スケジューリングの山くずし法の
ニューラルネットによる定式化*福田 収一^{*1}, 志村 諭^{*2}Neural Network Formulation of Workload Leveling Technique
in Scheduling Problems

Shuichi FUKUDA and Satoshi SHIMURA

The workload leveling technique in the scheduling problem is a non-integer planning problem and one kind of solutions to the problem of allotment of manpower resources. But we have to determine the sequencing of activities at the same time. As workload leveling technique does not simply fill the concave part with the convex part, the constraints are tangled in a very complex manner. Thus, it is not easy to solve the problem by the conventional approach. Further, it often happens these days that we have to change the project schedule because of the situation of the ongoing activities or due to the unexpected disturbances. In this paper, the workload leveling technique is formulated in the framework of a neural network approach using the interconnected network and its effectiveness was examined by simulation.

Key Words: Scheduling Problem, Workload Leveling, Neural Network

1. はじめに

スケジューリング問題では、あるプロジェクトの遂行において、期間内に必要な人数や資源を割り付ける時、たいがい整数計画であり、また各要素が複雑に絡み合い非線形問題となる。プロジェクトをできるだけ早く安く仕上げるには、人員や機械が全期間で均等になる割付が望ましい。ある特定の工程の需要が急激なピークを持つと、プロジェクト自身の進行がその工程に左右され、またその期間では人数割付等でのバランスがくずれ、効率の悪いスケジュールとなる。例として、プロジェクトの工期と使用人員のみを図1で表すと、山がいくつもあることはその期間に過大な負荷がかかり、残業や一時的な人員増加・使用機械の需要増をもたらす。また、プロジェクト自身のコスト増加につながり、谷の部分に余剰資源を抱えむである。

従来山くずし法では、PERT/TIMEでクリティカル・パスを固定しプロジェクトの全期間を決め、高い山である工程が持つ余裕時間日数に着目して、作業開始日を前後にずらすことにより谷を埋め、全工程・全

期間を平準化する。この方法は有効であるが、実際の問題に適用するには平行して起こる作業工程は多種であり、1箇所山くずしを行うことにより工程が多いほどスケジュールを操作する箇所が飛躍的に増大する。また、クリティカル・パスの改善も望めない。

これを数式で解くには問題が多様で条件が異なるため、一般に処理には困難である。そこで、一度に各工程期間と使用人員、クリティカル・パス、すべてを統括的に処理するため、非線形・整数計画問題に有効な相互結合形のニューラルネットワークを用いて解決を計った。

2. 相互結合形ネットワーク

相互結合形ネットワークとは、原則としてあるユニットは他のユニットすべてに結合されている。このネットワークの特徴は、ある初期状態から出発して何度か状態変化を繰り返すうちに、そのネットワーク独自の安定な平衡状態へと移行して行く。ここでいう平衡状態とは、ネットワークが持つエネルギーが最小値をとる状態を指す。Hopfieldが提案した相互結合形のネットワークの特徴としては、ユニット間の結合が原則として対称結合である。すなわち、ユニット*i*から*j*へ強さ w_{ij} の結合がある場合には、その逆の結合、

* 原稿受付 平成6年3月22日。

*¹ 正員、東京都立科学技術大学 (〒191 日野市旭が丘6-6)。*² 准員、東京都立科学技術大学大学院。

図 1 山くずし前

j から i への結合の重みも w_{ji} はまた同じである。そして、状態変化は非同期的であり各ユニットは各自のタイミングで、結合の強さで重みづけられた入力信号を検知し、その総和をとって、その値により状態を変化させ、出力値を出す。

式に表すと、 i 番めのニューロンへの入力値 u_i を

$$u_i = \sum_{j=1}^n w_{ji} x_j + s_i - \theta_i \dots \dots \dots (1)$$

とすると、 i 番めのニューロンはこの u_i からの出力関数 $g(u_i)$ によってその状態を

$$x_i = g(u_i) \dots \dots \dots (2)$$

と決定する。そして、先の状態変化規則により変化させる。

$$\left. \begin{array}{l} \text{if } g(u_i) > 0 \text{ then } x_i = 1 \\ \text{if } g(u_i) \leq 0 \text{ then } x_i = 0 \end{array} \right\} \dots \dots \dots (3)$$

