

情報システム開発プロジェクト管理システムにおける 知識型スケジューリング方式†

大場 みち子^{††} 都 島 功^{††}
薦 田 憲 久^{††} 山 中 止 志 郎^{†††}

ソフトウェアの開発を中心とする情報システム開発プロジェクトの管理は、定量的な分析が難しい非定形業務である。工数の見積り、日程計画表への展開、あるいは、工程遅れ解消のためのスケジュール調整では、考慮すべき要因や調整の手段が、各プロジェクト管理者によって異なり、各人の経験や勤に大きく依存している。また、進捗管理での進捗状況の診断では、問題点を早期に発見する方法は、各プロジェクト管理者の経験から得たノウハウに基づいている。以上のように情報システム開発プロジェクトの管理は、ノウハウに大きく依存する業務であり、従来、計算機化が困難であった。筆者らは、上記問題を解決するプロジェクト管理システムを開発してきた。本論文では、このシステムにおいて新しく開発したノウハウを組み込むスケジューリング方式を提案する。本方式では、標準的な工数・開発期間の基本データをプロジェクトの特性の違いに応じてノウハウにより修正することにより、開発工数および開発期間を見積もる。また、スケジュール作成・調整のための負荷山崩しや調整の標準アルゴリズムを幾つか持ち、管理者の持つ負荷山崩しや調整方針のノウハウに従ってそれらを組み合わせて実行することにより、スケジュール作成・調整を行う。このプロジェクト管理システムは、実際の情報システム開発プロジェクトの管理に適用し、実用的な解が得られることを確認している。

1. はじめに

ソフトウェアの開発を中心とする情報システム開発プロジェクトは、ますます増加する一方、その開発期間は短期化している。このような状況下で、限られた資源（人、計算機等）のもとで、要求された納期までにシステムを完成するためには、適切なプロジェクトの管理が重要である。情報システム開発プロジェクトの管理は、定量的な分析が難しい非定形業務のため、各人の経験や勤といったノウハウに極めて依存する業務であり、計算機化が困難であった。

情報システム開発におけるプロジェクト管理の主要な業務として、①適切な工数と開発期間を見積り、それらを日程計画表（作業スケジュール）へ展開する計画立案業務と、②進捗状況の的確な把握により問題点を抽出し、発見された問題点に対してスケジュール調整等の適切な対策をタイムリーに行い、プロジェクトの進捗を制御する進捗管理業務がある。

第1の工数の見積りに関しては、いくつかのモデルが提案されているが^{1),2)}、考慮されている要因はわず

かで拡張性に乏しい等のため、実際の計画業務に適用できるレベルに至っていない。また、工数の見積りの定量的な手法も確立していない。一方、日程計画表に展開する問題は、種々の制約下で、見積った工数と開発期間との相互関係を考慮した特殊な組合せ問題であり、開発期間の長さにより工数が異なることも相まって、単純には自動化できない。

第2の問題抽出とスケジュール調整の問題は、ソフトウェアの成果の把握が難しい上に、進捗状況や人員調達の容易さ等を考慮してスケジュール調整をする必要があり、ベテランの管理者にとっても困難である。最近、この種の問題を扱うシステムが多く発表されているが³⁾⁻⁵⁾、これらは、再スケジュールリング機能が弱く、管理者のノウハウに依存している。

筆者らは、以上の問題点を解決する知識工学を利用したプロジェクト管理システムを開発してきた⁶⁾⁻⁸⁾。本論文では、このシステムにおけるスケジュール作成・調整のために新しく開発した、負荷山崩しや調整の標準アルゴリズムを幾つか持ち、ノウハウに従ってそれらを組み合わせて山崩しを行う新スケジュールリング方式を提案する。

以下、第2章で、プロジェクト管理システムの基本的考え方と構成を述べ、第3章では、対象となるスケジュールリング問題を定義する。第4章では、新しく開発した計画立案業務における作業スケジュールリング方式とその適用例を述べ、第5章では、進捗管理業務に

† Knowledge Based Project Scheduling Method for Project Management System of Information Processing System Development by MICHIKO OBA, ISAO TSUSHIMA, NORIHISA KOMODA (Systems Development Laboratory, Hitachi, Ltd.) and TOSHIRO YAMANAKA (Omori Software Works, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所

††† (株)日立製作所大森ソフトウェア工場

における再スケジューリング方式と適用例を述べ、その有効性を示す。

2. プロジェクト管理システム

1つのシステムの開発は、いくつかのサブシステムに分けて行われる。さらに、各サブシステムの開発は、いくつかのフェーズ（例えば、システム設計、プログラム設計、プログラム作成、組合せテスト、総合テスト）に従って進められる。

2.1 プロジェクト管理の知識

プロジェクト管理業務においては、以下のような知識が存在する。これらは、各プロジェクト管理者のノウハウであり、対象プロジェクトの特性の違いや開発ツールの進歩等に応じて異なり、追加や変更も発生する。そこで、これらのノウハウを統一的に表現するために、IF-THEN形式のルールで記述し、知識として蓄え、利用する方法を採用した。

(1) 工数・期間の見積り知識

プロジェクトの対象業務は多岐にわたり、工数は、開発規模、開発環境、対象業務の特性など多種類の要因に依存する。したがって、標準的な工数・期間のデータを一律に適用することはできない。そこで、サブシステムの開発規模（ステップ数）に対する標準的な基本となる工数・期間のデータ（見積り基本データ）および標準的な資源配分率をプロジェクトの特性に応じて修正し、工数、期間を見積もる。この修正ノウハウは、「どのようなプロジェクト特性のときに、何をどのように修正するか」を図1に示すようなルールで表す。

