

「最適化学のすすめ－賢い決
め方のワークベンチ」別冊付録

(清水 良明 著、コロナ社 2010 刊行)

はじめに

以下は、分量の制限によって「最適化学のすすめ」(コロナ社 2010 刊行)に納めきれなかった内容を取りまとめたものである。特に PERT については、書籍では全く解説されていない。その他の部分についてもやむなく削ったものである。書籍と併せて活用して頂きたい。

2.2.2 機能構造モデル化手法—IDEF0

[1] IDEF0 モデル化における全般的な留意点

IDEF0 の記述方法は恣意的なため、モデル化に当たっては、目的(Purpose)、視点(View point)、範囲(Scope)を明確にすることが求められる。すなわち、モデル化の目的だけでなく、どの観点からものを見ているか、だれの立場で見ているかを明確にすることが特に必要とされる。また、モデル化の範囲を明確にし、記述しようとする内容が範囲に入っているかを絶えず確認する必要がある。こうした観点から IDEF0 ではダイアグラム以外にも、コンセプト、背景、モデルの説明文(Text)、説明図(Figure)、使用された用語の定義を行う用語集(Glossary)、これらの作成者とレビューを行う手順などを記載したドキュメント(文書)が作成される。モデル化作業では、ダイアグラムとこれらを合わせてセットで扱うようにしており、キットと呼ばれる。一般に IDEF0 での取り組みはダイアグラム表現に集中しがちであるが、多人数や基盤の違った人々の間でコミュニケーションするためには、システム分析や設計の過程で、共通の理解を得るベースとなるこれらのキットの整備が重要となる。

こうして作成したキットを評価し見直す仕組みはレビューと呼ばれ次のような役割を果たす4者によって実施される。

- ① オーサ(モデル作成者): IDEF0 モデルを作成する。
- ② コメンテータ(専門家): オーサが作成した資料を検証し、コメントを与える。
- ③ ライブラリアン(文書管理者): 作成文書の保管、コピー作成、配布、記録を行う。
- ④ リーダ(専門家): 対象モデルについての知識があり、オーサに情報を提供する。文章の評価のみでコメントはできない。

これらの役割は同一の人が複数担当しても良いが、レビューのキットサイクルを何回か通して、モデル化の精度を高めていくことが必要である。こうしたレビューや変更の履歴を確実に管理できる仕組みが IDEF0 によるモデル化の成否にとって重要である。IDEF0 を使用したシステム設計の進め方を図 AA.1 に示す。

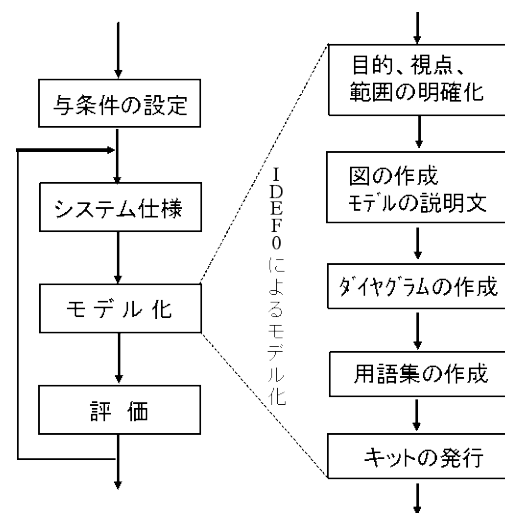


図 AA.1 IDEF0 を用いたシステム設計の流れ

[2] 生産システムにおける IDEF0 モデルの構築例

「**を工場生産する」という基本アクティビティを一段階、階層化したダイアグラムの中で「素材を加工する」を取り出し、加工の機能を詳細化すると図 AA.2 のようになる。加工工程は、加工セル管理アクティビティによって統制されるマシニングセンタ (MC) 加工、旋盤加工、研削加工からなる。研削加工には、MC 加工の一部の半加工品が投入される。三つの異なる加工工程 (MC、旋盤、MC と研削) を通った

部品がそれぞれ最終的に加工品として出力される。この中で MC 加工は、フレキシブル生産システム (FMS) ラインを構成しており、さらに必要に応じてこの FMS の各工程を詳細化するが、どれだけ詳細化するかは、モデル化の目的による。

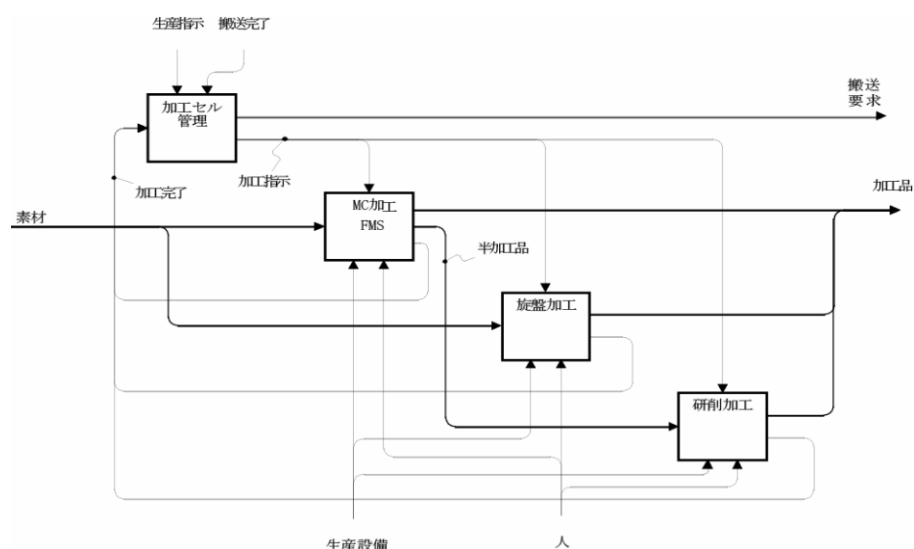


図 AA.2 「素材を加工する」を詳細化したダイアグラム

3.2 ペトリネットによる離散事象システムのモデル化

[1] 動的性質と構造的性質

初期マーキングに依存する動的性質は以下のような意味をもつ。

- (1) 可達性: ペトリネット N において、初期マーキング μ_0 から、マーキング μ に到達するトランジションの発火系列が存在するとき、マーキング μ は、ペトリネット N において可達であるという。なおペトリネット N において可達なすべてのマーキングの領域をマーキングの許容領域という。
- (2) 活性: トランジション t は、任意の可達なマーキングにおいて、 t を発火させるような発火系列が存在するとき活性であるという。ペトリネット N は、すべてのトランジションが活性であるとき、活性であるという。なおペトリネットが活性であるということは、どのような発火系列がとられても、デッドロックに陥らないことを保証するものである。
- (3) 有界性: ペトリネット N のすべての可達なマーキングにおいて、すべてのプレースのトークン数が自然数 k を超えないとき、 N は k -有界であるという。なおペトリネットが有界であるということは、どのような発火系列がとられても、モデル化したシステムのバッファやレジスタでオーバフローが起きないことを保証する。
- (4) パーシステンス: ペトリネット N は、任意の二つの発火可能なトランジションに対して、一方のトランジションの発火が他方を発火不可能にしなければ、パーシステントと呼ばれる。なおパーシステントネットのトランジションは、いったん発火可能になれば、それが発火するまで発火可能で

あり続ける。発火可能な二つのトランジションにおいて、一方のトランジションが他方のトランジションの発火を妨げるとき、二つのトランジションは競合しているという。

一方、初期マーキングに依存しない構造的性質は以下のような意味を持つ。

- (1) 反復性: ペトリネット N は、あるマーキング μ_0 と、すべてのトランジションが無限に生起するような μ_0 からの発火系列 σ が存在すれば、反復的であると呼ばれる。
- (2) 反復一致性: ペトリネット N は、あるマーキング μ_0 と、すべてのトランジションが少なくとも 1 回は発火する系列 σ が存在すれば、反復一致的であると呼ばれる。
- (3) 構造的有界: ペトリネット N は、任意の初期マーキング μ_0 に対して有界であれば、構造的に有界であると呼ばれる。
- (4) 保存性: ペトリネット N は、任意の初期マーキング μ_0 からの可達なマーキングについて、すべてのプレース p に入るトークンの重みつき総和が一定であれば、保存的であるという。

[2] オーダーエントリー・オペレーションモデル

以下では、ペトリネットのもつ多様な特徴の一例として経営問題における MIS(Management Information Systems)活動を単純化したモデルを例示する。

[例 AA.1] 図 AA.3 は、簡単な小売注文における顧客と売り手の振る舞いを表している。注文がされるという事象が生起するためには、売り手は「注文を受けられる」という状態に、顧客は「注文できる」という状態になければならない。この事象が起こると、売り手は「注文を受けた」という状態に、客は「注文をした」という状態に入る。このモデル化でわかるようにペトリネットは、売り手と顧客の活動という、二つの独立した活動の関連を容易に表現できる。

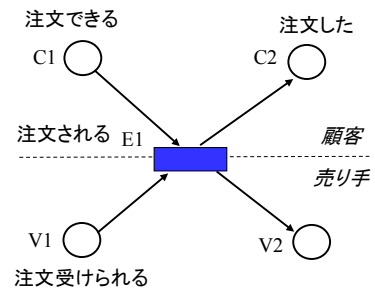


図 AA.3 小売注文モデル

この図は、役割一分割と呼ばれるクラスのペトリネットであって、それぞれが独立した役割を果たすサブシステムに分割された状況を、ある役割に属する状態に対しては、同一のアルファベットを用いて表現している(例えば、V1, V2 は売り手の C1, C2 は顧客側の役割状態を表す)。一つの役割は一つのサブシステムを表わし、状態集合のうち、一つの状態だけがいつでもホールドしているようになっている。このことは、ネットの各事象が同一の役割に属する二つ以上の入出力状態を持たないことを意味する。二つ以上の役割が共通の事象(E1)を分け合っているときは、これらの役割が互いに相互作用を及ぼしていることに対応する。

また発火規則は、「両者の準備ができるまでは注文がなされない」という事実をうまくモデル化している。さらに、「注文されない」という相補的な選択を事象として付加することで、図 AA.4 に示すようにネットの選択的行動を表すことができる。ここで注文がなされるか否かを最終的に決定するのは、売り手ではなく顧客である、ということも仮定されている。

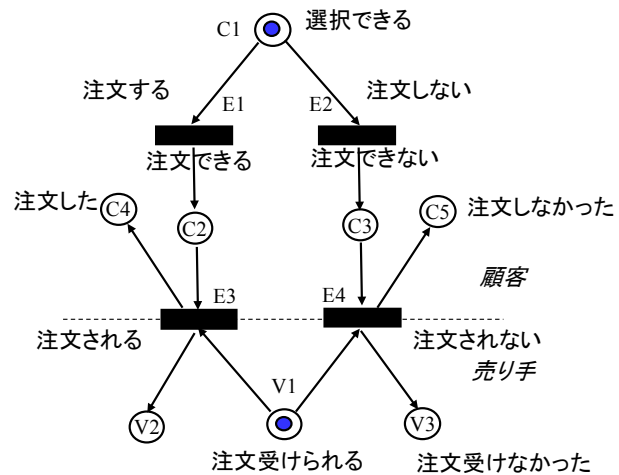


図 AA.4 選択的行動を追加したモデル

いま客が、図 AA.4 の状態 C1 にあるとき、コンフリクト規則により、E1かE2かのどちらかの事象だけが生起することになる。これ

らの事象に対して、他の役割(売り手)は関係していないので、客側の自由な決定がモデル化されている。一方、事象 E3 と E4 は、売り手と客という二つの役割で分け合っているため、売り手は状態 V1 にいても、自由な決定を行なえない。この形態において、売り手側のサブシステムと無関係な顧客の自由な決定によって、事象 E3 と事象 E4 は完全に支配されることになり、売り手の決定は顧客に依存するという仮定をうまく表現できる(例えば、客が E2 でなく E1 を選ぶと、C2 と V1 がホールドし、発火規則に従って E3 が生起することになる)。

[例 AA.2] ペトリネットは、経営上の情報に依存する意思決定過程を表現するためにも用いられる。いまある品物についての売り値を上げたとする。そしてその後、マネージャは次の三つのうちのどれか一つのメッセージを受け取ったとする。

{①収益が増した, ②収益が変わらない, ③収益が減った。}

そして受け取った情報に基づいて、マネージャは、次の三つの行動のうちどれか一つを決定しなければならないとする(値の上げ下げ幅については無視する)。

{①値を上げる, ②値をそのまま保つ, ③値を下げる。}

図 AA.5 は、このマネージャが直面している意思決定過程をモデル化したものである。収益が減ったときには(CC1), 意思決定者の状態は DM2 となり、以前の不首尾を取り戻そうとして、値を下げることになる。また、もし収益が増加していたら(CC2), 意思決定者の状態は DM3 となり、この戦略がどれくらい進められるかを知らうと、値上げすることになる。収益が変わらないときは(CC3),