ここで、 $\sum w_{ji} u_j$ は他のユニットからの入力の総和、 s_i はネットワークの外部からの入力値、 θ_i はユニット i のしきい値を表す。全ユニットの状態の総和ネットワークのエネルギー状態は、式(1)～(3)の変化規則によりしだいに安定状態へと落ち着こうとする。ネットワークのエネルギー状態は、各ユニットの重みとしきい値により表現できる。相互結合形は、 $w_{ij} = w_{ji}$ であることを考慮して、次のエネルギー式が定義される。

$$E = -1/2 \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \theta_i x_i \dots \dots \dots (4)$$

3. 山くずし法の操作

山くずし法問題を総括的に扱い、柔軟性のある新たな方法を考える。まず、プロジェクトの初期計画に必要な条件をあげる。

1-(1) 全プロジェクトの工期

- (2) 必要資源(使用人数・使用機械)
- (3) 作業工程順序
- (4) 各工程の使用人数
- (5) 各工程の作業開始日・終了日

プロジェクトの初期計画 PERT/TIME を作成した時点で、条件 1-(1), (3), (4) はクリティカル・パスより具体的な数値と順序が定まり、条件(2)は全工期と 1 日当たりの制約資源をかけた量で決まる。条件(5)はネットワークの変動により値が変化し一度に定まらない。

山くずし法は、縦軸に人員資源・使用機械台数、横軸にプロジェクト期間を取り、二次元平面上に PERT/TIME を投影して視覚的に分かりやすくした解決法である。1 日当たりの最大使用資源×プロジェクト期間の面積に対し、条件(2)の必要資源(実働人員)の割合が大きくなれば労働資源の利用効率が高いことになる。よって、理想的な工程の配置は、実働人員の資源数に最も近い面積をもつ長方形形状である。

山くずし法の制約条件を整えると、

- 2-(1) 全必要資源数は決まっている。
- (2) 最短工期を超えない。
- (3) 余剰資源を最小にする。
- (4) 各工程の必要資源を割り当てる。
- (5) 各工程は重ならない。
- (6) 同一工程の形状は同層内で連続する。

条件 2-(1)～(3) は労働力の利用効率を最大限にする事であり、最大使用資源数にクリティカル・パスの期間(最短工期)を掛けた資源量が全必要資源数に近ければ、利用効率が高まり余剰資源が減少する。すなわち、あらかじめ余剰資源が最小で全工程が含まれる資源長方形が決められ、同時に最大使用資源数/単位日数が制限できる。図 1 で、使用資源数 (i)、日数 (j) と

図2 山くずし後

し、最小単位を1ブロック(日・人)とする。最大使用資源数は3人、最短工期は11日。余剰資源を含めた総ブロック数は33(日・人)。全工程に必要な資源数は19(日・人)。条件2-(4)より工程A~Gの必要ブロック数を D_k (日・人)。条件2-(1)を D_k で表現すると、

$$\sum_k D_k = 19 \quad (\text{日・人}) \quad (5)$$

全体を横軸と平行に使用資源ごとの階層に分け、各層のもつブロック数を H_i 、また日数ごとの使用資源数を S_k とする。

$$\sum_i H_i = 33 \quad (6)$$

余剰資源数を R とし、 H_i 、 D_k で表せば

$$R = \sum_i H_i - \sum_k D_k = 14 \quad (7)$$

$\sum D_k$ はプロジェクトを計画した段階で決まっているが、 H_i はクリティカル・パスを計算した時点で決まる。余剰資源の最小化は $\sum H_i$ と最大使用資源数とにより、 $\sum H_i$ が $\sum D_k$ と等しくなればよく、労働資源の利用効率が100%になる。したがって、工程の必要ブロック数を二次元 D_{ik} に拡張し、式(7)を H_i 層ごとに注目する。

余剰資源を表すと、

$$R = \sum_i \left(H_i - \sum_k D_{ik} \right) = 14 \quad (8)$$

次に、日数当たりの余剰資源を考える。よって工程の必要ブロック数に日数の条件を加え、三次元 D_{ijk} に拡張する。

$$R = \sum_j \left(S_j - \sum_i \sum_k D_{ijk} \right) = 14 \quad (9)$$

余剰資源数を決定する要素は、余剰資源は山と谷部分の資源ブロック数の差である。最大使用資源数と最短工期よりもとまる資源長方形と、全必要資源数の差であり、最大資源数が制約されるとき、最大資源ブロック数の整数倍と必要資源ブロック数が、最も近い工期ならば最適スケジュールといえる。

$$S_k \times H_n - \sum_{ijk} D_{ijk} = R \approx 0 \quad (10)$$

(n は整数)