(2) 負荷山崩し手段の適用知識

見積もった工数・期間を日程計画へ展開するためには、工数と期間の相互関係を考慮しながら、確保可能な人員内に人員負荷を納めるための負荷山崩しを行

```

[期間修正ルール]
(1) if [対象業務] は [事務処理]
    then 見積り期間を [ 0 ] % 増
(2) if [対象業務] は [在庫管理]
    then 見積り期間を [ 15 ] % 増
    :
[工数修正ルール]
(1) if [開発場所] は [1箇所での開発]
    then 見積り工数を [ 0 ] % 増
(2) if [開発場所] は [複数箇所での開発]
    then 見積り工数を [ 20 ] % 増
    :
    
```

図1 修正ルールの記述例

Fig. 1 An example of standard parameter modification rules.

```

(1) if [初期負荷山積み] を実行
    か 結果は、 [NG]
    か ピーク負荷時のサブシステムは [X]
    then サブシステム [X] に [右シフト機能] を適用
        ; 調整量 [0] ヶ月
(2) if サブシステム [X] に [右シフト機能] を適用
        ; 調整量 [0] ヶ月
    か 結果は、 [NG]
    か ピーク負荷時のサブシステムは [X]
    then サブシステム [X] に [期間延長機能] を適用
        ; 調整量 [1] ヶ月
(3) if サブシステム [X] に [期間延長機能] を適用
        ; 調整量 [1] ヶ月
    か 結果は、 [NG]
    か ピーク負荷時のサブシステムは [X]
    then サブシステム [X] に [期間短縮機能] を適用
        ; 調整量 [1] ヶ月
(4) if サブシステム [X] に [期間短縮機能] を適用
        ; 調整量 [1] ヶ月
    か 結果は、 [NG]
    か ピーク負荷時のサブシステムは [X]
    then サブシステム [X] に [期間延長機能] を適用
        ; 調整量 [2] ヶ月
    :
    
```

図2 負荷山崩し方針ルールの記述例

Fig. 2 An example of load balancing algorithm select rules.

う。この負荷山崩しの手段は複数あり、負荷山崩し手段を適用するための方針ノウハウは、「どのような場合に、どの負荷山崩し手段を実行するか」を図2に示すようなルールで表す。負荷山崩しの手段として、単に開始時刻を遅らせる右シフト機能だけではなく、開発期間を延長または短縮したときの工数を求め、それぞれの変更値に対して負荷山崩しを行う期間延長操作機能、期間短縮操作機能がある。

(3) 進捗状況の診断知識

プロジェクトの進捗状況を的確に把握し、問題点を早期に発見するためには、日程計画に対する進捗の遅れ、投入人員の山積みの予実績差、開発フェーズ完了ごとに見直される開発規模の変化、開発体制、完成プログラムの品質、ツール、端末数等の開発環境、レビ

```

(1) if 診断対象サブシステムの現在工程は [X]
    か フレーム名 [Y] の工程区分が [X] で
        診断区分が [管理体制] で
        解答が [なし] である
    then フレーム名 [Y] の質疑応答を実行
(2) if 診断対象サブシステムの現在工程は [X] で
        開発規模の現状が [良好] でなく
        期間現状が [良好] でない
    か フレーム名 [Y] の工程区分が [X] で
        診断区分が [原因究明] で
        開発規模の現状が [良好] でなく
        期間現状が [良好] でない
        解答が [なし] である
    then フレーム名 [Y] の質疑応答を実行
    :
    
```

図3 診断ルールの記述例

Fig. 3 An example of project diagnosing rules.

- ```

(1) if 余裕期間が[なし]
 かつ 遅れが[余裕期間オーバー]
 かつ 現在の工程は [システム設計]
 then 適用機能の優先度 [1] は [未着手工程への人員投入機能]
 適用機能の優先度 [2] は [現在工程への人員投入機能]
 適用機能の優先度 [3] は [仕様変更]
 適用機能の優先度 [4] は [納期変更]
(2) if 余裕期間が[あり]
 かつ 遅れが[余裕期間オーバー]
 かつ 現在の工程は [プログラム設計]
 then 適用機能の優先度 [1] は [余裕期間での遅れ吸収機能]
 適用機能の優先度 [2] は [人員投入+余裕期間での遅れ吸収機能]
 適用機能の優先度 [3] は [未着手工程への人員投入機能]
 適用機能の優先度 [4] は [現在工程への人員投入機能]

```

図 4 調整機能適用順序ルールの記述例

Fig. 4 An example of rescheduling algorithm ordering rules.

ユーやドキュメントの整備状況等の管理項目、等の多面的な角度からプロジェクトを診断する必要がある。そこで、各診断項目に関する各診断要因の因果関係をフレームで表現し、各診断項目に関する質問の制御と評価は、ルールにより行う。ルールの例を図 3 に示す。

(4) スケジュール調整知識

スケジュール調整では、人の追加手配が容易でない等のため、できる限り小規模な範囲の調整によって、発生している遅れを解消し、納期内にすべての作業を

- ```

(1) if  スケジュール調整機能は [未着手工程への人員投入機能]
      かつ 現在の工程は [設計フェーズ]
    then 調整工程候補は [プログラミングフェーズ] で優先度 [1]
         調整工程候補は [テストフェーズ] で優先度 [2]
(2) if  スケジュール調整機能は [未着手工程への人員投入機能]
      かつ 現在の工程は [プログラミングフェーズ]
    then 調整工程候補は [テストフェーズ] で優先度 [1]
(3) if  スケジュール調整機能は [現在工程への人員投入機能]
      かつ 現在の工程は [設計フェーズ]
    then 調整工程候補は [設計フェーズ] で優先度 [1]
  
```

図 5 調整工程選択ルールの記述例

Fig. 5 An example of rescheduled jobs selecting rules.