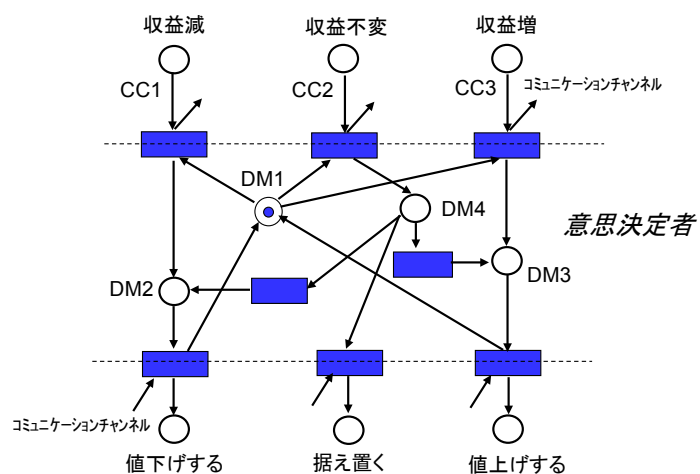


図 AA.5 経営上の情報に依存する意思決定過程モデル

意思決定者の状態はDM4となり、そこで現在の値を上げるか下げるか、またはそのまま据え置くかの自由な選択がなされる。情報に依存して決まる部分と自由判断に任せられている部分が混在する中で、後者がどのように分離して表現され得るかをこの例は示している。

[例 AA.3] 意思決定の結果と基礎となる情報との間に、既知の確率的関係があるような意思決定過程を考える。例えば図AA.6は、先のオーダーエントリーモデルの

例での意思決定過程を拡張したものである。先には、注文の決定は客の「自由選択」としてモデル化されていた。ここでは値をつけるという売り手の行動によって客は影響を受けて選択をするものと仮定する。売り手による価格の決定に対応して、「注文がなされる」という事象の生起確率を各々(P1, P2, P3)とする客の三つの自由選択の状態(C2, C3, C4)がモデル化されている。これらの三つの確率の差が大きければ大きいほど、売り手の客に及ぼす影響が大きいことになる。

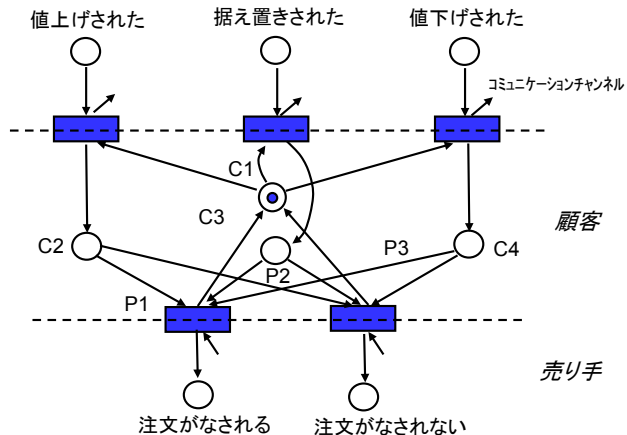


図 AA.6 確率的関係がある意思決定過程モデル

3.1 動的システムのモデル化と解析

[1] 遷移行列の性質

遷移行列 $\Phi(t, t_0)$ について以下の性質が知られている。

$$\text{推移性: } \Phi(t, t_f)\Phi(t_f, t_0) = \Phi(t, t_0)$$

$$\text{反射性: } \Phi(t, t_0) = \Phi(t_0, t)^{-1}$$

$$\text{共役性: } \Psi(t, t_0) = \Phi(t_0, t)^T$$

ここで、 $\Psi(t, t_0)$ は共役な方程式、 $d\Psi(t, t_0)/dt = -\Psi(t, t_0)A^T(t)$ 、 $\Psi(t_0, t_0) = I$ の解である。

3.3 ニューラルネットワークによるモデル化

[1] 動的モデリング

NN による動的モデリングとしては、化学プロセスにおける複雑な現象のモデル化^{AA1)}や、重合反応プロセスにおいて定常状態でのデータから動的なモデル化を行ったもの^{AA2)}などがある。またリカレント型ネットワークを利用したもの^{AA3)}や、CSTR の pH 調整のためのモデルベース制御において、NN モデルに基づく構成の方が ARMA (Average Regress on Multiple Analysis) による場合より優れていたという結果を示したもの^{AA4)}などが報告されている。

さて、NN において利用できる動的データは、操業データなどの一定のサンプリング間隔毎に得られる入出力変数間の関係を表す離散列である。これを用いたモデル化は過去及び現在の入出力関係から将来の出力値を予測するのが一般的である。これを数学的に表現すれば

$$\left\{ \begin{array}{l} y(k+1) = f_1(y(k), y(k-1), \dots, y(k-n), u(k), \dots, u(k-n)) \\ y(k+2) = f_2(y(k), y(k-1), \dots, y(k-n), u(k), \dots, u(k-n)) \\ \text{(AA.1)} \\ \dots\dots\dots \\ y(k+m) = f_m(y(k), y(k-1), \dots, y(k-n), u(k), \dots, u(k-n)) \end{array} \right.$$

なる関係を表すベクトル関数 f を同定せよということになる。ここで u は入力変数を、 y は出力変数を表す。また現在の時刻 k を基準として、 n は過去方向へのステップ数を、 m は将来方向へのステップ数を表す。

動的モデリングに用いられてきた NN の構造は、

(a) Feedforward (FF) 型

(b) Feedback (FB) 型あるいは外部リカレント型 (External recurrent network)

に大別される。このうち FF 型は、最も一般的に用いられる構造で、静的パターンマッチングには適するが、時間相関が反映されにくいので、1ステップ先の予測には対応できるが、長期予測には不適といわれている。一方、FB 型は、データ間に相関を有する場合に適するので、化学プロセスのように一定の物理・化学法則に依存して変化し、現在の出力が過去の入力によっても影響されるような現象のモデル化には適しているといえる。一方、データにランダム性が強い場合はむしろ FF 型の方がよいとされている。

またデータの取得法や前処理の方法も、NN によるモデル化が成功するかどうか大きな影響を与えるのが通常である。取得法は、

- (1) 過去の操業データなどを用いて1バッチ間や一定の設定期間内のデータを一括して利用する、
- (2) 観測窓を設けその区間内のデータを利用し、観測窓は順次ずらして学習結果を更新していく、に大別される。後者はオンラインで同定する場合の基本形である。またデータとしてはできるだけ生の値を使うのがよいという意見もあるが、BP ネットワークにおける学習効率の面からは、必要に応じて規格化した方がよい場合も少なくない。値自身も、

- (1) 観測値そのものを用いる、
- (2) 観測値の偏差を用いる ($\Delta y(k+1) = y(k+1) - y(k)$),

場合があるが、どちらにすべきかは一概にいえずモデル化の利用目的に依存する。

さらにデータの取得や予測範囲に関して、

- (1) 過去のどの時点まで遡ったデータを使うか、
- (2) 将来のどの時点まで予測するか、といった個別の問題があるが、実際には、
- (3) 将来のある時点まで予測するには過去のどの時点まで遡ったデータが必要か、

といった一体化した検討が必要となる。この一つの解決策としてデータにランダム性が強い経済予測に応用された例として、フラクタル次元を利用するという考え方^{AA5)}がある。しかし先と同様、対象プロセス毎にモデリング能力や適用範囲の検証が必要となる。

ところで、

- (1) 入力数(参照データ数)が多いほど推定精度は向上する、

(2) より将来になるほど予測精度は低くなる、
といった知見は自然に理解されるものと思う。

また動的モデリングにおいては、全くのブラックボックスモデルを使うより、現象を多少とも解析的に理解できるモデルの方が、安心感や不測の事態への対応力は高いと考えられる。こうした点からも部分的に物理モデルを併用したり^{AA6)}、プロセス技術者の経験的モデル化法を取り込こんだりした形で NN の構成を考えることも、実際の適用では重要と考えられる。

4.2 線形計画法(LP)

[1] 双対問題に関わる定理の証明

[定理 AA.1] 原問題に有界な最適解があるならば、その双対問題にも有界な最適解が存在し、その目的関数の最適値は等しい。

(証明) $\mathbf{c}^T \mathbf{x}^* \leq \mathbf{b}^T \boldsymbol{\pi}$ において、最適時の基底変数は $\mathbf{x}_B^* = \mathbf{B}^{-1} \mathbf{b}$ で、このときの $\boldsymbol{\pi}$ を $\boldsymbol{\pi}^*$ と書くと、 $\mathbf{c}^T \mathbf{x}^* = \mathbf{c}_B^T \mathbf{x}_B^* = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} = \mathbf{b}^T \boldsymbol{\pi}^* \leq \mathbf{b}^T \boldsymbol{\pi}$ という関係が得られる。

[定理 AA.2] 原問題とその双対問題の両方に実行可能解が存在するならば、両方に有界な最適解が存在し、その目的関数の最適値は等しい。

(証明) 両方の任意の実行可能解を $\mathbf{x}, \boldsymbol{\pi}$ とすれば、 $\mathbf{c}^T \mathbf{x} \leq \boldsymbol{\pi}^T \mathbf{A} \mathbf{x} \leq \mathbf{b}^T \boldsymbol{\pi}$ が成立する。これより直ちに、 $\text{Max } \mathbf{c}^T \mathbf{x} \leq \text{Min } \mathbf{b}^T \boldsymbol{\pi}$ が導かれる。従ってそれぞれの最適値は有界となる。ここで[定理 AA.1]を適用すれば、 $\text{Max } \mathbf{c}^T \mathbf{x} = \text{Min } \mathbf{b}^T \boldsymbol{\pi}$ が成り立つ。

[定理 AA.3]

(1) 原問題が上に有界でない(最適)解 ($\text{Max } z = \infty$)をもつならば、その双対問題は実行可能解をもたない。この逆もなりたつ。

(2) 双対問題が下に有界でない(最適)解 ($\text{Min } z = -\infty$)をもつならば、その原問題は実行可能解をもたない。この逆もなりたつ。

(1)の証明 : 原問題が上に有界でない解をもつということは、(初期基底から出発して z を増加させるようにピボット操作を続けてみても、) どのような $\boldsymbol{\pi} \geq \mathbf{0}$ も最適条件、 $\mathbf{A}^T \boldsymbol{\pi} - \mathbf{c} \geq \mathbf{0}$ を満たすことができないことを意味する。このことは双対問題の条件を満たす可能解が存在しないことにほかならない。

逆に、双対問題が可能解をもたないことは、 $\mathbf{A}^T \boldsymbol{\pi} - \mathbf{c} \geq \mathbf{0}, \boldsymbol{\pi} \geq \mathbf{0}$ を満足するような $\boldsymbol{\pi}$ が存在し得ないことを意味する。これを原問題から見れば、永遠に解を改善できることに他ならない。すなわち原問題は有界でない解をもつことになる。

(2)の証明もこれとおなじ論法で行える。

[2] 漸進的線形計画法(PROGRESSIVE LP; PROLP)の求解手順

PAPA (Pivot And Probe) 法^{AA7)}は、(p.AA.1)のような原点 $\mathbf{x} = \mathbf{0}$ が原問題の可能解の一つとなることや、LE 不等式のみで構成される問題だけにおいて成立する条件を前提としてアルゴリズムが与えられている。種々の種類の制約式が混在する一般的な場合に拡張した方法は漸新的線形計画

法(PROLP)^{AA8, AA9}として提案されている。以下では PROLP の原型となる PAPA 法の考え方を示す。

$$(p.AA.1) \quad \text{Max} \sum_{j=1}^n c_j x_j \quad \text{subject to} \quad \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad \forall i \in I, \quad x_j \geq 0$$

ここで、 $I = \{i|1, \dots, m\}$, $J = \{j|1, \dots, n\}$ で $b_i \geq 0$ ($\forall i \in I$)とする。このとき、原点 $\mathbf{x}=\mathbf{0}$ は明らかに (p.AA.1)の一つの可能解を与える。

最初に、以下に定義する緩和問題の解の有界性を常に保証するため、 $m+1$ 番目に正規化条件 $\sum_{j \in J} x_j \leq M$ を追加する。ここで、 M は適当な大数とする。また添え字集合 K_j を $K_j = \{i | \text{Min } b_i/a_{ij}, \forall j \in J, a_{ij} > 0\}$ のように定義して、

$$F(1) = \bigcup_{j \in J} K_j \cup \{m+1\} \tag{AA.2}$$

で与えられる制約式のみから構成される緩和問題を考える。

一般に k 段階での制約式の添字集合を $F(k)$ で表すとき、この段階での緩和問題は、次のように与えられる。

$$(p.AA.2) \quad \text{Max} \sum_{j=1}^n c_j x_j \quad \text{subject to} \quad \sum_{j \in J} a_{ij} x_j \leq b_i, \quad (\forall i \in F(k)), \quad \mathbf{x} \geq \mathbf{0}$$