最短工期 H_n を求めるには

$$(\sum_{ijk} D_{ijk}) / S_k = m_k \quad (11)$$

式(11)の商 m_k を切り上げた値である。初期条件を最短工期に変えても、式(10)と同様に

$$(\sum_{ijk} D_{ijk}) / H_i = m_i \quad (12)$$

であり、余剰資源を最小化する S_k が決定可能である。

式(7)を用い初期条件の一つ最大使用資源数あるいは最短工期から余剰資源 R を表現可能なことを確認できた。また、スケジュール表の総ブロック数も D_{ijk} と R で表現でき、あらためて H_i 、 S_k 、 D_{ijk} 、 R の関係式を表現する。

$$\sum_i (H_i - \sum_{jk} D_{ijk}) = R \quad (13)$$

$$\sum_j (S_j - \sum_{ik} D_{ijk}) = R \quad (14)$$

4. 状態変化式

3章で山くずし法の各条件を式に表現したので、ニューラルネットワークの目的関数化をはかる。山くずし法はスケジュールの平準化と余剰資源の最小化が目的である。よって、ネットワークの目的関数も式(13)、(14)から以下に定める。

$$\left. \begin{aligned} \phi A &= A/2 \{ R - \sum_i (H_i - \sum_{jk} D_{ijk}) \}^2 \\ \phi B &= B/2 \{ R - \sum_j (S_j - \sum_{ik} D_{ijk}) \}^2 \end{aligned} \right\} \quad (15)$$

ここで、最も重要なことはニューラルネットワークのユニットの、スケジュール表との対応である。目的関数が三次元であるから、ユニットも i, j, k で表され、 u_{ijk} とする。

いま、使用資源 $i \times$ 日数 j で表される平面 k を考える。平面 k は各工程に対応し、発火ユニット数の合計が工程 k での必要資源ブロック数となる。各工程で発火ブロック数は異なるが、全合計は式(5)である。こ

図 3 定式化の概念図

のとき、各平面でかつてにユニットが発火すると、総和が式(5)であっても、

$$\left. \begin{aligned} \sum_k (S_k - \sum_{ij} u_{ijk}) &\neq 0 \\ \sum_i (H_i - \sum_{jk} u_{ijk}) &\neq 0 \end{aligned} \right\} \dots\dots\dots (16)$$

が起こり得る。これを防ぐため、余剰資源もダミー工程として扱えば、目的関数 ϕA , ϕB も簡潔となる。

$$\left. \begin{aligned} \phi A &= A/2 \sum_i \left(H_i - \sum_{jk}^{k+1} u_{ijk} \right)^2 \\ \phi B &= B/2 \sum_j \left(S_j - \sum_{ik}^{k+1} u_{ijk} \right)^2 \end{aligned} \right\} \dots\dots\dots (17)$$

全体の目的関数が表現できたので、制約条件 2-(5)より各工程に必要な資源数を発火させる条件をつくる。

$$\phi C = C/2 \sum_k (D_k - \sum_{ij} u_{ijk})^2 \dots\dots\dots (18)$$

クリティカル・パスの工程は H_i の層に発火される必要があり、その制約は階層数 i と工程 k との関係を表す評価値 C_{ik} を用意する。 C_{ik} はコストと同様な数値であり、ユニット u_{ik} が値 C_{ik} の最も低い階層で発火すればエネルギー $C_{ik} \times u_{ik}$ が最小となる条件を付加する。

$$\phi D = D/2 \sum_{ik} (C_{ik} u_{ik})^2 \dots\dots\dots (19)$$

以上、式(17)～(19)を展開しネットワークのエネルギー関数と係数比較し状態変化式を求める。

$$\begin{aligned} \phi A &= 1/2 (A \sum_{ijk} \delta_i u_{ijk} u_{ijk} - 2A \sum_{ijk} H_i u_{ijk}) + \text{const.} \dots\dots\dots (20) \end{aligned}$$

$$\begin{aligned} \phi B &= 1/2 (B \sum_{ijk} \delta_j u_{ijk} u_{ijk} - 2B \sum_{ijk} S_j u_{ijk}) + \text{const.} \dots\dots\dots (21) \end{aligned}$$

$$\phi C = 1/2 (C \sum_{ijk} \delta_j u_{ijk} u_{ijk}$$

$$- 2C \sum_{ijk} D_k u_{ijk}) + \text{const.} \dots\dots\dots (22)$$

$$\phi D = 1/2 (D \sum_{ijk} C_{ik} u_{ik} C'_{ik} u'_{ik}) \dots\dots\dots (23)$$