終了させる必要がある。このスケジュール調整の手段として、人員投入による遅れ解消や余裕期間での遅れ吸収等の調整機能があり、その適用に関する知識は、「どのような進捗状況のときに、どの優先順位でスケジュール調整機能を適用するのが望ましいか」を図 4 に示すようなルールで表す。また、各スケジュール調整機能での調整工程の選択知識は、「どのような進捗状況で、スケジュール調整機能を適用するときに、どの優先順位で開発フェーズを調整するのが望ましいか」を図 5 に示すようなルールで表す。

2.2 プロジェクト管理システムの概要

プロジェクト管理システムは、計画立案業務を支援する計画立案サブシステムと、実施段階の進捗管理業務を支援する進捗管理サブシステムとから構成する。本システムの構成を図 6 に示す。

見積り処理部では、計画対象プロジェクトのプロジェクト名、開始予定日、納期、難易度・信頼性の度合い、利用形態(オンライン/バッチ)等のプロジェクトの特性、各サブシステムの開発規模等の入力データをシステムに取り組み、工数・期間を見積もる。この見積り方式は次章で説明する。

作業割り付け処理部では、見積り処理部で見積もった工数・期間を日程計画表へ展開する。日程計画表への展開では、まず、人員制限を考慮せず、各開発フェーズの実施順序の規制に基づいて、初期の負荷山積みを行う。つぎに、負荷山崩し方針ノウハウに従って負荷山崩し手段を実行し、日程計画表を作成する。各負荷山崩し手段の実行は、手続き型言語で記述された実行モジュールで行う。この負荷山崩し方法については、第 4 章で説明する。

診断処理部では、各診断項目に関して、対応する診断ノウハウにより質疑応答を繰り返し、プロジェクト

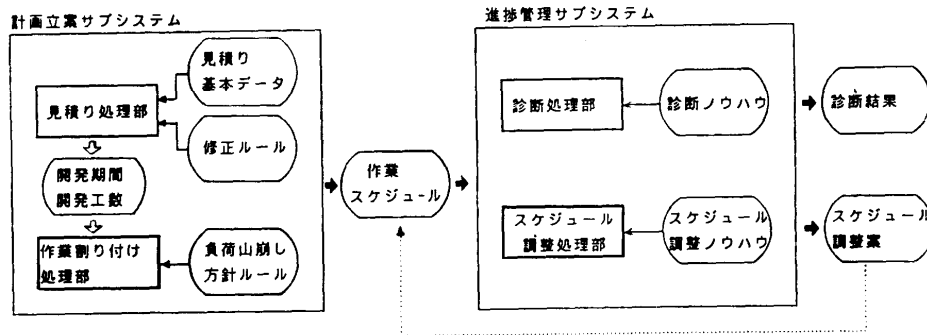


図 6 プロジェクト管理システムの構成

Fig. 6 Configuration of a project management system.

の状況を診断する。遅れ、見積り誤差等の問題点が検知された場合は、さらに、質疑応答によって原因を究明する。また、今後、問題を引き起こす可能性のある原因も洗い出す。

スケジュール調整処理部では、日程計画の遅れに対する再スケジューリングを行う。本システムでは、人員を投入して遅れを回復するか、余裕期間を使って遅れを吸収するかの方針は、諸般の事情を考慮して人間が指定する。その際、スケジュール調整方針のガイダンスを調整機能適用順序ルールを使って表示する。システムは、指定された方針のもとで、調整工程選択ルールで選択された開発フェーズのスケジュールを調整する。これにより、利用者は試行錯誤的な判断を効率的に行えるようになる。スケジュールの調整は、各調整機能ごとに手続き型言語で記述された実行モジュールで行う。このスケジュール調整方法は第5章で説明する。

3. 作業スケジューリング問題の定義

日程計画表への展開における作業スケジューリング問題の定義を以下に示す。

3.1 入出力変数と評価関数

(1) 入力変数

(a) 人員制限 確保可能な人員数の時間分布を

$$r(t) \quad \text{for } t_1 \leq t \leq t_2$$

とする。なお、 $t_1 \sim t_2$ は、プロジェクトの実施可能時間帯である。

(b) サブシステムと開発フェーズ 各サブシステム i の開発規模を s_i 、 j 番目の開発フェーズを p_{ij} 、その集合を I とする。サブシステム間および各サブシステム内での開発フェーズの実施順序には規制があり、それらは、PERT 図⁹⁾で表現できる。また、各サブシステム i には、着手可能日 e_i ($t_1 \leq e_i \leq t_2$) と納期 d_i ($t_1 \leq d_i \leq t_2$) がある。

(2) 出力変数

集合 I に属するすべての開発フェーズ $p_{ij} \in I$ の開始時刻 τ_{ij} を決め、その時の時刻 t ($t_1 \leq t \leq t_2$) での人員負荷 $L^{(i)}(t)$ を求める。なお、開始時刻 τ_{ij} を決めることを割付けと呼ぶことにする。