$F(k)$ は I の部分集合 ($F(k) \subseteq I$)であり、制約式は順次追加されていくため、許容集合間には $X \subset X(k) \subset X(k-1)$ となる関係が成立する。このとき、正規化条件により、(p.AA.1)が解を持つとき (p.AA.2)は必ず有界な解を持つので、これを $\mathbf{x}^d(k)$ で表す。

次段階での実行可能領域は、 k 段階で $F(k)$ の要素でない制約式 $i \in F^N(k)$ の中から適当な制約式を追加して決める。この決め方としては最適解が位置しそうな方向に実行可能領域を最も制限するように選ぶのが効率的である。実際には、(p.AA.1)の可能解が一つ既知でありこれを \mathbf{x}^p と表せば、ベクトル $\mathbf{x}^d(k) - \mathbf{x}^p$ の線上で、できるだけ \mathbf{x}^p 側に近い点をよぎる制約式を選べば良い。この Probe (貫通)操作のために、 $F^N(k)$ 中で $\mathbf{x}^d(k)$ を満足しないものに対するベクトル $\mathbf{x}^d(k) - \mathbf{x}^p$ の貫通比 λ_i を次式のように計算する。

$$\lambda_i = (b_i - \mathbf{a}_i^T \mathbf{x}^p) / (\mathbf{a}_i^T \mathbf{x}^d(k) - \mathbf{a}_i^T \mathbf{x}^p), \quad \forall i \in L(k) \tag{AA.3}$$

ここで、 $L(k) = \{i | \mathbf{a}_i^T \mathbf{x}^d(k) > b_i, \forall i \in F^N(k)\}$ である。そして i が小さい順に適当個数 (p とする)選んで決められる制約式集合、 $H(k) = \{i_1, i_2, \dots, i_p\}$ を追加して次段階の制約式集合を $F(k+1) = F(k) \cup H(k)$ とする。

$\mathbf{x}^d(k)$ を満たさない制約式が存在する限り、 $0 \leq \lambda_i < 1$ となるので、以下、同様の手順を繰り返しているとき、 $\lambda_i = 1$ ($\forall i \in H(k)$)となるか、或は $L(k) = \{\emptyset\}$ となった段階で最適解が与えられることになる。

ところで各段階での Probe 操作によって与えられる貫通点 (Piercing point),

$$\underline{\mathbf{x}}(k) = (1 - \lambda_i) \mathbf{x}^p + \lambda_i \mathbf{x}^d(k), \quad \forall i \in H(k) \tag{AA.4}$$

が可能解 ($\underline{\mathbf{x}}(k) \in X$)となる事は容易に示されるので、原点以外に $\underline{\mathbf{x}}(k)$ を別の Probe 操作の基準点とすることもできる。また対応する目的関数値間には

$$c^T x^d(k) \geq c^T x^* \geq c^T \underline{x}(k) \tag{AA.5}$$

なる関係が成立するので、任意の段階での $x(k)$ を上式で評価される近似解と見なすこともできる。ここで x^* は原問題の最適解である。

[例 AA.4]表 AA.1 に示すような係数行列を持つLP, $\sum_{j \in I} a_{ij} x_j \leq b_i \ (i=1, 2, \dots, 7)$ について、上述

の手順を示す(図 AA.7 参照)。

まず、 $j=1, 2$ に対する K_j より制約式 2, 1 がそれぞれ選ばれ、 $F^1(1)=\{1, 2\}$ となる(正規化条件はここでは無視した)。そして(p.AA.2)より、 $x^d(1)=(5.0, 5.0)$ (図中の点 a)と求められた。次に原点から $x^d(1)$ への Probe 操作 (b が貫通点) によって、まず $F^1(1)=\{3, 4, 5, 6, 7\}$ の中から制約式 3 が追加された($H(1)=\{3\}$)。そして $F(2)=\{1, 2, 3\}$ として(p.AA.2)へ戻り、 $x^d(2)=(2.0, 3.5)$ (点 c) が得られた。

Eq. No.	a_{i1}	a_{i2}	b_i
1	5	-3	10
2	-1	2	5
3	7	4	28
4	1	2	8
5	3	2	15
6	1	3	12
7	0	1	3
Obj.	1	1	Max

さらに $L(2)=\{4, 6, 7\}$ に対して、ここで基準点1(原点)及び基準点2 ($x(1)$ すなわち点 b)からの Probe 操作により、 $H(2)=\{7, 4\}$ と選ばれ、 $F(3)=\{1, 2, 3, 7, 4\}$ に対して、 $x^d(3)=(2.4, 2.8)$ (点 e) が得られた。以上で $L(3)=\{\varnothing\}$ となるので、結局7式中5式を用いて最適解が得られたことになる。

これらの方法は、新規に問題を解く場合以外にもシステム環境の変化などによって、このような多少変更された問題を頻繁に繰り返し解く必要がある場合には先の感度解析の利用との使いわけによって特に効果を発揮できる。一例として PROLP の線形計画モデル予測制御への適応例 AA10)があるように現実的な最適化に向けた解法といえる。

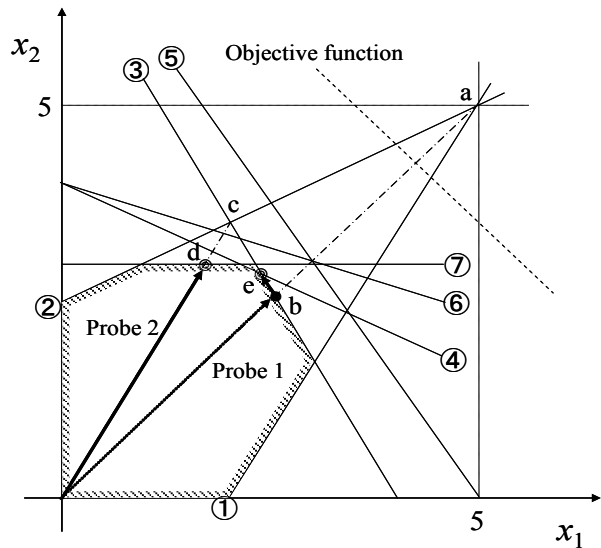


図 AA.7 例題の求解

[3] 内点法の求解手順

ここでは、次の標準形の線形計画問題を対象として考える。

$$(p.AA3) \quad \text{Min } c^T x \quad \text{subject to } Ax = b, \quad x \geq 0$$

この問題の実行可能領域を X と表すとき、

$$Ax = b, \quad x > 0 \tag{AA.6}$$

を満たす点 x を実行可能領域 X の内点 (interior point) と呼ぶ。

いま、ある点 $x^{(k)}$ が実行可能領域 X の内点であるとき、目的関数値を改善する合理的な方法は、 $c^T x$ の減少率が最大となる方向に動くことである。しかし、この方法は点 $x^{(k)}$ が X の中央付近にあると

きには効果的であるが、 $\mathbf{x}^{(k)}$ が X の境界の近くにあるときには極く少しか動くことができず、目的関数値を十分に減少させることは期待できない。

そこで、まず点 $\mathbf{x}^{(k)}$ を領域の境界から十分離れた点に移すような変換を行ってから、変換先の空間において目的関数値が最も減少する方向に移動し、その後に変換を行って、得られた点をもとの空間に戻せばよいという考え方ができる。いま、現在の点 $\mathbf{x}^{(k)}$ に対して定義された対角行列

$$D^{(k)} = \begin{bmatrix} x_1^{(k)} & 0 & \cdots & 0 \\ 0 & x_2^{(k)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_n^{(k)} \end{bmatrix}$$

を用いて、アフィン(線形)変換

$$\mathbf{y} = (D^{(k)})^{-1} \mathbf{x} \tag{AA.7}$$

を行えば、変数 \mathbf{y} の空間上では点 $\mathbf{A}^{(k)}$ は常に点 $(1, 1, \dots, 1)$ に移される。

そこで式(AA.7)を適用すれば、元の問題は次のように表現できる。

$$(p.AA.4) \quad \text{Min } \mathbf{c}^{(k)T} \mathbf{y} \quad \text{subject to } \mathbf{A}^{(k)} \mathbf{y} = \mathbf{b}, \mathbf{y} \geq \mathbf{0}$$

ここで、 $\mathbf{c}^{(k)} = D^{(k)} \mathbf{c}$, $\mathbf{A}^{(k)} = A D^{(k)}$ である。また、実行可能領域内で目的関数値の減少率が最大となる方向を $\mathbf{d}^{(k)*}$ とすれば、 $\mathbf{d}^{(k)*}$ はベクトル $-\mathbf{c}^{(k)}$ を部分空間 $\{\mathbf{y} | \mathbf{A}^{(k)} \mathbf{y} = \mathbf{0}\}$ へ射影して得られるベクトル方向になるので

$$\mathbf{d}^{(k)*} = -\mathbf{P}^{(k)} \mathbf{c}^{(k)} = -\mathbf{P}^{(k)} D^{(k)} \mathbf{c} \tag{AA.8}$$

と書くことができる。ここで、 $\mathbf{P}^{(k)}$ は部分空間 $\{\mathbf{y} | \mathbf{A}^{(k)} \mathbf{y} = \mathbf{0}\}$ への射影行列であり、行列 A の階数が m のときには、

$$\mathbf{P}^{(k)} = I - A^{(k)T} (A^{(k)} A^{(k)T})^{-1} A^{(k)} \tag{AA.9}$$

で与えられる。ところで、 $\mathbf{d}^{(k)*}$ は元の変数 \mathbf{x} の空間では、 $\mathbf{d}^{(k)} = D^{(k)} \mathbf{d}^{(k)*}$ となるので、結局、 $\mathbf{x}^{(k)}$ における探索方向は

$$\begin{aligned} \mathbf{d}^{(k)} &= -D^{(k)} \mathbf{P}^{(k)} \mathbf{c}^{(k)} \\ &= -D^{(k)} (I - D^{(k)} A^T (A D^{(k)2} A^T)^{-1} A D^{(k)}) D^{(k)} \mathbf{c} \\ &= -D^{(k)2} (I - A^T (A D^{(k)2} A^T)^{-1} A D^{(k)2}) \mathbf{c} \\ &= -D^{(k)2} (\mathbf{c} - A^T \mathbf{w}^{(k)}) \end{aligned} \tag{AA.10}$$

となる。ここで、 $\mathbf{w}^{(k)} = (A D^{(k)2} A^T)^{-1} A D^{(k)2} \mathbf{c}$ は、双対変数(Dual variable)ベクトルと呼ばれる。また、 $\mathbf{r}^{(k)} = \mathbf{c} - A^T \mathbf{w}^{(k)}$ で与えられる n 次元ベクトル $\mathbf{r}^{(k)}$ はシンプレックス法の相対コスト係数に対応している。

そして $\mathbf{c}^T \mathbf{x}$ の線形性より、半直線 $\mathbf{x}^{(k)} + t \mathbf{d}^{(k)}$ ($t \geq 0$) 上でこの値は単調に減少する。そしてここでの最大ステップ幅は次式によって与えられる。

$$t_{max} = \text{Min } \{x_j^{(k)} / (-d_j^{(k)}) | d_j^{(k)} < 0\} \tag{AA.11}$$

しかし、 t_{max} まで移動した点は実行可能領域の境界上に位置する。そこで次の反復点が常に実行可能領域の内点にとどまるようにするために、 $0 < \alpha < 1$ であるような定数 α を用いて、実際には、次式のように決める

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha t_{max} \mathbf{d}^{(k)} \quad (\text{AA.12})$$

以上のアフィン変換法の手続きをまとめれば次のようになる。

ステップ 1: 初期解として式(AA.6)を満たす点 $\mathbf{x}^{(0)}$ を選び、 $k:=0$ とおく。

ステップ 2: 行列 $\mathbf{D}^{(k)}$ と双対変数 $\mathbf{w}^{(k)}$ を求め、これらを式(AA.10)に代入して、 $\mathbf{d}^{(k)}$ を求める。

ステップ 3: 式(AA.11)より最大ステップ幅 t_{max} を求める。

ステップ 4: 式(AA.12)により次の反復点 $\mathbf{x}^{(k+1)}$ を求める。 $k:=k+1$ とおいて、ステップ 2 へ戻る。

上述のアルゴリズムのステップ 1 において、式(AA.6)を満たす初期解 $\mathbf{x}^{(0)}$ を求めることは一般に自明ではない。そこで初期解となる内点が容易に得られ、さらにそこでの最適解が元の問題の最適解に等しくなるような補完的な LP を解くことを考える。具体的には、 $x_j^{(0)} > 0$ ($j=1, 2, \dots, n$) を任意に選び、点 $\mathbf{x}^{(0)}$ における制約条件式 $\mathbf{Ax}=\mathbf{b}$ の残差を次式から計算する。

$$s_i = b_i - (a_{i1}x_1^{(0)} + a_{i2}x_2^{(0)} + \dots + a_{in}x_n^{(0)}), \quad i=1, 2, \dots, m \quad (\text{AA.13})$$

