

個人の能力差を考慮した開発における 人員資源割当問題に対する解法の提案

高須賀 将秀^{†1} 吉田 寛^{†1}

概要：本研究では個人の能力差を考慮した開発における人員資源割当問題のモデル提案と本モデルにおける解法の提案を行う。EMS 開発のような大規模かつ複雑なシステム開発では、多数のタスクが存在し、その各々のタスクは専門性を要するものが多い。しかしながら、既存で研究されているモデルでは、個人の能力差は考慮されておらず、各々のタスクの難しさは誰が実行しても同じと仮定しており、その結果、開発前に算出する期間や稼働は実績とズレが生じている。そこで、高須賀ら^{†1}は各々のタスクの処理時間を可変値とすることで、より現実問題に近い実開発の期間や稼働の算出精度を上げるが可能な個人の能力差を考慮したモデルを示してきた。本研究では、そのモデルに対して現実的時間で解くことのできる多項式時間アルゴリズムを提案する。

キーワード：最適化問題，NP 困難，人員資源割当問題，アルゴリズム，能力差，システム開発

Proposal of the solution to the personnel resource allocation problems in the development that takes into account the ability difference

Masahide Takasuka^{†1} Hiroshi Yoshida^{†1}

Abstract: In this research, we propose a model proposal of personnel resource allocation problem in the development that takes into account the ability difference and suggest solution method in this model. In large-scale and complicated system development such as EMS development, there are many tasks. Many of those tasks require expertise, however, in existing models being studied, individual capacity differences are not taken into consideration and assumption is made that the difficulty of each task is the same regardless of who executes. As a result, the period and operation to calculate before development are occurring and the deviation has occurred. Therefore, by setting the processing time of each task to be a variable value, we realize a model that takes into consideration differences in capabilities of individuals and raise the accuracy of the actual development period and operation that is closer to the real problem, and also describe the polynomial time algorithm that can be solved in real time at the same time.

Keywords: Optimization problem, NP-hard, Personnel resource allocation problem, Algorithm, Difference in ability, System development

1. はじめに

EMS(Element Management System)開発では、装置機能部、アプリケーション部、ミドルウェア部等の機能群の開発を行う必要がある。その各機能群の開発を遂行するにあたり、各機能群の開発を得意とする人員を適材適所に割り当てることで、スケジュール短縮やコスト削減を図った EMS 開発を行うことが可能になる。これまで、複数種類の工程がある業務を行う際、各工程のタスクをどのように人員に割り当てると効率化できるかについての問題をモデル化した。本稿では、そのモデルに対しての多項式時間で解くことのできるアルゴリズムを提案する。

2. 背景

各タスクをある一定以上の品質を担保するために必要なタスク処理時間は各々の人の能力に依存する。例えば、あるタスクを実行するのに、人 A は得意なタスクであるため 1 の処理時間を要するのに対し、人 B は不得意なタスクであるため 8 の処理時間を要する、また、人 C はスキル未取得であるため、タスクを処理することはできないといったケースの場合、人 A、人 B、人 C のいずれの人物にもタスクが割り当てられていないという条件下では、タスクを人 A に任せるのが最適である。このようにあるタスクに対し、それを処理するのが得意な人を極力割り当てるようにし、全タスクの終了時刻が短く、かつ人員資源が最小となることを目指す。さらに、1 人の人にタスク割り当てが偏らないように平準化することが望まれる。

本タスク処理時間が人別で異なるという制約がない場合、マルチプロセッサスケジューリング問題[1]に帰着でき、ヒューリスティック解法もいくつか提案されている。また、

^{†1} 日本電信電話株式会社 NTT アクセスサービスシステム研究所
〒239-0847 神奈川県横須賀市光の丘 1-1
E-mail: {takasuka.masahide, yoshida.h}@lab.ntt.co.jp.
NTT Access Network Service Systems Laboratories, NTT Corporation
1-1 Hikarinooka, Yokosuka-shi, Kanagawa, 239-0847, Japan
E-mail: {takasuka.masahide, yoshida.h}@lab.ntt.co.jp