ここにユニットは連続関数で扱ってきたため、ユニットの状態を 0 か 1 に収束する条件を加える。

$$\begin{aligned} \phi E &= -E/2 \sum_{ijk} (1 - 2u_{ijk})^2 \\ &= -1/2 (4E \sum_{ijk} \delta_{ijk} u_{ijk} u'_{jk} - 4E \sum_{ijk} u_{ijk}) + \text{const.} \dots\dots\dots (24) \end{aligned}$$

目的関数 ϕA , ϕB は基本的に同じ条件を表している。制約条件により使い分けられよい。ここでは ϕA を使う。式(20)～(24)と式(4)との係数比較により各ユニットに必要な重みとしきい値を求める。

$$\begin{aligned} w_{ijk} &= -A\delta_i - C\delta_k - DC_{ik}C'_{ik} \\ &\quad + 4E\delta_i\delta_j\delta_k \dots\dots\dots (25) \end{aligned}$$

$$h_{ijk} = AH_i + CD_k - 2E \dots\dots\dots (26)$$

式(1)～(3)の状態変化規則より状態変化式は

$$u_{ijk} = f(u) = \sum_{ijk} w_{ijk} u_{ijk} + h_{ijk} \dots\dots\dots (27)$$

各工程に必要な資源数の割当が行う式が特定できたが、状態変化式(5)～(26)だけでは平面 j 上のユニットが H_i 層ごとに幾つ発火するかを割り当てたに過ぎない。プロジェクトではスケジュール表に従って工程の優先順位がある。順位は階層 H_i のどこを取り上げても同じである。また、工程 j が H_i 層で 1 つ以上発火した場合連続しているべきであるため、一つ以上のときはグループ化し 1 個とみなす。そして、新たに $j \times j$ の行列を用意して各工程間の距離により、優先順位を決定すれば良い。プロジェクトの工程順も式(19)と同様に日数 j と工程 k との関係を表す評価値 L_{jk} を用意する。

$$\phi F = \sum_{jk} (L_{jk} u_{jk})^2 \dots\dots\dots (28)$$

このとき、新たにユニット v_{ji} を対応させると、式(20)～(24)より

$$H_i = 1, D_j = 1$$

と考え、式(26)、(27)を利用でき

$$w_{ik} = -A\delta_i - C\delta_j + 4E\delta_i\delta_k - FL_{ik}L'_{ik} \dots\dots\dots (29)$$

$$h_{ik} = A + C - 2E \dots\dots\dots (30)$$

以上、式(25)、(26)、(29)、(30)を得た。

これを適用した結果、図1の余剰資源数14(日・人)、全工期11日をそれぞれ図2の2(日・人)、7日に減少させることができた。

定式化の概念図を図3に示す。

5. ま と め

以上のように求められた山くずし法の定式化について例題により検討した結果、本定式化の有効性が確認できた。

例えば最大使用資源数3人、最短工期11日、余剰資

源を含めた総ブロック数33(日・人)、全工程に必要な資源数19(日・人)の例題では、本方法を適用し、山くずしを行うことにより、余剰資源数を当初の14(日・人)から2(日・人)に、全工期を当初の11日から7日に減少させることができた。

相互結合形ネットワークは非線形問題に適しているが、ここでは割当問題と優先順位に対するネットワークに分割したことにより条件を簡単化でき、山くずし問題に対する柔軟性を高めることができた。

文 献

- (1) 麻生, ニューラルネットワーク情報処理, (1988), 産業図書.
- (2) 松本・大津, ニューロコンピューティング, (1992), 培風館.
- (3) 甘利, 神経回路網の数理, (1978), 産業図書.
- (4) 太原, 認知情報処理, (1991), オーム社.
- (5) 大平, ORのはなし, (1989), 日科技連.
- (6) 刀根, PERT 講座 I, (1966), 東洋経済新法社.
- (7) ニューロコンピュータ研究会編, ニューロコンピューティングの基礎理論, (1990), 海文堂.
- (8) 平野, Cでつくるニューラルネットワーク, (1991), パーソナルメディア.