(3) 評価関数

負荷山崩しでは、人員負荷が人員制限内に納まっているか否かを次式により判定する。

$$r(t) - L^{(i)}(t) \geq 0 \quad \text{for all } t \in [t_1, t_2] \quad (1)$$

(1)式を満足できない場合には、次式で示す人員負荷と人員制限の差の2乗和 S を評価関数とし、その最小化を目的とする。

$$S = \sum_{t=t_1}^{t_2} \{r(t) - L^{(i)}(t)\}^2 \rightarrow \min. \quad (2)$$

なお、上式は、人員負荷のばらつきを少なくし、人員制限をオーバする人員負荷が可能な限り少ない人員数になるとして、PERT 等で経験的に利用されているものである。

3.2 開発期間と工数に関する標準的な関係

(1) 見積り基本データ

見積り基本データは、標準的な特性を持つプロジェクトに対し、標準的な能力を持つ人員が、ソフトウェアを開発するのに要する開発期間および工数の関数である。なお、開発規模 s に対応して、開発期間設定の基準となる期間を基準期間 x^* とし、関数 $x^* = \text{STD}(s)$ で設定する。Basic COCOMO モデル¹⁾の値を基に想定した見積り基本データの例を図7に示す。

(2) 標準的資源配分率

サブシステム i の開発期間 x_i および工数 m_i を、開発フェーズ j ($1 \leq j \leq h_i$, h_i はサブシステム i の総フェーズ数) の期間 PD_{ij} 、工数 PM_{ij} に配分する標準的な配分率をそれぞれ RD_{ij} 、 RM_{ij} と表すと、以下の関係式を満足する。

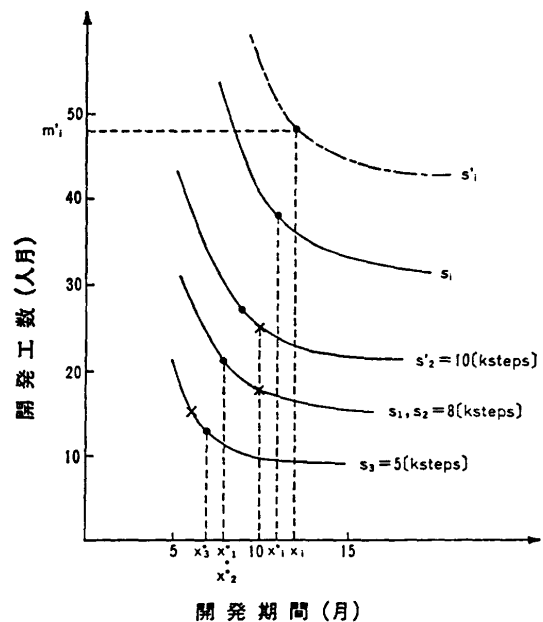


図7 見積り基本データの例

Fig. 7 An example of basic estimation data of man-months and duration.

$$PD_{ij} = RD_{ij} \times x_i \quad (3)$$

$$PM_{ij} = RM_{ij} \times m_i \quad (4)$$

4. 作業スケジュールリング方式

4.1 開発期間と工数の見積り方式

工数は開発期間により変化する。このため、負荷山崩しにおいて期間を延長あるいは短縮する度に、改めて工数の見直しが必要である。その際、標準的なソフトウェア開発に対する値は、プロジェクトの特性により補正することが必要である。この開発期間ならびに工数の見積り方法を、以下に説明する。

(ステップ1) 開発期間、工数の見積り

① 開発規模 s_i に対する基準期間 x_i^* を求める。
 $x_i^* = \text{STD}(s_i)$

② 基準期間 x_i^* に対して期間修正ルールを適用して開発期間 x_i を求める。納期条件が厳しく、 $x_i > d_i - e_i$ となる場合は、納期条件を優先させて次式により開発期間 x_i を求める。

$$x_i = d_i - e_i \quad (5)$$

③ x_i を基準期間とする開発規模 s'_i ($=\text{STD}^{-1}(x_i)$) に対する仮想的な工数 m'_i を求める。この仮想的な工数 m'_i に対して、工数修正ルールを適用して、最終的な工数 m_i を求める。

(ステップ2) 各開発フェーズの開発期間、工数、所要人員数の見積り

① x_i を基準期間とする開発規模 s'_i ($=\text{STD}^{-1}(x_i)$) に適した標準配分率 RDs'_{ij} , $RM_s'_{ij}$, $j=1 \sim h$ を選ぶ。

② これらの標準的配分率に、資源配分修正ルールを適用して、各資源配分率 RD_{ij}' , RM_{ij}' を求める。これを用いて、各開発フェーズ p_{ij} における開発期間 PD_{ij} , 工数 PM_{ij} を(3), (4)式を用いて算出する。

③ サブシステム i の各開発フェーズ p_{ij} での所要人員数 PP_{ij} を計算する。

$$PP_{ij} = [PM_{ij}/PD_{ij}] \quad (6)$$

4.2 負荷山崩しアルゴリズム

① 人員制限 $r(t)$ を除く制約条件下で、PERT/TIME 計算⁹⁾により各開発フェーズ p_{ij} の最早開始可能時刻 t_{ij} を求め、開始時刻 τ_{ij} とする。各時刻の人員負荷 $L^{(t)}(t)$ を次式により求める。

$$L^{(t)}(t) = \sum_{p_{ij} \in I} (PP_{ij} | \tau_{ij} \leq t \leq \tau_{ij} + PD_{ij}) \quad (7)$$

$$; t_1 \leq t \leq t_2$$

次に、以下で本アルゴリズムの収束判定に利用するパラメータ S_0 に十分大きい値を初期値として代入する。

② 人員負荷が人員制限内に納まっているか否かを(1)式により判定する。もし、(1)式が成立するならば、処理を終了する。そうでなければ、(2)式により S を計算し、 $S_0 - S > \varepsilon$ なら、 $S_0 := S$ として③へ。 $S_0 - S \leq \varepsilon$ なら、処理を終了する。なお、 ε は、改善の度合いの下限値を示すパラメータ値である。