マルチプロセッサスケジューリング問題の各プロセスの平準化に関する研究[2]も行われている。これらの問題に対し、各タスク処理時間が人別で異なるという観点を加えたい。

3. 既存研究について

3.1 既存モデル：順序制約付人員資源割当問題

任意の処理時間および任意形状の先行制約を持つ n 個のタスクからなるタスク集合 $J = \{J_1, J_2, \dots, J_n\}$ を、能力の等しい m 人の人員で処理割り込みなしに並列処理するために、各タスクを人員にどのように割り当て、どのような順序で実行するかを決定する問題である。以下ではこのタスクの人への割当をスケジュールと表し、あるスケジュールにおいてタスクの処理を開始してから全タスクの処理が終了するまでにかかる時間をスケジュール長 $Len(S)$ と表す。また、一般に、あるタスク集合 J の m 人の割り当て方は複数考えられるが、それらのスケジュールの中でスケジュール長が最小である、すなわちあるスケジュール S が他の全てのスケジュールの集合 S' に対して

$$Len(S) \leq Len(S')$$

を満たすとき、 S はその順序制約付人員資源割当問題の最適解とする。

このタスク集合 J はタスクグラフ $G(J, E)$ (E は有効エッジの集合)として無サイクル有効グラフで記述され、タスクグラフは1つの入口ノードと1つの出口ノードを持ち、全てのノードは入口・出口ノードから到達できるものとする。図1は8個のタスクからなるタスクグラフの例で、図中の各ノードは1つのタスクを表し、ノード内の数字はタスク、ノード周りの数字は各タスクの処理時間 T_i をそれぞれ表している。なお、本モデルでは、各タスクの処理時間 T_i は既知であることを仮定する。また、図中の各アークはタスク間の先行制約を表し、ノード i からノード j へのアークは、タスク J_i がタスク J_j に先行する半順序制約を表す。

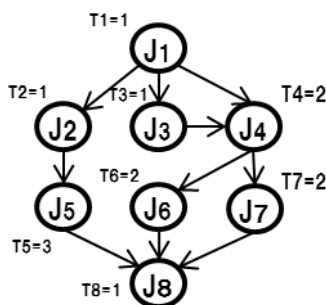


図 1. 順序制約付人員資源割当問題の例

3.2 既存モデルに対する解法

順序制約付人員資源割当問題は強 NP 困難な難しい最適化問題であることが知られている[3]。このため、従来この種の問題を解くアルゴリズムの研究においては、常に最適

解、すなわち最小のスケジュール長を持つ解を得られる保証はないが、多くの問題に対して現実的かつ短時間で比較的良い解を得ることのできる実用的なアルゴリズム[1], [4], [5], [6], [7]や、問題に何らかの制約条件、例えば全タスクの処理時間が全て等しいことやタスク間の先行制約が tree 形状であることなどの制約を設け、問題を狭義にすることで、その特定条件下の問題に対しては最適解が得られるアルゴリズムが提案されるほか、GA(遺伝的アルゴリズム)を用いる方法なども提案されている。

3.3 既存モデルの課題

大規模かつ複雑な EMS 開発において、既存モデルを適用し、各々の人にタスクを割り当てることを行っていたが、計画されていた開発期間や開発コストが実績の開発期間や開発コストとの間に差が生じ、結果、個人々に設定されている時間以上のタスクを個人に負担してしまうような開発が多数存在した。この大きな要因として、各々のタスクの設定されている処理時間が実際のタスクの処理時間と乖離していることが挙げられる。この乖離している要因は、既存モデルでは各々のタスクが定数となっており、このタスクを誰が行っても同じ時間で処理できるという前提条件があるのに対して、現実では人が異なればタスクの処理時間も異なってくるのが挙げられる。そのため、タスクの処理時間は固定値ではなく、人によってタスクの処理時間が異なるよう可変値とし、個人の能力差を考慮したモデル[8]が必要となってくる。