ここで、もし $s_i=0$ がすべての i に対して成立しているならば、 $\mathbf{x}^{(0)}$ は内点になるから、これを初期解として直ちに上述の手順を始められる。一般にはそうではないので、 $n+1$ 個の変数 $(\mathbf{x}^T, \mathbf{y})=(x_1, \dots, x_n, y)$ をもつ次の補完的 LP を考える。

$$(\text{p.AA.5}) \quad \text{Min } \mathbf{c}^T \mathbf{x} + M y \quad \text{subject to } \begin{cases} \mathbf{Ax} + \mathbf{sy} = \mathbf{b} \\ \mathbf{x} \geq 0, \quad \mathbf{y} \geq 0 \end{cases}$$

ここで、 y は 2 段階シンプレックス法の場合と同様、人為変数 (Artificial variable)、 M は正定数、 \mathbf{s} は式(AA.13)の s_i を成分とする m 次元定数ベクトルである。 s_i の定義より、 $(\mathbf{x}, \mathbf{y})=(\mathbf{x}^{(0)}, 1)$ は (p.AA.5) の内点になっているので、それを初期解としてこの問題に内点法を適用することができる。そして、定数 M が十分大きいとき、(p.AA.3) と (p.AA.5) は等価となることがわかる。

ところで内点法の計算労力の大半は、各反復におけるベクトル $\mathbf{d}^{(k)}$ の計算、とりわけ双対変数のベクトル $\mathbf{w}^{(k)}$ に費やされる。実際、 $\mathbf{w}^{(k)}$ をどのように計算するかは内点法の計算効率を左右する最も重要な要因となる。

4.3 非線形計画法 (NLP)

[1] 共役傾斜法の基本定理

[定理 AA.4] $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n$ が互いに共役ならばこれらは一次独立となる。

(証明) $\mathbf{d}_i (\neq \mathbf{0})$ ($i=1, \dots, n$) は一次独立でないと仮定する。すると全ては 0 でない定数 c_i に対して、 $\sum_i c_i \mathbf{d}_i = \mathbf{0}$ が成立する。いま $c_s \neq 0$ とし、この両辺に $\mathbf{d}_s^T \mathbf{A} =$ を左から乗じると $\mathbf{d}_s^T \mathbf{A} \mathbf{d}_s = 0$ となる。 \mathbf{A} は正定値行列であるので $\mathbf{d}_s = \mathbf{0}$ でなければならない。これは仮定と矛盾する。よって \mathbf{d}_i ($i=1, \dots, n$) は一次独立である。

[定理 AA.5] 2次形式 $f(\mathbf{x}) = a + \mathbf{b}^T \mathbf{x} + (1/2) \mathbf{x}^T \mathbf{A} \mathbf{x}$ の最小点は、任意の出発点 $\mathbf{x}^{(0)}$ から、各ベクトル \mathbf{d}_i を、探索の方向として1回ずつ使って、有限回の降下ステップの計算によって求められる。またこのとき \mathbf{d}_i を使う順番は関係しない。

(証明) $\mathbf{g}^{n+1}(\mathbf{x})$ は一次独立な $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n$ と直交するので、恒等的に $\mathbf{0}$ でなければならない。 $\mathbf{g}^{n+1}(\mathbf{x}^{n+1}) = \mathbf{0}$ すなわち \mathbf{x}^{n+1} で最適となる。

[2] 一般縮小勾配法のアゴリズム

P. Wolfe による縮小勾配法 (reduced gradient method) に基づいて、J. Abadie と J. Carpenter は、目的関数のみならず制約式も非線形の場合を取り扱える一般縮小勾配法 (Generalized Reduced Gradient method; GRG 法)^{AA11)} を提案した。

ここでは、次のような上下制限約のある非線形計画問題を考える。

$$(p.AA.6) \quad \text{Min } f(\mathbf{x}) \quad \text{subject to} \quad \begin{cases} \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U \end{cases}$$

ここで \mathbf{x} の次元を n , \mathbf{h} の次元を m とする。

いま制約条件を満たす任意の点 \mathbf{x} において、 m 個の基底変数のベクトル \mathbf{x}_B と $n-m$ 個の非基底変数のベクトル \mathbf{x}_N に対応する $\mathbf{x}^L, \mathbf{x}^U$ の分割を $\mathbf{x}^L = (\mathbf{x}_B^L, \mathbf{x}_N^L), \mathbf{x}^U = (\mathbf{x}_B^U, \mathbf{x}_N^U)$ とすれば、制約条件は次のように表わされる。

$$\mathbf{h}(\mathbf{x}_B, \mathbf{x}_N) = \mathbf{0} \tag{AA.14}$$

$$\mathbf{x}_B^L \leq \mathbf{x}_B \leq \mathbf{x}_B^U, \quad \mathbf{x}_N^L \leq \mathbf{x}_N \leq \mathbf{x}_N^U \tag{AA.15}$$

また、行列 $\nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x})$ の基底行列 (basis matrix), $\mathbf{B}(\mathbf{x}) = \nabla_{\mathbf{x}_B} \mathbf{h}(\mathbf{x})$ は正則であるという非退化の仮定 (nondegeneracy assumption) をおく。このとき、もとの問題 (p.AA.6) は、非基底変数 \mathbf{x}_N の上限 \mathbf{x}_N^U と下限 \mathbf{x}_N^L のみを制約とする次の縮小問題 (reduced problem) に帰着される。

$$(p.AA.7) \quad \text{Min } F(\mathbf{x}_N) = f(\mathbf{x}_B(\mathbf{x}_N), \mathbf{x}_N) \quad \text{subject to} \quad \mathbf{x}_N^L \leq \mathbf{x}_N \leq \mathbf{x}_N^U$$

ここで関数 $F(\mathbf{x}_N)$ は縮小目的関数と呼ばれ、その縮小勾配 (reduced gradient) $\nabla F(\mathbf{x}_N)$ は、次式で計算される。

$$\nabla F(\mathbf{x}_N) = \nabla_{\mathbf{x}_N} f(\mathbf{x}) + \nabla_{\mathbf{x}_B} f(\mathbf{x}) (\partial \mathbf{x}_B(\mathbf{x}_N) / \partial \mathbf{x}_N) \tag{AA.16}$$

また同様に式 (AA.14) より

$$d\mathbf{h}(\mathbf{x}_B(\mathbf{x}_N), \mathbf{x}_N) = \nabla_{\mathbf{x}_B} \mathbf{h}(\mathbf{x}) d\mathbf{x}_B + \nabla_{\mathbf{x}_N} \mathbf{h}(\mathbf{x}) d\mathbf{x}_N = \mathbf{0} \tag{AA.17}$$

となるので、 $\nabla_{\mathbf{x}_B} \mathbf{h}(\mathbf{x}) = \mathbf{B}(\mathbf{x})$ が \mathbf{x} で正則であることを考慮すれば

$$\partial \mathbf{x}_B(\mathbf{x}_N) / \partial \mathbf{x}_N = -\mathbf{B}^{-1}(\mathbf{x}) \nabla_{\mathbf{x}_N} \mathbf{h}(\mathbf{x}) \tag{AA.18}$$

が得られる。これを式 (AA.16) に代入すれば、縮小勾配の公式

$$\nabla F(\mathbf{x}_N) = \nabla_{\mathbf{x}_N} f(\mathbf{x}) - \nabla_{\mathbf{x}_B} f(\mathbf{x}) \mathbf{B}^{-1}(\mathbf{x}) \nabla_{\mathbf{x}_N} \mathbf{h}(\mathbf{x}) \tag{AA.19}$$

が導かれる。

このように求められた縮小勾配を利用して縮小問題(p.AA.7)を上下限を考慮した降下法で解くことにする、いま現在の点 \mathbf{x}_N^i における方向ベクトル \mathbf{d}_N^i が $\nabla F(\mathbf{x}_N^i)\mathbf{d}_N^i < 0$ を満たすように定められたとする。このときのステップ幅 t^i を、 $\mathbf{x}_N^i + t\mathbf{d}_N^i$ が上下限制約を満たすように t に制限を加えた 1 次元探索問題、

$$(p.AA.8) \min_{t>0} F(\mathbf{x}_N^i + t\mathbf{d}_N^i) = f(\mathbf{x}_B(\mathbf{x}_N^i + t\mathbf{d}_N^i), \mathbf{x}_N^i + t\mathbf{d}_N^i)$$

$$\text{subject to } \mathbf{x}_N^L \leq \mathbf{x}_N^i + t\mathbf{d}_N^i \leq \mathbf{x}_N^U$$

を解いて、 $\mathbf{x}_N^{i+1} = \mathbf{x}_N^i + t^i\mathbf{d}_N^i$ とする。

この計算に必要な基底変数 \mathbf{x}_B の値は、非線形方程式、 $\mathbf{h}(\mathbf{x}_B, \mathbf{x}_N^{i+1}) = 0$ に対して擬 Newton 法の更新公式を適用して求めればよい。

$$\mathbf{x}_B^{k+1} = \mathbf{x}_B^k - \mathbf{B}^{-1}(\mathbf{x}_B^k, \mathbf{x}_N^{i+1})\mathbf{h}(\mathbf{x}_B^k, \mathbf{x}_N^{i+1}), \quad k = 1, 2, \dots \quad (AA.20)$$

ここで擬 Newton 法において基底変数 \mathbf{x}_B のある成分 $(\mathbf{x}_B)_r$ が、上下限制約を破るときには、現在の点と前の点との間で線形補間をして、 $(\mathbf{x}_B)_r$ が上下限制約の境界上になるようにする必要がある。そして、 $(\mathbf{x}_B)_r$ の代わりに上下限制約の間にある非基底変数 $(\mathbf{x}_N)_s$ を基底変数とする基底変換を行う。以上のアルゴリズムを整理すれば以下のようなになる。

ステップ 1: 制約式と非退化の仮定を同時に満たす初期点 $\mathbf{x}^1 = (\mathbf{x}_B^1, \mathbf{x}_N^1)$ を選び、 $i=1$ とする。

ステップ 2: 現在の点 $\mathbf{x}^i = (\mathbf{x}_B^i, \mathbf{x}_N^i)$ において停止基準を満たせば終了。そうでなければ縮小勾配 $\nabla F(\mathbf{x}_N^i)$ を計算して、降下方向 \mathbf{d}_N^i を定める。

ステップ 3: 1 次元探索問題 (p.AA.8) を解き、ステップ幅 t^i と \mathbf{x}_N^{i+1} に対応する \mathbf{x}_B^{i+1} を求める。ここで基底変数が上下限制約を破る場合には t^i を縮小して境界上の点を求め、基底を入れ替えた後、基底変換を行う。 $i:=i+1$ としてステップ 2 へもどる。

4.5 メタヒューリスティック最適化手法

[1] GA のその他の突然変異の説明

1. 転座: 染色体の一部が同じ染色体の他の部分または他の染色体上に位置を変える。
2. 重複: 染色体上である長さのコードを重複させる。
3. 逆位(あるいは反転): 染色体上で部分的に遺伝子の配列順序を入れ換える。
4. 挿入: 染色体上である長さのコードを挿入する。この結果、染色体の長さが変わる
5. 欠失: 染色体上である長さのコードを除去する。この場合も、染色体の長さは変わる。

[2] 適応 DE のアルゴリズム

適応 DE は、目的関数空間に傾斜場を定義し、交叉率の適応的な調整のために個体に年令を与えた DE の変種の一つである。以下にそのアルゴリズムの要約を示す。

ステップ 1 (発生): $2N_p$ 個の個体を発成させる。これらの世代を 1, 年令 $\text{Age}(i)$ を 0 才に初期化する。ここで年令は個体 i が生き続けた世代数と定義される。

ステップ 2 (傾斜場): 集団内に任意の一对の組を生成し, 互いの目的関数値を比較する。そして値が優れている(最小化問題では小さい)方を勝者, 他方を敗者として登録する。また勝者は一つ年をとり, 敗者は一つ若返える。

ステップ 3 (突然変異): 勝組から任意の一つの基準となる個体 $\mathbf{x}_{\text{base},t}$ を選択し, 突然変異個体を次式で計算する。

$$\mathbf{v}_{i,t+1} = \mathbf{x}_{\text{base}(i),t} + F(\mathbf{x}_{\text{win}(i),t} - \mathbf{x}_{\text{loss}(i),t}) \quad (i = 1, \dots, 2N_p) \quad (\text{AA.21})$$

ここで \mathbf{x}_{win} と \mathbf{x}_{loss} は, それぞれ選ばれた対の勝者と敗者を表す。この操作によって探索空間のいたるところで全般的には目的関数を減少させる方向(傾斜場)に突然変異個体を発生させることができる。

ステップ 4 (交叉): 通常の DE と同じ方法で交叉を行なう。しかし交叉率 P_c は年令に関する単調減少関数として与える。例えば

$$P_c = (a + c)e^{-b \cdot \text{Age}(i)} + c, \quad P_c = \max[a + c - b \cdot \text{Age}(i), c], \text{ など}$$