③ 負荷山崩し機能適用の対象となるサブシステムとして、未評価のサブシステムの中から、次式を満たすものを選ぶ。

$$r(t) - L^{(t)}(t) < 0 \text{ かつ}$$

$$\max \{ |r(t) - L^{(t)}(t)| \}, t_1 \leq t \leq t_2$$

t で実施されるサブシステムが複数個ある場合は、自由余裕の最も少ないサブシステムを選ぶ。そのサブシステム番号を i^* とする。負荷山崩し機能適用の対象となるサブシステムがない場合は、処理を終了する。

④ 負荷山崩しルールにおいて、各ルールの IF 部の条件に書かれている適用機能のパラメータ値と負荷山崩し実行中の適用機能の内容を比較する。条件を満足するルールの THEN 部に記述されている負荷山崩し機能が、延長または短縮の場合、4.2 節の方式により、サブシステム i^* における各開発フェーズ p_{i^*j} の開発期間 PD_{i^*j} , 工数 PM_{i^*j} , 所要人員数 PP_{i^*j} を求め、⑤へ。右シフト機能の場合、そのまま⑥へ。

条件を満足するルールがなければ③へ戻る。

⑤ PERT/TIME 計算により、サブシステム i^* の各開発フェーズ p_{i^*j} を最早開始時刻 t_{i^*j} に割り付ける。

⑥ サブシステム i^* の各開発フェーズ p_{i^*j} の新たな開始時刻 $\bar{\tau}_{i^*j}$ を次式により求める。

$$\bar{\tau}_{i^*j} := \{d | \min_{\tau_{i^*j} \leq d \leq \tau_{i^*j} + FF_{i^*j}} \{r(t) - L^{(t)}(t) - L^{(t)}(p_{i^*j})(t) - L^{(t)}(p_{i^*j})(t, d)\}^2\}$$

ただし、 $L^{(t)}(p_{i^*j})(t, d)$ は、 p_{i^*j} の開始時刻を d としたときの時刻 t での p_{i^*j} の人員負荷であり、 $d \leq t \leq d + PD_{i^*j}$ のとき PP_{i^*j} となり、それ以外は 0 である。なお、 FF_{i^*j} は p_{i^*j} の自由余裕である。

⑦ $\bar{\tau}_{i^*j}$ を τ_{i^*j} に代入し、人員負荷 $L^{(t)}(t)$ を(7)式により算出して、自由余裕を変更後、②へ。

4.3 計画例

提案手法を、表1に概要を示すプロジェクトに適用した計画例を示す。プロジェクトは、開発場所が複数箇所での開発であり、それ以外は基準特性とする。プロジェクトの実施期間は1月から12月であり、各月の確保可能な人員数は図8の点線で示す。サブシステム1と2は、1月から着手可能であるが、サブシステ

表 1 計画対象データ
Table 1 A project profile.

サブシステム 番号; i	開発規模 s_i (Ks)	基準期間 x^*_i (月)	標準期間配分率 (%)			標準工数配分率 (%)			納期	実施順序制約
			$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$		
1	8	8	36	32	32	25	40	35	12/31	サブシステム1の設計フェーズ ($j=1$)の終了後に着手可能
2	8	8	36	32	32	25	40	35	10/31	
3	5	7	35	38	27	30	43	29	12/31	

表 2 初期山積みデータ
Table 2 Initial loading data.

サブシステム 番号; i	初期見積り期間 x_i (月)	初期見積り工数 m_i (人月)	フェーズ別開発期間 (月)			フェーズ別開発工数 (人月)			フェーズ別所要人員 (人)		
			$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$
1	8	25.2	2.9	2.6	2.6	5.1	10.1	8.9	2	4	4
2	8	25.2	2.9	2.6	2.6	5.1	10.1	8.9	2	4	4
3	7	15.6	2.5	2.7	1.9	4.7	6.8	4.6	2	3	3

表 3 負荷山崩し結果データ
Table 3 Load balancing data.

サブシステム 番号; i	調整後期間 x_i (月)	調整後工数 m_i (人月)	フェーズ別開発期間 (月)			フェーズ別開発工数 (人月)			フェーズ別所要人員 (人)		
			$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$
1	10	21.7	3.6	3.2	3.2	5.4	8.7	7.6	2	3	3
2	10	21.7	3.6	3.2	3.2	5.4	8.7	7.6	2	3	3
3	6.1	18.5	2.1	2.3	1.7	5.4	7.8	5.3	3	4	4

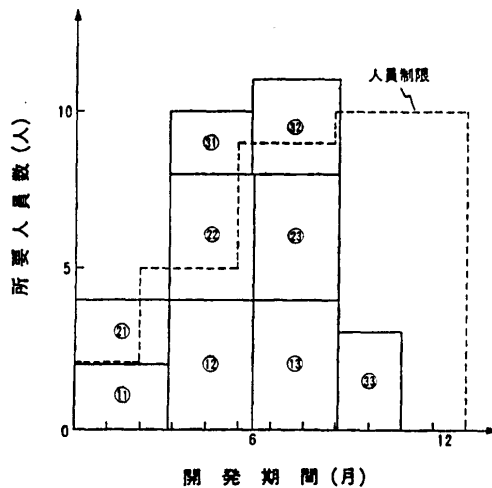


図 8 初期山積み結果

Fig. 8 Result of initial loading at the earliest executable time.