前節にて、順序制約付人員資源割当問題に全タスク処理時間が全て等しいという制約を付け加えた問題を挙げた。本研究で対象となるモデルに各々のタスクの処理時間を固定値とする制約を付けると、既存モデルになるといえる。すなわち、本研究で対象となるモデルは、順序制約付人員資源割当問題の各々のタスクの処理時間を固定値から可変値にパラメータ化したモデルとすることができ、本研究のモデルは既存モデルに比べて広義な問題といえることができる。従って、個人の能力差を考慮した開発における人員資源割当問題は NP 困難な問題であることがいえる。

また、ある特定の人物にタスクが偏りすぎると疲労のような体力的要因や不公平感のような心理的要因から作業効率が落ち、タスクの処理時間が伸びてしまうといった影響が出るのが想定される。そのため、本研究では、タスクの処理時間の分散化も目的に加えるとする。

4. 個人の能力差を考慮した場合のモデル化

4.1 提案モデル

先行制約を持つ n 個の集合 $J = \{J_1, J_2, \dots, J_n\}$ を、能力差のある m 人の人員で並列処理するために、各タスクを人員にどのように割り当て、どのような順序で実行するかを決定

する問題である。以下ではこのタスクの人への割当をスケジュールと表し、あるスケジュールにおいてタスクの処理を開始してから全タスクの処理が終了するまでにかかる時間をスケジュール長 $Len(S)$ と表す。また、一般に、あるタスク集合 J の m 人の割り当て方は複数考えられるがそれら全てのスケジュールの集合を S とする。また、ある人 H_j がタスク J_j にかかるタスク処理時間を T_{ij} とし、そのタスクを処理有無を W_{ij} で表現し、実行する場合、 $W_{ij} = 1$ 、実行しない場合 $W_{ij} = 0$ とする。このとき、下記 3 つの条件を満たす解を本モデルにおける最適解とする。

4.2 評価関数

EMS 開発では、昨今の IT 技術進展の早さから、開発のスケジュール短縮が最も重要であり、かつその中で、どれだけ開発コスト削減できるかが求められる。また、前章でも述べたが、タスクの処理時間の分散化を行う。そのため、本研究で取り扱うモデルにおける評価関数は、スケジュール長、コスト、稼働平準化の順の優先度で下記 3 つの条件を満たすものとする。

評価関数 (①)>②>③の順に優先度が高い)

① $\min_{S \in \mathcal{S}} Len(S)$

: スケジュール長が最も短いスケジュール S とする。

② $\min_{0 < t \leq m} t$

: タスクを割り当てる人数を最小化する。

③ $\min_{S \in \mathcal{S}} \sum_{t=1}^m (\sum_{k=1}^n T_{kt} W_{kt} - (\sum_{j=1}^m \sum_{i=1}^n T_{ij} W_{ij}) / m)^2 / m$

: 決まったスケジュールにおいて各々の人の総タスク処理時間の分散を最小化する。

5. 提案アルゴリズムについて

能力差を考慮した開発における人員資源割当問題は能力差を考慮していない問題、つまりマルチプロセッサスケジューリング問題より広義な問題であるため、NP 困難問題となる [1]。また、既存モデルで提案されている解法は処理時間が固定値であることが前提条件であるため、処理時間が可変値であるモデルでは適用することができない。そのため、全体最適解を得るのは非現実的であるため、ヒューリスティック解法を用い、多項式時間で精度の高い解を得ることのできるアルゴリズムを提案する。

評価関数①～③に対して順番に、下記のようなアルゴリズムで解の探索を行っていく [図 2]。

- 手順 1. 評価関数①に対する木構造型 PERT 法 [図 3]
- 手順 2. 評価関数②に対する k-近傍局所探索
- 手順 3. 評価関数③に対する k-近傍局所探索