とする。ここで a, b および c は適当な正の定数である。ただし $0 < a + c \leq 1$ とする。この交叉率の与え方によって長生きした(年令の高い)個体の形質を次世代により継承され易くすることができる。

ステップ 5 (選択): 試行個体が標的個体より優れていれば更新し, 適当な年令を与える。例えば 0 才とすればよい。反対の場合は標的個体を残し年令を一つ加える。

ステップ 6: 収束条件を調べ, 満足していれば終了する。そうでなければ世代数を更新してからステップ 2 へ戻る。

5. 多目的計画法による実行支援

[1] その他のメタ多目的解析法

これまで多数の方法が提案されているものの中で例えばパレート基準に分類される VOES (vector optimized evolution strategy)^{AA12)} と predator-prey evolution 法^{AA13)} は, エリート主義を用いないものである。一方, DPGA (distance-based Pareto genetic algorithm)^{AA14)}, SPEA (strength Pareto evolutionary algorithm)^{AA15)}, MOMGA (multi-objective messy genetic algorithm)^{AA16)}, PAES (Pareto archived evolution strategy)^{AA17)} および GENMOP (multi-objective program)^{AA18)} は, エリート主義による方法である。一般的に多目的進化アルゴリズム(MOEA)の方法を比較してみて, エリート主義は探索の効率化にとって重要な役割を果たすことがわかっている^{AA19)}。

[2] 多目的最適化手法の対話型シンプレックス法

[例 AA.5] トルエン脱アルキル反応プロセスの信頼性を考慮した最適設計への適用例^{AA20)}

トルエンとスチームを原料とするベンゼン製造プロセスにおいて, 熱交換により予熱された原料は 2 段の反応器で脱アルキル化される。このとき生成ガス中に副製する水素ガスや低分子ガスは, そ

それぞれ高圧、低圧の分離器で除去される。さらに、未反応トルエンは蒸溜塔で分離回収され、原料の一部に加えられる。

このプロセスの基本設計上の評価項目として、経済性とそれに競合する安全性を考える。さらに、経済性評価は投資コスト(f_1)と年あたりの運転コスト(f_2)に分け、それぞれを個別的に取り扱い、両者間の適切な配分を問題にする。ここで、前者は、各プロセスユニットの投資コストの和として、後者は、使用ユーティリティコストの和として与える。

一方、安全性はプロセスの信頼性として次式で評価される。

$$f_3 = \{1 - (1 - R_R)Pr\}R_D \quad (\text{AA.22})$$

ここで、 R_R 、 R_D は、それぞれ反応器、蒸溜塔サブシステムの信頼性を、また Pr は低圧分離器の蒸溜滞時間内に反応サブシステムに起こったトラブルを修復できない確率を表す。

プロセスモデルは、各プロセスユニット毎の熱、及び物質に関する収支式や単位操作に関する化工計算式から構成され、非線形の代数方程式 $\mathbf{g}(\mathbf{x}, \mathbf{D}, \mathbf{p}) \geq \mathbf{0}$ として表わされる。ここで、 \mathbf{x} はシステムの状態変数を、 \mathbf{D} は、トルエン/スチーム比、蒸溜塔還流比、及び低圧分離器の容量を表す設計変数である。ここでの問題は、最終的に以下のように定式化される。

$$(\text{p.AA.9}) \quad \{ \text{Min}(f_1, f_2), \text{Max } f_3 \} \quad \text{subject to} \quad \begin{cases} \mathbf{g}(\mathbf{x}, D_1, D_2, D_3) \geq \mathbf{0} \\ \underline{D}_i \leq D_i \leq \bar{D}_i, (i=1,2,3) \\ \mathbf{x} \geq \mathbf{0} \end{cases}$$

ここで $\bar{D}_i, \underline{D}_i$ は、それぞれ各設計変数の許容値域の上下限值を表す。

数値計算における探索過程の一例を図 AA.8 に示す。探索初期に調整幅の大きかった状況が、探索の進展に従って収束していく様子と共に、経済性と信頼性間においては顕著で、投資コストと操業コスト間では弱いトレードオフの存在が観察される。また、ここでの選好最適化の結果は、信頼性向上のための投資を積極的に行っているといえる。

[3] 価値評価法

[例 AA.6] 車種の選択問題への適用例

ここでは、「最良の車種を選ぶということ」が検討課題となる。なお対象候補となる車種(代替案)は M 種とする。このとき、漠然と好ましさを比較するより、以下のような系統的な手順を採ったほうが総合的によい判断が可能となる。

ステップ 1: 最終目標(最良の車種を選ぶ)を最上位レベルの目標として設定する。

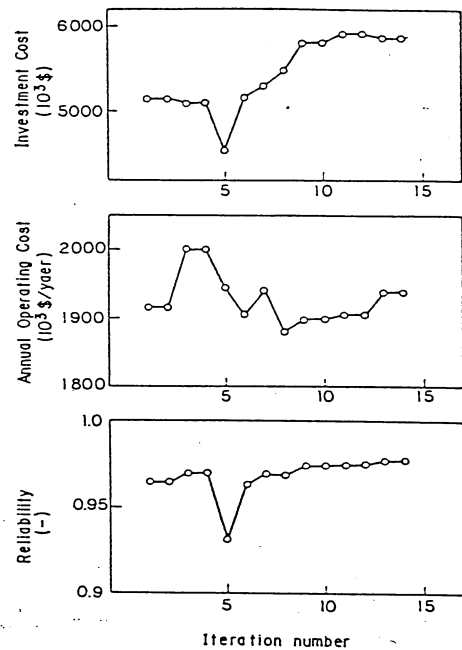


図 AA.8 対話型シンプレックス法による探索課程

ステップ 2: 上位レベルの目標を、より具体的に表現する一つ下位レベルの目標に分割する。こうした分割を順次行い、DM にとって最上位目標が現実的に評価可能な目標によってのみ表現されるようにする。結果は、ツリー構造を持つ目的木として表される。この例では、まず「費用」と、「見た目」と「安全性」に着目し、さらに費用は、「購入時の出費」と「維持に必要な費用」で評価し、維持費用は「定期点検時の費用」と「修理時の費用」を考えるといった手順で図 AA.9 に示すような評価木が作成される。ここでの最下位目標の総数を N とする。

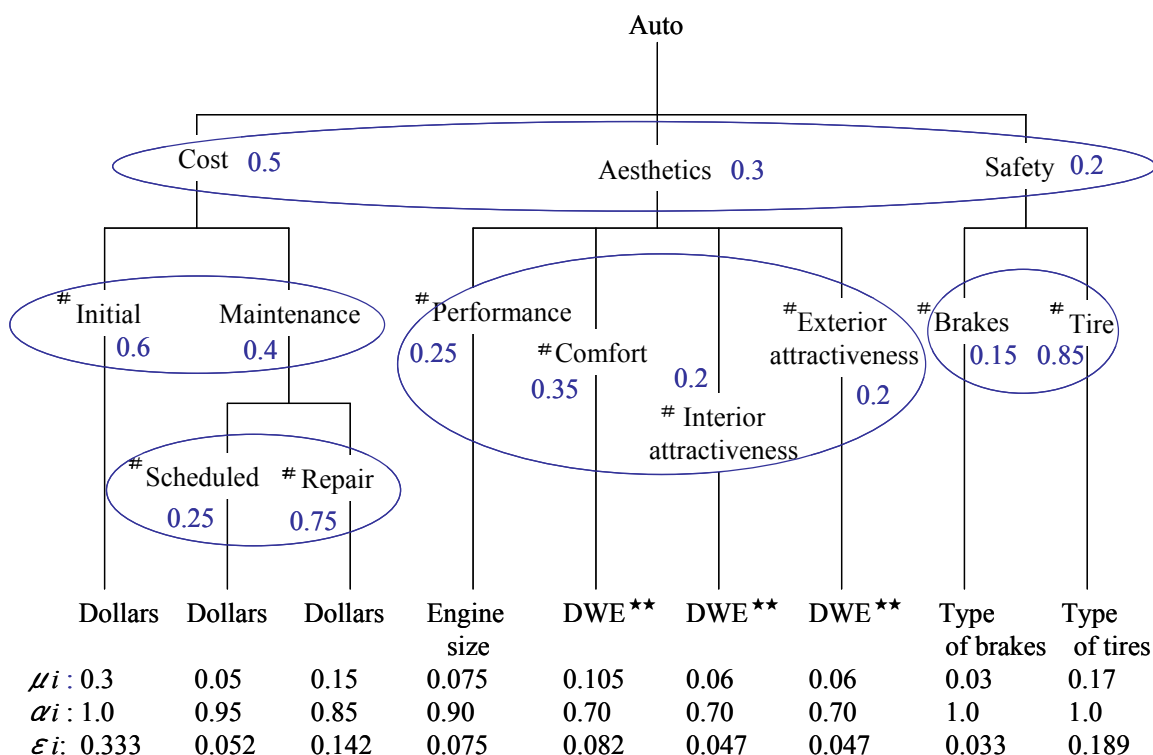


図 AA.9 価値評価法の自動車選択問題への適用例

ステップ 3: 分枝毎の最下位目標の各々に評価の基準を決め、これに基づき各代替案の満足度を得点 $s_i(j)$ として、例えば $[0, 1]$ のように一定範囲内で与える。ここで $s_i(j)$ は代替案 j の目標 i に関する得点を表す。 ($j=1, \dots, M, i=1, \dots, N$)

ステップ 4: 同一レベルに属し、かつ分枝元(親)を同じくする各目標(兄弟)に対して、分枝元の目標の達成にとっての相対的な重用度を表す重み係数を和が1となるよう配分する。これを分枝毎に全レベルに渡って行う。例えば、費用、見た目、安全性は同一レベル(兄弟関係)にあり、最終目標を共通の分枝元(親)としており、これら間の相対的重要度が重み係数としてそれぞれ 0.5, 0.3, 0.2 と与えられている。以下同様にする。

ステップ 5: 最下位目標毎に分枝を上位へ遡っていき、各レベルで配分された重み係数を順次掛け合わせる事によって、実効重み係数(Effective weight) μ_i を計算する。例えば、定期的な保守の実効重み係数は $0.25 \times 0.4 \times 0.5 = 0.05$ と計算される。

ステップ 6: 最下位目標毎に, 最終目標の評価に関する重要性に対して上述までの判断の確信

度(Adjusting factor) α_i ($0 < \alpha_i \leq 1$)を反映させる事によって実効重み係数を $\varepsilon_i = \mu_i \alpha_i / \sum_{i=1}^N \mu_i \alpha_i$ の

ように修正する。

ステップ 7: 修正実効重み係数(Adjusted weight)を用いて, 各代替案, ($j=1, \dots, M$)の総合評価点

を計算し, 最高得点のものを選択する。 *i.e.* $Max_{j=1 \dots M} \sum_{i=1}^N \varepsilon_i s_i(j)$

価値評価方法の構成原理は判りやすく, 有限個の代替案に対する多目的評価を簡単に行うことができる。しかしレベル毎の重み係数を意思決定者に直接的に判断することを求めることは, 必ずしも人間の判断になじまない。

[4] AHP の適用例

[例 AA.7] ここでは 1986 年の事故の後のスペースシャトルの開発計画に関する意思決定問題を取り上げる^{AA21)}。まず, 事故後にアメリカ政府が取るべき策として,

- {①しばらく保留して調査する。(保留), ②全体計画を中止する。(中止), ③このまま継続する。(継続), ④このプロジェクトを他の国に任せる。(外国)}

の四つの代替案が提案されたとする。(以後, ()内に略記したもので表す。)

またそれらを選択する基準として以下の五つがあがったとする。

- {①国のメンツを守る。(メンツ), ②安全の優先。(安全), ③軍事優先。(軍事), ④経済の発展を計る。(経済), ⑤科学技術の発展。(技術)}

結局, ここでの階層図は, 図 AA.10 のようになるので, まずレベル 1 の目標のためにはレベル 2 の五つの評価基準 (メンツ, 安全, 軍事, 経済, 技術)の重要度を示す重みを AHP の手順に従って求めたところ,

$$w = (0.343, 0.331, 0.207, 0.080, 0.038)^T \tag{AA.23}$$

であったとする。次に, レベル 2 の各基準からみた四つの代替案, (保留, 継続, 中止, 外国)のどれが望ましいかを表す重みを求める。そして, 国のメンツを保つという基準からみた四つの代替案の望ましさを表す重みは, (0.550, 0.232, 0.135, 0.083) のようであり, そのほか四つの基準からみた重みも表 AA.2 のように求められたとする。

代替案の総合重みは, 行列の各行の値を式(AA.23)の評価基準の重みで重み付けした総和で与えられる。この計算より, (0.394, 0.268, 0.258, 0.0800)となり, 保留が 0.394 と一番大きく, 結局「プロジェクトの進行を一時保留すべし」が選択されることになる。

