成分が2つ以上のグラフを非連結グラフと言う。

5. 木、二分木、全域木

閉路を含まないグラフを林と定義し、連結な林を木と定義する。

（定理）グラフGに頂点がn個あるとき、次の命題は同値である。

(1) Gは木である。
(2) Gは閉路を持たず、辺がn–1本ある。
(3) Gは連結であり、辺がn–1本ある。
(4) Gは連結であり、すべての辺は橋である。
(5) Gの任意の2点を結ぶ道は1本だけである。
(6) Gは閉路を持たないが、新しい辺をどのように付け加えても1つの閉路ができる。

（定理終）

二分木とは、1つの辺が頂点で2本の辺に分かれていく木である。各頂点の次数は3を超えない。

連結グラフを与えられたとき、その任意のサイクルを選び1本の辺を除去しても残りのグラフは連結である。全域木（スパニングツリー）とは、このように連結グラフのサイクルの辺を除去することで作られる木のことを指し、1つの連結グラフから1つ以上の全域木を作ることができる。一般的には、n個の頂点とm本の辺を持つ連結グラフから全域木Tを作ると、除去される辺の本数は$\gamma(G) = m - n + 1$となり、$\gamma(G)$をグラフGの閉路階数と呼ぶ。

全域木Tに含まれていないグラフGの任意の辺を全域木Tに付加すると、定理の命題（6）より閉路が1つできる。このようにして、全域木に含まれていないグラフGの各辺を個別に付加してできる閉路全体の集合を、全域木Tに関連した基本閉路系と呼ばれている。基本閉路系の中の閉路はすべて異なり、閉路の個数はグラフGの閉路階数に等しい。