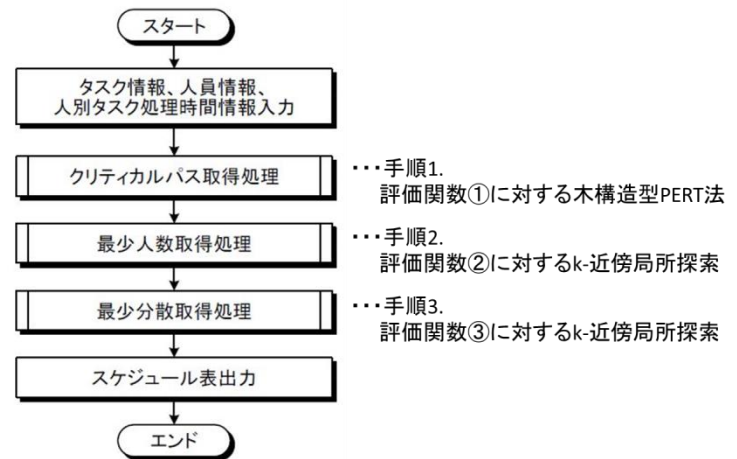


図 2. 提案アルゴリズム 全体像

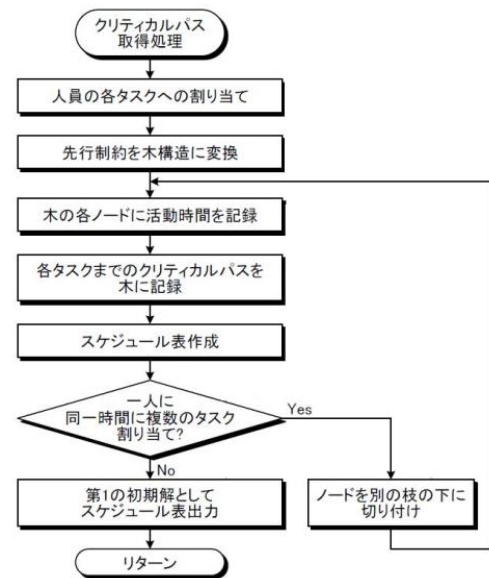


図 3. 木構造型 PERT 法のアルゴリズム

5.1 木構造型 PERT 法

PERT (Program Evaluation and Review Technique) 法とは、複雑なタスクの処理順序の関係をネットワークの形でアローダイアグラムによって表現し、プロジェクトの開始から終了に至るまでの仕事の処理時間に余裕のない経路 (クリティカルパス) を明確にして、予定工期までにプロジェクトを完成できるかどうかの計画の実行可能性を検討し、管理の重点を明らかにする手法である。順序制約付人員資源割当問題に対し、人員リソースが無限大の場合、本手法も用いることで開発期間の最短時間を多項式時間で算出することが可能である。しかしながら、本研究のモデルでは、人員リソースの制約に加え、個人の能力を考慮しているため、人員リソースの個々人全ての人を同等に扱うことができず、個々人にラベルが付いた状態となるため、PERT 法をそのまま利用することができない。そこで、タスクのネ

ットワークダイアグラムを木構造に変換し、各ノードまでの最早時刻と最遅時刻を算出することで、各々の人に同時刻のタスクの割り当てがないかを確認する。同時刻に複数のタスクの割り当てが存在した場合、クリティカルパス上にあるタスクを先に処理し、他のタスクの処理時間を重ならないように動かす。本動作を繰り返すことで、スケジュールが短くなるようなスケジュール S を導出する。本解法を新たな手法として提案し、木構造型 PERT 法[図 4]と定義する。また、本解法の計算量は高々 $O(n^2)$ となる。