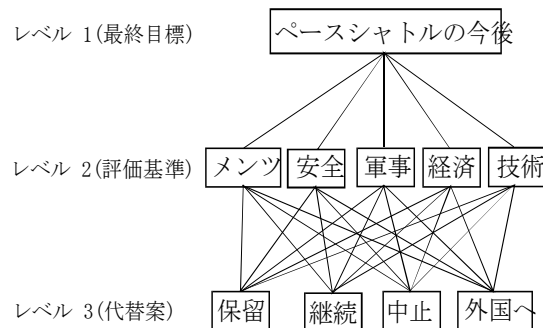


図 AA.10 例の評価の階層図

表 AA.2 最終目標からみた代替案の重み

	メンツ	安全	軍事	経済	技術
保留	0.550	0.260	0.313	0.409	0.553
継続	0.232	0.125	0.519	0.391	0.213
中止	0.135	0.547	0.109	0.063	0.074
外国	0.083	0.068	0.059	0.137	0.160

[5] 不完全問題に対するCookとKressの方法

まず2組の対比較行列 $P=\{p_{ij}\}$ と $Q=\{q_{ij}\}$ 間の距離を

$$d(P, Q) = 1/2 \sum_{i=1}^n \sum_{j=1}^n |Ln[p_{ij}/q_{ij}]| \quad (AA.24)$$

で定義する。そして不整合な行列, $A=\{a_{ij}\}$ を整合な行列に近付ける一つの方法として, 完全整合行列, $A^*=\{w_i/w_j\}$ との距離を最小にする次の問題を与える。

$$(p.AA.10) \quad \text{Min} \sum_{i=1}^n \sum_{j=1}^n |Ln[a_{ij}/(w_i/w_j)]| \quad \text{subject to} \quad \prod_{i=1}^n w_i = 1, w_i > 0$$

ここで, $\prod_{i=1}^n w_i = 1$ は規準化のための条件である。この問題は, $Ln(w_i) = x_i^+ - x_i^-$ と変数変換した上

で, 偏差変数, P_{ij} と N_{ij} の導入によって次のような目標計画問題に帰着される。

$$(p.AA.11) \quad \text{Min} \sum_{i=1}^n \sum_{j=1}^n (P_{ij} + N_{ij})$$

$$\text{subject to} \quad \begin{cases} x_i^+ - x_i^- - x_j^+ + x_j^- + N_{ij} + P_{ij} = Ln(a_{ij}), \\ \sum_{i=1}^n (x_i^+ - x_i^-) = 0 \\ x_i^+, x_i^-, P_{ij}, N_{ij} \geq 0, \quad i, j = 1, \dots, n \end{cases}$$

(p.AA.11)は線形計画法によって求解可能であり, 最終的に重み係数は以下のように計算される。

$$w_i' = w_i / \sum_{j=1}^n w_j, (w_i = \exp[x_i^+ - x_i^-])$$

5.5 まとめ中の補足

[1] プロジェクトスケジューリング(PERT)の概要

「製造工場の建設」, 「新製品の企画から設計, 生産, 消費, 廃棄に至るライフサイクルを考慮した開発計画」, 「コンピュータなどの新システムの導入計画」といったような一連の作業からなる非反復的な複合開発計画をプロジェクト(Project)と呼んでいる。ある意思決定の結果として決められたプロジェクトを完成させるには, 通常長期間にわたる多くの関係者の協力と, 設備, 資材の効果的な投入が必要となる。このためプロジェクト完成に必要な個々の作業の日程計画を正しく立てて,

その遂行を管理していかなければならない。従来このようなプロジェクトの日程管理には図 AA.11 に例示するような、ガントチャート(Gantt chart)が広く利用されてきた。

しかし、生産規模が大きくなり、生産活動のテンポが早くなるにつれて、ガントチャートによる日程管理では計画の内容を正確に表現することがむずかしく、プロジェクト実施中の計画のフォローアップも非常に困難となる。またプロジェクト一般は、業務遂行に際して不確定要素を多く含む上に、工期または納期や経済上の制約が存在する。このように未知の要素を多く含むプロジェクトを、定められた期日や経費で完成させるために、どのような手順に従えばよいかを明確にする必要がある。

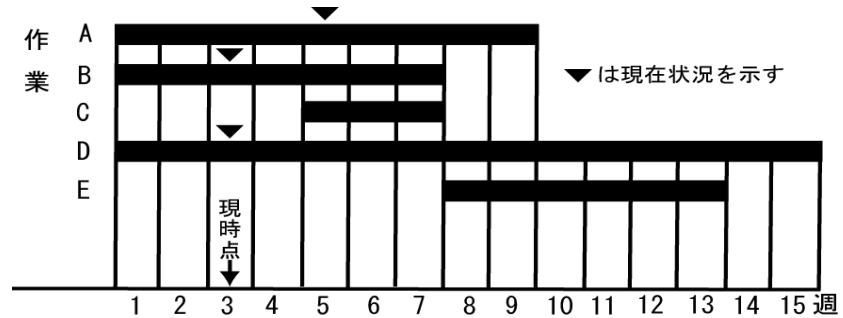


図 AA.11 ガントチャート

ある。PERT(Program

Evaluation and Review Technique)は、このような背景の中から1958年にアメリカ海軍とハミルトン社により開発されたネットワークを用いる計画管理手法である。工期に着目したPERT/Timeと経費に着目したPERT/Costがあるが、以下では前者について述べる。この手法により、問題の発生以前に日程計画のボトルネックとなる作業を見つけ、全体の仕事を調整することによって生産のおくれ、中断などの障害を最も少なくすることができるようになる。

A. PERT ネットワーク

ガントチャートは、これから実行しようとする作業を時間の経過に沿って描いた図で、横軸の時間目盛によって、ある作業あるいはプロジェクト全体が完了する時刻を知ることができる。ガントチャートでは、一つの作業中での作業の開始点あるいは終了点などの主要管理点の予定時点を明確にできるが、異なった作業間の主要管理点の相互関係を表すことはできない。従って、ある作業に何らかの変化が生じたとき、その変化がプロジェクト全体の計画にどのように影響してくるかを、早くしかも正確に把握することはむずかしい。

この問題点を克服するために、PERTではアローダイアグラムと呼ばれるネットワークを用いる。すなわち、目的のプロジェクトを構成する作業を意味するアクティビティと主要管理点であるイベントを用いてアローダイアグラムを作成する。アクティビティの開始点を開始イベント、終了点を終了イベントといい、各々唯一でなければならない。

通常、イベントはその中にイベント番号を書いた○印で、アクティビティは実線の矢印で示される。矢線の長さは意味をも

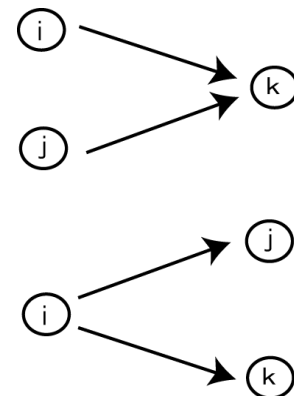


図 AA.12 イベントとアクティビティ

たず、方向は順序関係のみを表す。このとき、**図 AA.12** のように開始イベントもしくは終了イベントのいずれかを共有しているアクティビティが複数個存在していてもよいが、複数のアクティビティが開始、終了イベントをともに共有することは許されない。いいかえると、**図 AA.13** のように 1 対の開始イベントと終了イベントを二つ以上の矢印で結んではならない。これは、各アクティビティを一つのイベント対によって一意的に表現するためである。

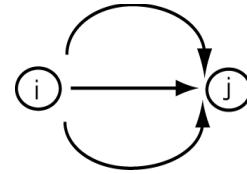
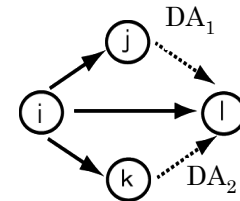


図 AA.13 禁止表示例

このような制約の下で、各アクティビティの論理関係を与えるため、ダミー・アクティビティ(dummy activity)を導入する必要がある。なお、ダミー・アクティビティは通常、点線の矢印で示される。例えば、 $DA_1(j, l)$ 、 $DA_2(k, l)$ のダミー・アクティビティを用いて**図 AA.13** を**図 AA.14** のように表現すれば、ダミー・アクティビティを含めて全てのアクティビティが同一のイベント対を持たないようにできることがわかる。



図AA.14 ダミーによるイベント対の一意性を表示

PERT 手法によりプロジェクト管理を実行する場合、ネットワーク作成がその根幹をなし、その作成要領は、大方次のとおりである。

- (1) 各アクティビティを一つのイベント対(i, j)、($i < j$)によって表す(逐次番号づけ)。A と B が同時作業となる場合、必ずダミー C を入れて**図 AA.14** のようにしなければならない。
- (2) 各イベントに入ってくるアクティビティや、イベントより出ていくアクティビティは、それぞれ毎に後続作業や先行作業をもつ。**図 AA.15 (1)**では C、D がともに A、B を先行アクティビティとしており、A、B のどちらの作業も完成した後でないと C も D も開始することができない。一方、同**図(2)**では、C は先行アクティビティとして A のみをもち、D は先行アクティビティとして A、B をもつことになる。

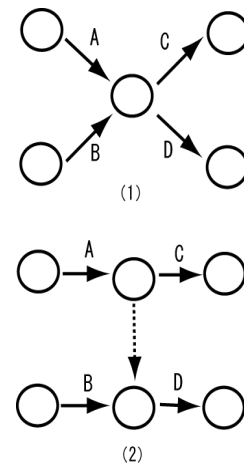


図 AA.15 順序関係の表

従って、A が完了すれば C は着手できるが、D は A、B がともに完了した後でないと着手できない。

- (3) 作業分割を正しく行う。最初、別々の作業と思われていたものが、実際にはまとめたほうがより現実的である場合や、同一作業と思われていた作業が実際にはいくつかの作業にわけて、別々のアクティビティとしたほうがよい場合もある。従って各作業について、分割または合成の検討を行って、作業単位をはっきりとしたものにする必要がある。

B. アクティビティの時間見積

PERT ネットワークを完成し、アクティビティの順序が決定された後は、この順序関係に基づいて日程計画を立てることになる。プロジェクトを構成する各アクティビティは、それを遂行のための作業時間を必要とする。各アクティビティの所要時間が分かれば、それによって PERT ネットワーク上の各点での時間的な前後関係が明らかになる。従って、それぞれのアクティビティをいつ開始する

べきかという日程計画が立てられる。また、このプロジェクトは何日までに完成したいとか、何日までに納入しなければならないかという期限が普通はある。この完成期限が守れるかどうか、それを守るためには日程管理上どういう処置が必要かという問題が検討できるようになる。

アクティビティの所要時間には、正確に分かっている場合と、推定である場合とが考えられる。このため、PERTには所要時間を一定の値とする場合と、一定の分布を仮定して確率的に考えていく立場とがある。多くの場合、個々のアクティビティの所要時間は推定による値であるため、一定のパラッキをもつ値と考えるほうがより一般的である。そこで以下では、所要時間の変動にベータ分布を仮定して、その平均と分散だけに注目する3点見積法について述べる。

(1) 楽観時間値(optimistic time) a

万事が通常状態よりもうまく運んだとき、そのアクティビティを完了させるのに要する時間の推定値(アクティビティを完成させるのに必要な最小期間)

(2) 最確時間値(most likely time) m

そのアクティビティを完成させるのに必要な時間の最良推定値

(3) 悲観時間値(pessimistic time) b

(1)とは反対に万事うまくいかなかったとき、そのアクティビティを完了させるのに要する時間の推定値(アクティビティを完成させるのに必要な最大期間)

これらの三つの値をもとに所要時間の期待値 t_e とその分散は、それぞれ次式のように与えられる。

$$t_e = (a + 4m + b)/6 \quad (\text{AA.25})$$

$$\sigma^2 = [(b - a)/6]^2 \quad (\text{AA.26})$$

また明らかに定義により、 $a \leq m \leq b$ の関係が成り立つ。なお1点見積法ではこのうち最確時間値 m のみを用いればよい。

C. PERT 計算

プロジェクトの日程を正確に算出するには、ネットワークの各イベントの期待生起期日と、それにもとづく種々の特性値を求める必要がある。以下ではこの種の計算をまとめて、PERT 計算と呼ぶことにする。

(1) イベント期日(event date)

ネットワークの各イベントについて、つぎの二つの期日を決める。その一つは、イベントの最も早く生起することが期待される期日で、最早期待期日(earliest expected date)と呼ばれる。この T_E は、

$$(T_E)_j = \max_i \{ (T_E)_i + (t_e)_{ij} \} \quad (\text{AA.27})$$

から計算される。ここで T_E の初期値 $(T_E)_{\text{init}}$ は 0 とする。なお、逐次番号づけが施されていれば、 T_E の計算は出発イベントから番号の増加する順序で行えばよい。