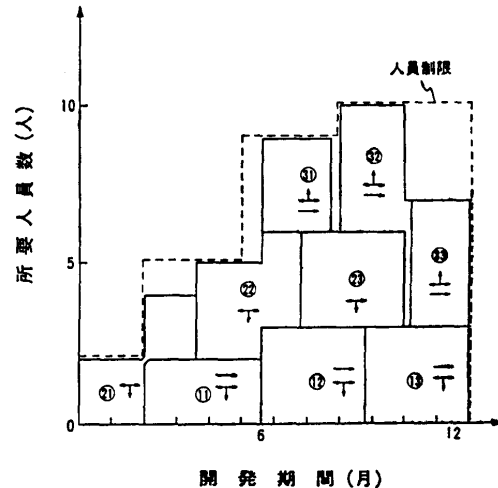


図 9 負荷山崩し結果

Fig. 9 Result of load balancing.

ム3はサブシステム1の設計フェーズ ($j=1$) 終了後に着手可能とする。また、サブシステム2は、納期が10月とする。

提案方式により、図1の修正ルールを用いて見積も

った値を表2に示す。次に、初期山積み結果を図8に示す。図中の“①”は、サブシステム i の j 番目の開発フェーズの作業を示す。図2の負荷山崩し方針ルールを用いて負荷山崩しを行った結果を図9および表3

に示す。図中の“→”, “⇄”, “⇆”は、それぞれ初期山積み時に割り付けた実施時刻を負荷山崩して右シフトした作業、開発期間を基準期間より延長または短縮した作業である。提案方式によれば、工数と期間の従属関係を考慮しながら、所定の人員内に人員負荷を納めることができる。

5. 再スケジューリング方式

5.1 再スケジューリング問題の定義

遅れ解消のための再スケジューリング問題は、3章の作業スケジューリング問題と基本的に同じであるが、以下の点が異なる。

(1) 入力変数

すでに終了した開発フェーズ p_{ij} の集合を J 、現在進行中の開発フェーズ p_{ij} の集合を K 、未着手の開発フェーズ p_{ij} の集合を N とする。

(a) 追加投入人員 各開発フェーズ p_{ij} の追加投入可能人員数を $PP_{add_{ij}}$ とする。

(b) 現在月日を T_{now} とする。

(c) 開発規模 各サブシステム i の当初予想開発規模を s^0_i 、修正された予想開発規模を s_i とする。

(d) 実績スケジュール 各開発フェーズ p_{ij} ($p_{ij} \in J \cup K$) の開始時刻の実績を $\tilde{\tau}_{ij}$ とする。また、現在の開発フェーズ p_{ij} ($p_{ij} \in N$) の作業の進捗度を w_{ij} とする。

(e) 作業スケジュール 各サブシステム i の当初予想開発規模 s^0_i に対する開発期間、工数は、それぞれ x^0_i 、 m^0_i であり、各開発フェーズ p_{ij} ($p_{ij} \in I$) の開始時刻、開発期間、工数、所要人員数は、それぞれ τ^0_{ij} 、 PD^0_{ij} 、 PM^0_{ij} 、 PP^0_{ij} である。時刻 t の人員負荷を $L^{(KUN)}(t)$ とする。

(2) 出力変数

(a) 開発フェーズ p_{ij} ($p_{ij} \in N$) の開始時刻 τ_{ij} 、および開発フェーズ p_{ij} ($p_{ij} \in K \cup N$) の開発期間 PD_{ij} 、所要人員数 PP_{ij} を決める。

(b) 開発フェーズ p_{ij} ($p_{ij} \in K \cup N$) を割り付けたときの時刻 t の人員負荷 $L^{(KUN)}(t)$ を求める。

$$L^{(KUN)}(t) = \sum_{p_{ij} \in K \cup N} (\text{時刻 } t \text{ に行われている } p_{ij} \text{ の } PP_{ij}) ; T_{now} \leq t \leq t_2 \quad (8)$$

(3) 評価関数

サブシステム i の最終開発フェーズ p_{i^*j} がその納期内に終了するか否かを次式により判定する。ここで、 $t^{E_{ij}}$ は最終開発フェーズ p_{ij} の終了時刻である。

$$\max t^{E_{ij}} \leq d_i \quad (9)$$

5.2 スケジュール調整アルゴリズム

① 調整済みフェーズの集合 Q を、 $Q := \emptyset$ とする。

② サブシステム i の新しい予想開発規模 s_i が $s^0_i < s_i$ のときは、新しい予想開発規模 s_i に対して、サブシステム i の当初開発期間 x^0_i での工数 m_i と各開発フェーズ p_{ij} ($p_{ij} \in K \cup N$) の工数 PM_{ij} を3.2節の方式により求める。未着手の開発フェーズ p_{ij} ($p_{ij} \in N$) を当初予定の所要人員数 PP^0_{ij} で開発した際の開発期間 PD_{ij} を次式により計算する。

$$PD_{ij} = PM_{ij} / PP^0_{ij} \quad (10)$$

$s^0_i \leq s_i$ のときは、各開発フェーズ p_{ij} ($p_{ij} \in K \cup N$) の開発期間 PD_{ij} 、工数 PM_{ij} 、所要人員数 PP_{ij} は、当初予定の値と同じとする。