Algorithm

- 1: 各ジョブの最短時間となるような担当者を各ジョブに割り当てる。
 - 2: 先行グラフを木構造に変換する。
 - 3: 1と2を元に木の各ノードに活動時間を記録する。
 - 4: 各ジョブまでのクリティカルパスを2の木に記録する。
 - 5: 4を元にスケジュール表を記載し、1人で同一時刻に複数のジョブが割り当てられているか確認する。
 - 5-1: 割り当てられていない場合 そのスケジュールが①でのアウトプット
 - 5-2: 割り当てられている場合
クリティカルパス上のノードをクリティカルパスでない枝の下に切りつける。3へ
-

図 4. 木構造型 PERT 法のアルゴリズム

5.2 k-近傍局所探索

評価関数②の解の導出には、①で得られた解を初期解として、k-近傍局所探索[10]を行うことで、局所的な最適解の探索を行う。さらに、評価関数③の解の導出には、②で得られた解を初期解として、k-近傍局所探索を行うことで、個人の能力差を考慮した人員資源割当問題の解の出力を行う。なお、kの次数は2したとき、計算量は $O(n^2)$ となる。

5.3 提案アルゴリズムの計算量

これら手順1～手順3のアルゴリズムの計算量はいずれも高々 $O(n^2)$ に抑えられるため、手順1から手順3をシリアルに実行した場合も $O(n^2)$ の定数倍となるため、総計算量は $O(n^2)$ となり、多項式時間で解くことが可能である。

6. おわりに

能力差を考慮した開発における人員資源割り当て問題に対する多項式アルゴリズム解法を提案した。本モデルを解くことで EMS 開発などにおける適切な人員資源割り当てを即座に行え、スケジュールの短縮やコスト削減の検討を行うことが可能となる。

しかしながら、現実問題では開発工程が進むにつれて、新たなタスクが増えたり、タスク処理時間に変動があった

り、連続してタスクを処理することによるタスク処理速度の低下等、各パラメータが変動する要素が多々ある。今後はそれらの問題を解くための手法を提案し、各パラメータの変動要素をモデルに組み、より現実に即したモデル化を行っていく。

参考文献

- [1] 飛田 高雄, “マルチプロセッサスケジューリングアルゴリズムの性能評価に関する研究”, phd thesis, 2003.
- [2] ReaKook HWANG, “A Study on Load Balancing Problem Solving by Genetic Algorithm: Case Analyses on Assembly Line and Multiprocessor Systems”, phd thesis, 2009
- [3] M. R. Garey and D. S. Johnson. Computers and Intractability ; A Guide to the Theory of NP-Completeness . 1979.
- [4] Thomas L. Adam, K. M. Chandy, and J. R. Dickson. A comparison of list schedules for parallel processing systems. Communications of the ACM , Vol. 17, No. 12, pp. 685–690, dec 1974.
- [5] E. G. Coffman. Computer and Job-shop Scheduling Theory . John Wiley & Sons, 1976.
- [6] H. Kasahara and S. Narita. Practical multiprocessor scheduling algorithms for efficient parallel processing. IEEE Trans. on Computers , Vol. C-33, No. 11, pp. 1023–1029, nov 1984.
- [7] Behrooz Shirazi, Mingfang Wang, and Girish Pathak. Analysis and evaluation of heuristic methods for static task scheduling. J. Parallel and Distributed Computing , Vol. 10, pp. 222–232, 1990.
- [8] 高須賀 将秀, 吉田寛 “能力差を考慮した開発における人員資源割り当てに関するモデル提案”, 信学会総合大会通信講演論文集, Mar 2016.
- [9] 高須賀 将秀, 吉田寛 “能力差を考慮した開発における人員資源割当問題に対する解法の提案”, 信学会総合大会通信講演論文集, Sep 2016.
- [10] Wikipedia 局所探索法
<https://ja.wikipedia.org/wiki/%E5%B1%80%E6%89%80%E6%8E%A2%E7%B4%A2%E6%B3%95>