二つ目は、そのイベントの生起が許される範囲で最も遅れる場合の期日で、最遅許容期日(latest allowable date)と呼ばれる。この T_L は、

$$(T_L)_i = \min_j \{(T_L)_j - (t_e)_{ij}\} \quad (\text{AA.28})$$

により得られる。最終イベントに予定完成日 T_P が与えられているときは、 $(T_L)_{\text{term}}=(T_P)_{\text{term}}$ とし、そうでなければ、 $(T_L)_{\text{term}}=(T_E)_{\text{term}}$ とする。 T_L の計算は T_E の場合と逆に、最終イベントから番号の減少する順序で行われる。

またイベント j における $(T_E)_j$ の分散 V_j は、出発イベントの分散 V_{init} を 0 として、

$$V_j = V_i + \sigma_{ij}^2 \quad (\text{AA.29})$$

より求められる。ここに、 i は式(AA.27)で Max を計算する際にとられた値を用いる。

以上から、イベント j での予定完成期日を $(T_P)_j$ とすれば、この予定完成期日に間にあう確率 P_j は、

$$P_j = \int_{-\infty}^{y_j} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy \quad (\text{AA.30})$$

となる。ここで

$$y_j = \{(T_P)_j - (T_E)_j\} / \sqrt{V_j} \quad (\text{AA.31})$$

である。この P_j の値はプロジェクト遂行上の決定に関わる一つの重要な尺度となる。 $P_j=0.5$ すなわち 50% の確率で予定期日に間にあうときには $y_j=0$ 、すなわち $(T_P)_j=(T_E)_j$ となる。従ってこの値より小さすぎたり、大きすぎたりする場合には、スケジュールを再検討する必要がある。その際、 P_j の値をどのように解釈してそれに対していかなる処置をとるかは、管理者の主観的判断に依存するが、プロジェクトの取り扱いに対する一応の目安として、つぎのような指針がある。

(a) $P_j < 0.3$ の場合は、その計画を改訂しないかぎり、実現可能性についてかなりのリスクを覚悟しなければならない。

(b) $P_j > 0.6$ の場合は、あまりにも多くの資源が使われて無駄がありすぎることを意味する。

1) $(T_E)_i$ と $(T_L)_i$ の計算アルゴリズム

$(T_E)_i$ と $(T_L)_i$ は既述のとおり、ネットワークをたどって逐次的に求めることができる。しかし、ネットワークが大きくなれば、この方法は容易でなくなる。そこで、手計算では、以下に示す手順によれば便利である。なお、これは計算機プログラムで用いられるアルゴリズムの基礎にもなっている。

[例 AA.8] 図 AA.16 および表 AA.3 に示す条件の下でイベント期日を計算せよ。

ステップ 1: 表 AA.4 に示す行列の i 行 j 列の要素として、それぞれの個所に $(t_e)_{ij}$ を記入する。このとき、逐次番号づけが施されているので、 $(t_e)_{ij}$ の値は対角要素より上にあらわれる。ここでのネットワークにはダミーは存在しないが、もしダミーが存在すれば、その要素の値は 0 とする。

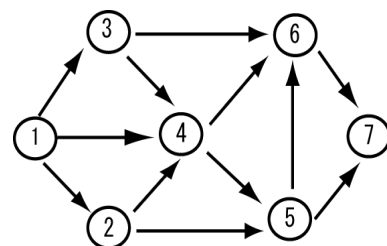


図 AA.16 例題のネットワーク

アクティビティ	a	m	b	t_e	σ_{ij}^2
1,2	7.0	10.0	11.0	9.67	0.44
1,3	5.0	8.0	10.0	7.83	0.69
1,4	11.0	16.0	18.0	15.50	1.36
2,4	3.0	6.0	8.0	5.83	0.69
2,5	12.0	15.0	17.0	14.83	0.69
3,4	6.0	9.0	11.0	8.83	0.69
3,6	13.0	16.0	18.0	15.83	0.69
4,5	4.0	8.0	12.0	8.00	1.78
4,6	5.0	6.0	7.0	6.00	0.11
5,6	4.0	4.0	4.0	4.00	0.00
5,7	6.0	9.0	11.0	8.83	0.69
6,7	3.0	4.0	6.0	4.17	0.25

T_L	0.00	9.83	7.83	16.66	24.66	29.32	33.49			
j	1	2	3	4	5	6	7	j	T_E	V
1		9.67	7.83	15.50				1	0.00	0.00
2				5.83	14.83			2	9.67	0.44
3				8.83		15.83		3	7.83	0.69
4					8.00	6.00		4	16.66	1.38
5						4.00	8.83	5	24.66	3.16
6							4.17	6	28.66	3.16
7								7	33.49	3.85

ステップ 2: 出発イベント($j=1$)で、 $(T_E)_1=0.0$ 、 $V_1=0.0$ とする。次に $j=2$ では、 $i=1$ に対応する $(t_e)_{12}$ の値は 9.67 であるから、式(AA.27)より

$$(T_E)_2 = (T_E)_1 + 9.67 = 9.67$$

となる。また式(AA.29)より分散は次式のように計算される。

$$V_2 = V_1 + \sigma_{12}^2 = 0.44$$

同様に $j=3$ では、 $i=1$ に対応する $(t_e)_{13}$ の値は 7.83 であるから、

$$(T_E)_3 = (T_E)_1 + 7.83 = 7.83$$

である。また分散は、次のようになる。

$$V_3 = V_1 + \sigma_{13}^2 = 0.69$$

$j=4$ では、 $i=1, 2$ および 3 に対応する行に値があるから、

$$\begin{aligned}(T_E)_4 &= \max \{(T_E)_1 + 15.5, (T_E)_2 + 5.83, (T_E)_3 + 8.83\} \\ &= \max_i \{15.50, 15.50, 16.66\} = 16.66 \quad (i=3)\end{aligned}$$

となり、分散は次のようになる。

$$V_4 = V_3 + \sigma_{34}^2 = 0.69 + 0.69 = 1.38$$

上述の計算を繰り返すことで、表の T_E , V 欄が完成される。

ステップ 3: 最終イベントについて、 $(T_P)_N$ が与えられていれば、 $(T_L)_N = (T_P)_N$ とし、与えられていなければ、 $(T_L)_N = (T_E)_N$ として、 $(T_L)_i$ の出発値とする。例では、 $(T_P)_N$ ($N=7$) は与えられていないので

$$(T_L)_7 = (T_E)_7 = 33.49$$

ととる。次に $i=6$ の行をみると、第 7 列に 4.17 があるから、次のようになる。

$$(T_L)_6 = (T_L)_7 - 4.17 = 29.32$$

以下、式(AA.28)に従って順次 $(T_L)_i$ を計算していき、 $(T_L)_1$ が求まれば手順はすべて完了し、表の行列の T_L , T_E , V の欄にそれぞれの値が定まる。

[例 AA.9] [例 AA.8] で、イベント 5 の予定期日が $(T_P)_5 = 23.0$ と与えられているとする時、これに間にあう確率を計算せよ。

表 AA.4 で $(T_E)_5 = 24.66$, $V_5 = 3.16$ となっているから、式(AA.31)より

$$y_5 = (23.0 - 24.66) / \sqrt{3.16} = -0.93$$

であり、正規分布表より予定に間にあう確率は、 $P_5 = 0.18$ (18%) となる。このように P が 30% 以下の値をとる場合、既述のようにスケジュールに無理があるとみて、再検討を行うのが普通である。

(2) アクティビティ期日

各アクティビティについて、以下のような期日を求める。

(a) 最早開始期日 (earliest start date): $(ES)_{ij} = (T_E)_i$

(b) 最遅開始期日 (latest start date): $(LS)_{ij} = (T_L)_j - (t_e)_{ij}$

(c) 最早完了期日 (earliest finish date): $(EF)_{ij} = (T_E)_i + (t_e)_{ij}$

(d) 最遅完了期日 (latest finish date): $(LF)_{ij} = (T_L)_j$

[例 AA.8]のネットワークの場合、アクティビティ期日は表 AA.5 のようになる。

表 AA.5 [例 AA.8]のアクティビティ期日に関するリスト

アクティビティ	ES	LS	EF	LF
1,2	0.00	0.16	9.67	9.83
1,3	0.00	0.00	7.83	7.83
1,4	0.00	1.16	15.50	16.66
2,4	9.67	10.83	15.50	16.66
2,5	9.67	9.83	24.50	24.66
3,4	7.83	7.83	16.66	16.66
3,6	7.83	13.49	23.66	29.32
4,5	16.66	16.66	24.66	24.66
4,6	16.66	23.32	22.66	29.32
5,6	24.66	25.32	28.66	29.32
5,7	24.66	24.66	33.49	33.49
6,7	28.66	29.32	32.83	33.49

(3) イベント・スラック(event slack)とアクティビティ・フロート(activity float)

最遅許容期日 T_L と最早期待期日 T_E との差 T_L, T_E をそのイベントのスラックとよび、イベントの余裕を表す尺度となる。すなわち、あるイベントのスラック S_i は、

$$S_i = (T_L)_i - (T_E)_i \quad (\text{AA.32})$$

で与えられ、

$S_i > 0$ のときスケジュールより進みすぎ(資源過剰)

$S_i = 0$ のときスケジュール通り(資源適正)

$S_i < 0$ のときスケジュールより遅れすぎ(資源不足)

を意味する。

ところで、アクティビティの余裕期間は一般にフロートとよばれる。フロートには数種のものがあるが、ここでは次に述べるクリティカル・パスの計算に必要となる TF (トータル・フロート; total float) に限定して述べる。これはアクティビティ(i, j)の先行アクティビティが最早開始期日に開始され、後続アクティビティが最遅許容期日に着手されるとした場合の余裕であり、

$$(TF)_{ij} = (T_L)_j - (T_E)_i - (t_e)_{ij} = (LS)_{ij} - (ES)_{ij} \quad (\text{AA.33})$$

から計算される。またこの値は、 $LF - EF$ に等しい。

トータル・フロート $(TF)_{ij}$ が最小の値($(T_L)_{\text{term}} = (T_E)_{\text{term}}$ のときは 0)のアクティビティを結ぶ経路はクリティカル・パス(critical path)とよばれる。[例 AA.7]のフロート・リストの表 AA.6 において、 CP の欄に *をつけたのがクリティカル・パス上のアクティビティである。結局、プロジェクトの期待所要時間は、

$$t_p = \sum_{(i,j)} (t_e)_{ij}, (i,j) \in \{CP\} \quad (AA.34)$$

となる。ここに $\{CP\}$ はクリティカル・パス上のアクティビティの集合を表す。従って、 t_p はクリティカル・パス上のアクティビティの所要時間によって決定され、その他のアクティビティには関係がないことがわかる。要するに、開始イベントと終了イベントを結ぶ最短のパスがクリティカル・パスとなる。そして、そのパス上のアクティビティの所要時間の総和によりプロジェクトの期待所要時間が決定されることから、クリティカル・パス上のアクティビティに管理の重点をおくことの必要性がわかる。

表 AA.6 [例 AA.8] のフロート・リスト

アクティビティ	1,2	1,3	1,4	2,4	2,5	3,4	3,6	4,5	4,6	5,6	5,7	6,7
TF	0.16	0.0	1.16	1.16	0.16	0.0	5.66	0.0	6.66	0.66	0.0	0.66
CP		*				*		*			*	

D. PERT のまとめ

PERT とはほぼ同時期に独立して CPM (Critical Path Method) が開発された。ともにネットワーク図を用いてプロジェクトスケジューリングを行うことから、併せて「PERT/CPM」と呼ばれる。またこれらを実用問題に適用するためにはコンピュータの利用が不可欠であり、多くのソフトウェアが開発されている。また解説書や成書^{AA22)}も多数ある。CPMとPERTとの違いは、CPMがコスト最適化の手法であるのに対して、PERT/Timeは不確定要素の多いプロジェクトにおいて確率的に時間見積もりを算出し、その完成時間の短縮を図る手法である点にある。アポロ計画の実施にあたってPERTが成功裏に適用されたことにみられるように、PERTは計画を実行に移す際の有力な実用的手法の一つとしてのより一層の活用が期待される。