③ 現在の開発フェーズ p_{ij} の残り開発期間 PD_{ij} 、終了時刻 $t^{E_{ij}}$ を次式により計算する。

$$PD_{ij} = (T_{now} - \tilde{\tau}_{ij})(1 - w_{ij}) / w_{ij}$$

$$t^{E_{ij}} = T_{now} + PD_{ij} \quad (11)$$

④ スケジュール調整機能適用の対象として、次式を満たす開発フェーズ p_{ij} ($p_{ij} \in K \cup N$) を含むサブシステムを1つ選び、そのサブシステム番号を i^* とする。

$$\min \{t^{E_{ij}} | FF_{ij} < 0, p_{ij} \in Q\}$$

スケジュール調整機能適用対象のサブシステムがない場合は、処理を終了する。

⑤ 調整機能適用順序ルールに基づき、スケジュール調整機能の望ましい適用順序を表示し、利用者に適用するスケジュール調整機能を選択させる。

⑥ 選択されたスケジュール調整機能が“余裕期間での遅れ吸収機能”の場合、⑦へ。“現在工程での遅れ回復機能”の場合、 $W := w_{i^*j}$ として⑧へ。“未着手工程での遅れ回復機能”の場合、 $W := 0$ として⑧へ。

⑦ PERT/TIME 計算により未着手の開発フェーズ p_{ij} ($p_{ij} \in N$) の新たな開始時刻 τ_{ij} 、終了時刻 $t^{E_{ij}}$ を求め、(8)式により人員負荷 $L^{(KUN)}(t)$ を算出する。

⑧ 納期内にプロジェクトが終了するか否かを(9)式により判定する。上式が成立するならば処理を終了する。成立しなければ④へ。

⑨ 調整工程選択ルールにより、調整対象の候補となる開発フェーズとその優先順位の組 $[a, b]$ を求める。

⑩ 調整対象の開発フェーズ $p_{i^*j^*}$ として、 Q に含まれないものの中から、優先順位の最小のものを選び、集合 Q の要素に加える。 Q に含まれない候補がな

表 4 調整対象データ
Table 4 An example of current project status.

サブシステム番号; i	開発規模 s_i (Ks)	現在の開発フェーズ	進捗度 w_{ij}	現在フェーズの着手日; r_{ij}	追加投入人員 (人)		
					$j=1$	$j=2$	$j=3$
1	8	$j=1$	85(%)	2月1日	1	2	1
2	10 (+2)	$j=1$	30	5月1日	1	1	1
3	5	未着手	—	—	1	3	2

() 内は当初計画との差異

表 5 スケジュール調整結果データ
Table 5 Rescheduling result.

サブシステム番号; i	調整後期間 x_i (月)	調整後工数 m_i (人月)	フェーズ別開発期間 (月)			フェーズ別開発工数 (人月)			フェーズ別所要人員 (人)		
			$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$
1	10	21.7	3.6	3.2	3.2	5.4	8.7	7.6	2	3	3
2	9.8 (-0.2)	30.5 (+8.8)	4.0 (+0.4)	3.1 (-0.1)	2.7 (-0.5)	8.0 (+2.6)	12.0 (+3.3)	10.5 (+2.9)	2	4 (+1)	4 (+1)
3	6.1	18.5	2.1	2.3	1.7	5.4	7.8	5.3	3	4	4

() 内は当初計画との差異

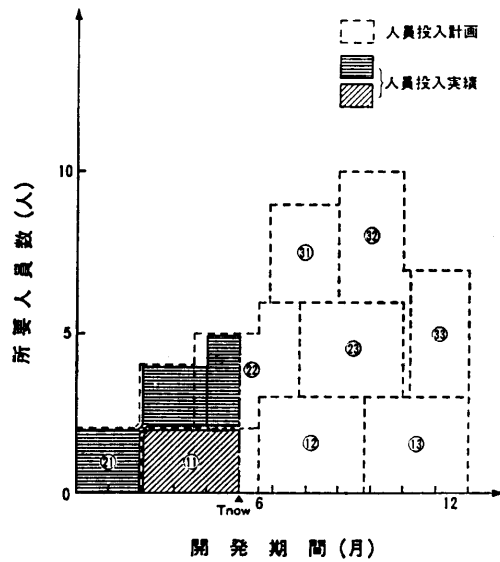


図 10 人員投入計画と人員投入実績

Fig. 10 Comparison between man schedule and result.

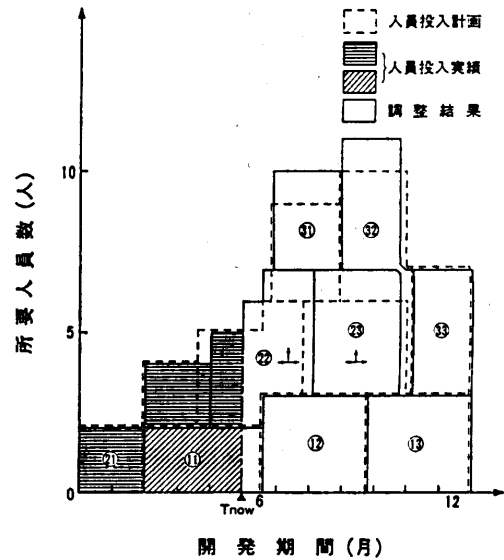


図 11 スケジュール調整結果

Fig. 11 Result of reschedule.