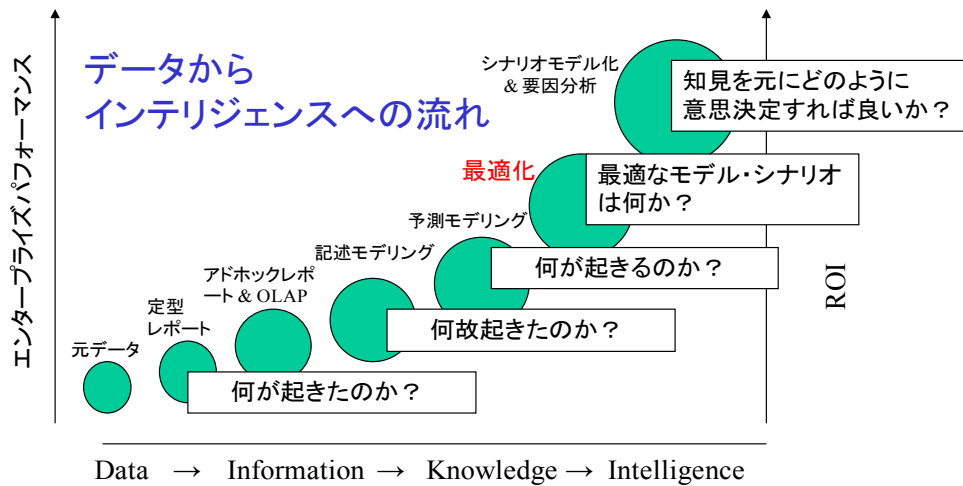
6.4.3 理念形成へのテーゼ

表 AA.7 に近年の最適化に関連するキーワードと最適化手法との相関を示した。行列中、塗りを入

表 AA.7 最適化問題の分類と手法の整理

Keywords	General	Combinatorial		Miscellaneous	Multi-objective		
	LP/NLP	Meta-heuristic	MIP		Societal System	Interactive	New Trend
SCM/ Logistics		[2]	[2]				Hybrid
Scheduling (Assignment)	centralized						MOGA
	decentralized			[1] プロトコル			エージェント?
Production Planning (JIT/TOC)	[4], [5]	[7] PSO		待ち行列	[4]		
Real-time Optimization	[7]						
Model Predictive Control							
Hybrid System			[3]	[8] マルチパラメトリック			
Decision Making					AHP DEA MAUT		エージェント
Miscellaneous	[6] Package ソフト	免疫アルゴリズム DE Scatter search	[1] 離散最適化 (グラフ)	形状最適化	IDEFO [4] UML	NIMBUS	MOON2R

れてある箇所は相互に関連が深いことを表す。また、情報の質の面から現場の粗野なデータを昇華された知識へ導く一連のプロセスを現実化することが現在求められているとする草刈の見方^{AA23)} (図 AA.17 参照)は、理念形成にとって参考にすべきものの一つである。



草刈君子 (SAS Institute Japan)

OLAP: On Line Analytical Processing

- A fundamental foundation for Intelligent Solutions including business performance management, planning, budgeting, forecasting, financial reporting, analysis, simulation models, knowledge discovery, and data warehouse reporting.
- Performs multidimensional analysis of enterprise data
- Provides the capabilities for complex calculations, trend analysis and very sophisticated data modeling.

図 AA.17 理念形成へのテーゼ

おわりに

最適化学は広いスコープの中で展開されていくべきである。このためにはたえず必要とすべき要素技術について関心を寄せ、さらに追加していく必要がある。

付録の参考文献

- AA1) J. Reisener, M. A. Reuter and J. Kru"er: "Modelling of the Mass Transfer in Gas-Sparged Electrolysers with Neural Nets", Chem. Eng. Sci., 48, 6, pp. 1089-1101 (1993).
- AA2) H. Su, T. J. and McAvoy: "Integration of Multilayer Perceptron Networks and Linear Dynamic Models: A Hammerstein Modeling Approach", Ind. Eng. Chem. Res., 32, 9, pp. 1927-1936 (1993).
- AA3) Y. You and M. Nikolaou: "Dynamic Process Modelling with Recurrent Neural Networks", AIChE J., 39, 10, pp. 1654-1667 (1993).
- AA4) N. V. Bhat and T. J. McAvoy: "Use of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems", Computers chem. Engng., 14, 4/5, pp. 573-583 (1990).
- AA5) 松葉育雄: "カオスと予測", 数理科学, 348, pp. 64-69 (1992).

- AA6) D. C. Psychogios and L. H. Ungar: "A Hybrid Neural Network-First Principles Approach to Process Modeling", *AIChE J.*, 38, 10, pp. 1499-1511 (1992).
- AA7) A. P. Sethi and G. L. Thompson: "Math. Program", 29, 219 (1984).
- AA8) Y. Shimizu: "Progressive application of linear programs concerning with problem-solving in chemical processes - Development and application of PROLP (in Japanese)", *Kagaku Kogaku Ronbunshu*, 21, 3, pp. 521-530 (1995).
- AA9) Y. Shimizu: "Customization of algorithm for progressive linear programs (PROLP) using neural net work (in Japanese)", *Kagaku Kogaku Ronbunshu*, 21, 3, pp. 614-617 (1995).
- AA10) Y. Shimizu: "Two-step algorithm of generalized PAPA method applied to linear programming solution of dynamic matrix control", *Annu. Rep. Res. Reactor Inst., Kyoto Univ.*, 24, pp. 84-89 (1991).
- AA11) J. Abadie and J. Carpentier (R. Fletcher, Ed.): "Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints In Optimization", Academic Press, New York, pp. 37-47 (1969).
- AA12) F. Kursawe: "A variant of evolution strategies for vector optimization", In *Proc. Parallel Problem Solving from Nature I (PPSN-I)*, pp. 193-197(1990).
- AA13) M. Laumanns, G. Rudolph and H. P. Schwefel: "A spatial predator-prey approach to multi-objective optimization: a preliminary study", *Proc. Parallel Problem Solving from Nature V (PPSN-V)*, pp. 241-249 (1998).
- AA14) S. Kundu and A. Osyczka: "The effect of genetic algorithm selection mechanisms on multicriteria optimization using the distance method", *Proc. Fifth International Conference on Intelligent Systems (Reno, NV). ISCA*, pp. 164-168 (1996).
- AA15) E. Zitzler and L. Thiele: "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach", *IEEE Transactions on Evolutionary Computation*, 3, pp. 257-271 (1999).
- AA16) K. Deb and D. E. Goldberg: "MGA in C: a messy genetic algorithm in C", Technical Report 91008, Illinois Genetic Algorithms Laboratory (IIIIGAL) (1991).
- AA17) J. Knowles and D. Corne: "M-PAES: a memetic algorithm for multiobjective optimization", *Proc. 2000 Congress on Evolutionary Computation, Piscataway*, 1, pp. 325-332 (2000).
- AA18) M. R. Knarr, M. N. Goltz, G. B. Lamont and J. Huang: "In situ bioremediation of perchlorate-contaminated groundwater using a multi-objective parallel evolutionary algorithm", *Proc. Congress on Evolutionary Computation (GEC, 2003), Piscataway*, 1, pp. 1604-1611 (2003).
- AA19) E. Zitzler, K. Deb and L. Thiele: "Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*", 8, pp. 173-195 (2000).
- AA20) T. Umeda, S. Kobayashi and A. Ichikawa: "Interactive solution to multiple criteria problems in chemical process design" *Computer and Chemical Engineering*, 4, pp. 157-165 (1980).
- AA21) 刀根薫: "ゲーム感覚意思決定法 AHP 入門", 日科技連出版社, 東京 (1990).
- AA22) 関根智明: "PERT・CPM", 日科技連出版社, 東京 (1975).

AA23) 草刈君子：“数理計画法の実用モデリングについて”，オペレーションズリサーチ, 50, pp. 238-242 (2005).

演習問題

[AA1] 図 AA.2 中の「MC 加工する」を詳細化するダイヤグラムを描け。

[AA2] 図 AE.1 のペトリネットにおいて，各事象を以下に従って分類せよ。

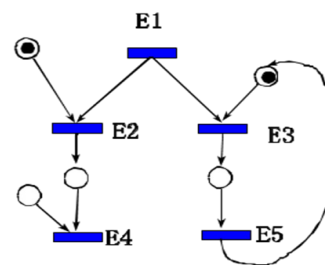


図 AE.1 問題 AA.2 の図

- (a) 一度も発火しない事象 []
- (b) 一度だけ発火する事象 []
- (c) 何度も発火する事象 [E1,]

[AA3] RBF ネットワークで， p 個の学習データに $(p+1)$ 番目を追加したときの更新式， p 番目を除去したときの更新式はそれぞれ次式で与えられることを確かめよ。

(1) 追加したとき

$$\mathbf{A}_{p+1}^{-1} = \mathbf{A}_p^{-1} - \frac{\mathbf{A}_p^{-1} \mathbf{h}_{p+1} \mathbf{h}_{p+1}^T \mathbf{A}_p^{-1}}{1 + \mathbf{h}_{p+1}^T \mathbf{A}_p^{-1} \mathbf{h}_{p+1}}$$

(1) 除去したとき

$$\mathbf{A}_{p-1}^{-1} = \mathbf{A}_p^{-1} + \frac{\mathbf{A}_p^{-1} \mathbf{h}_i \mathbf{h}_i^T \mathbf{A}_p^{-1}}{1 + \mathbf{h}_i^T \mathbf{A}_p^{-1} \mathbf{h}_i}$$

[AA4] いま二種類の食品AとBがあり，それぞれの単位 kg 量当りの炭水化物，蛋白質，脂肪の含有量[g]と単価[¥]が表 AE.1 のように与えられているとする。

表 AE.1 問題

	炭水化物[g/kg]	蛋白質[g/kg]	脂肪[g/kg]	単価[¥/kg]
食品A	400	100	150	1000
食品B	200	300	200	1500

このとき，炭水化物と蛋白質をそれぞれ 400[g]，300[g]以上含み，脂肪は 600[g]以下になるようにするためには，食品AとBを何グラムずつ食べるのが最も経済的になるか。

[AA5] $x_1 + x_2 + \dots + x_n = a$ (一定) $x_i \geq 0$ ($i=1, \dots, n$) のとき， $f(\mathbf{x}) = x_1, x_2, \dots, x_n$ を最大とする x_1, x_2, \dots, x_n とそのときの最大値を求めよ。

[AA6] 一定の長さ a の線分の両側を同じ長さ b を角度 θ まで折り曲げて台形をつくる時、この台形の面積を最大とする長さ b と角度 θ の値およびそのときの面積を求めよ。

[AA7] 以下の説明の正誤を示せ。

- (1) \mathbf{x}^* が X 中($\mathbf{x}^* \in X$)において f の大域的な最小解であれば、 $X' \subset X$ であるような X' においても大域的な最小解である。ただし $\mathbf{x}^* \in X$ とする。
- (2) $f(\mathbf{x})$ の局所的な最小解 \mathbf{x}^* が X の内点でないとする。このとき $X' \supset X$ であるような X' においても \mathbf{x}^* は局所的な最小解となる。
- (3) $f(\mathbf{x})$ の局所的な最小解 \mathbf{x}^* が X の内点であるとする。このとき $X' \supset X$ であるような X' においても \mathbf{x}^* は局所的な最小解となる。
- (4) 次元探索において黄金分割法は最も効率のよい探索法である。
- (5) ペナルティ関数法の内点法における繰り返しでは、ペナルティ係数は、順次減少させていく。

[AA8] A～Fの六つのアクティビティからなるプロジェクトがあり、各アクティビティ間に表 AE.2 に与える順序関係があるとする。このとき以下の問いに答えよ。

- (1) この PERT ネットワークを図示せよ。
- (2) クリティカル・パスを求め、完成までに必要となる時間を求めよ。
- (3) イベント5が期日に間に合う確率を求め、どのような措置によって妥当な(実現性の高い)スケジュールとすることができるか考察せよ。

表 AE.2 アクティビティ表/イベント表					
アクティビティ	直接後続アクティビティ	t_e	σ^2	イベント	T_p
A	B, C	2.2	0.2	1	0
B	D, E	1	0.1	2	2
C	E, F	4.3	0.4	3	3
D	F	3.2	0.2	4	6
E	-	2.5	0.25	5	6
F	-	2.8	0.3	6	9

演習問題解答

[AA1] 省略

[AA2] (a) [E4], (b) [E2], (c) [E1, E3, E5]

[AA3] 省略

[AA4] 食品 A と B をそれぞれ x_1, x_2 [kg] ずつ食べるとして、定式化された問題に図解法を適用すれば、最適解 $(x_1, x_2)=(0.6, 0.8)$ と最小値 $f=1800$ が得られる。

[AA5] $x_1 = x_2 = \dots = x_n = \frac{a}{n}, f(x) = \left(\frac{a}{n}\right)^n$

[AA6] $b=a/3, \theta=\pi/3, \text{面積}=a^2\sqrt{3}/12$

[AA7] (1) 正しい, (2) 誤り, (3)正しい, (4) 誤り, (5) 正しい

[AA8] (1) 図 AE.2 のとおり (2) ①→②→④→⑤→⑥がクリティカルパスとなり 9.3 日要する。

(2) $y_5=(6-6.5)/\sqrt{0.6}=-0.645$ より $P_5=0.26$ と計算される。即ち 26%となり、基準値の 30%以下であり、より多くの資源の投入や予定期日を延ばすなどの変更を検討する、など。

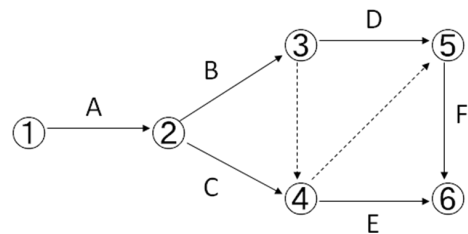


図 AE.2 問題 AE.2 の答

無断複製を禁ず