い場合は、④へ。

⑪ 開発フェーズ $p_{i,j}$ での人員投入による短縮後の開発期間 $PD_{i,j}$ と所要人員数 $PP_{i,j}$ を次式により求める、

$$PD_{i,j} = PM_{i,j} \times (1 - W) / (PP^0_{i,j} + PPadd_{i,j}) \quad (12)$$

$$PP_{i,j} = PP^0_{i,j} + PPadd_{i,j} \quad (13)$$

⑫ ⑦と同様の処理を実行した後、(9)式が成立するならば処理を終了する。成立しなければ⑩へ。

5.3 調整例

提案手法を、3.5節で計画した作業スケジュールに対し、表4に進捗状況および追加投入人員等の調整対象を示すプロジェクトに適用した調整例を示す。

現在の日付は、6月1日である。サブシステム1

は、現在、設計フェーズ ($j=1$) で、進捗度は 85% である。サブシステム 2 の開発規模は、当初予定の 8K ステップに対し、10K ステップとなっており、現在は、プログラミングフェーズ ($j=2$) で、進捗度は 30% である。サブシステム 3 は未着手である。各サブシステムの人員投入計画および人員投入実績を図 10 に示す。

提案方式により、図 5 の調整機能適用順序ルールが適用され、利用者は“未着手工程への人員投入機能”および“現在工程への人員投入機能”を選択したときのスケジュール調整結果を図 11 および表 5 に示す。図中の“④”は、サブシステム i の j 番目の開発フェーズの作業を示す。図中の“△”は、再スケジュールリング前の作業スケジュールに対し、スケジュール調整で追加人員を投入した作業である。提案方式によれば、工数と期間の従属関係を考慮しながら、所定の納期内にすべての作業を終了することができる。

6. むすび

プロジェクト管理者の持つ工数の見積りや日程計画表への展開、あるいは工程遅れ解消のためのスケジュール調整に関するノウハウを組み込める新しいスケジュールリング方式を提案した。本方式を組み込んだプロジェクト管理システムは、開発工数が 50 人月から 200 人月程度、開発期間が 2 年未満程度の比較的小規模な情報システム開発プロジェクトを対象に適用され、実用的な解を得られることを確認している。また、プロジェクト管理者のプロジェクト管理業務における計画・管理時間の短縮、管理ノウハウの蓄積と有効活用が図れるという効果が得られている。

謝辞 本研究の機会を与えていただいた(株)日立製作所システム開発研究所 堂免信義所長、春名公一博士、中尾和夫博士に感謝いたします。また、プロジェクト管理システムの開発にあたり御指導、御協力をいただいた同・大森ソフトウェア工場および同・ソフトウェア工場の方々に感謝いたします。

参 考 文 献

- 1) Boehm, B. W.: *Software Engineering Economics*, Prentice-Hall, Englewood Cliff, NJ (1981).
- 2) Putnum, L. H.: A General Empirical Solution to the Macro Software Sizing and Estimating Problem, *IEEE Trans. Softw. Eng.*, Vol. SE-4, No. 4, pp. 345-361 (1978).
- 3) 平ほか: プロジェクト管理支援エキスパートシステム“NINA”, 情報処理学会「知識工学と人工知能研究会報告」, 87-A1-55-2, pp. 9-16 (Nov. 1987).
- 4) 横山ほか: ソフトウェア管理診断システム SOMEX, 人工知能学会全国大会 (第 1 回), pp. 303-306 (Oct. 1987).
- 5) 松本ほか: プロジェクト管理エキスパートシステム, 信学技法, CPSY 86-62, pp. 41-52 (1986).
- 6) 大場ほか: 知識工学を応用したプロジェクト計画方式, 第 35 回情報処理学会全国大会論文集, pp. 1131-1132 (Oct. 1987).
- 7) 大場ほか: 知識工学を応用したプロジェクト進捗管理方式, 第 36 回情報処理学会全国大会論文集, pp. 1103-1104 (Mar. 1988).
- 8) 大場ほか: プロジェクト管理支援エキスパートシステム, 第 14 回システムシンポジウム, pp. 215-218 (Aug. 1988).
- 9) 刀根 薫: PERT 講座, 東洋経済新報社 (1977).
(昭和 63 年 10 月 27 日受付)
(平成元年 2 月 14 日採録)



大場みち子 (正会員)

昭和 57 年日本女子大学家政学部家政理学科 (物理学専攻) 卒業。同年、(株)日立製作所入社。システム開発研究所にて、事象駆動型システム制御、ソフトウェア開発プロジェクトの計画・管理システムに関する研究に従事。計測自動制御学会会員。



都島 功 (正会員)

昭和 20 年生。昭和 43 年名古屋工業大学計測工学科卒業。昭和 45 年大阪大学基礎工学部制御工学科修士課程修了。同年(株)日立製作所入社、中央研究所配属。昭和 48 年システム開発研究所の発足に伴い転属。現在、第一部主任研究員。一貫して、物流システムの計画、管理、制御技法に関する研究に従事。IEEE、電気学会、計測自動制御学会各会員。工学博士。



薦田 憲久 (正会員)

昭和 25 年生. 昭和 47 年大阪大学工学部電気工学科卒業. 49 年同大学院修士課程修了. 同年(株)日立製作所に入社. システム開発研究所にてシステム計画技法, システム構造化技法, ペトリネットなどの事象駆動型システム, 生産・流通業向情報処理システムなどに関する研究に従事. 現在同研究所第 1 部主任研究員. 56~57 年 UCLA に留学. 工学博士. IEEE, 電気学会, 電子情報通信学会, 計測自動制御学会などの会員. 61 年度計測自動制御学会論文賞, 62 年度計測自動制御学会技術賞受賞.



山中止志郎 (正会員)

昭和 45 年東京大学医学部保健学科卒業. 同年, (株)日立製作所情報システム研究所に入所. 現在, 大森ソフトウェア工場主任技師. システムエンジニア用支援ツールの開発に従事.