日程計画における作業履歴を活用した ファジィ・ランダム多目的最適化の並列分散処理

富山県立大学電子・情報工学科　　指導教員：奥原浩之
1515028　杉山桃香

1　はじめに

1.1　本研究の背景

現在，少子高齢化による労働者人口の減少は 1 つの社会的課題となっている. 国立社会保障・人口問題研究所の調査で，2030 年には，人口の 1/3 近くが 65 歳以上の高齢者になると推計されている. そして，このまま対策がないと GDP の減少は避けられない.[1].

この問題の対策として，限られた資源で最大の利益を得ることが求められる. 対策として，AI の導入や出生率の向上などが挙げられるが，今回はその対策の一つである「最適な人員・費用追加による生産性の向上」に着目した.

また，少子高齢化による労働人口の減少で生産性の向上や作業効率の向上が注目されるなか、建設・土木工事でも、「経済性」「迅速性」「確実性」という、3 つの要素間の適切なバランスをとりあげた工程計画の研究が非常に重要になってきている. 家づくりは長い期間をかけて行うため，進むほどに後戻りが難しくなるため，最初に全体の大まかな流れをつかんでおくことが大切だ.

現状，日本の住宅建設にかかる費用は米国や他国に比べると高いといわれており，その原因の約 50 %は生産性の悪さが問題だとされている. 住宅生産の各作業の中にはフレーミングのように、米国に近い生産性をあげている部分もあるが，全体としては，作業間の連携を含めてシステムが著しく遅れている.

1.2　本研究の目的

労働人口減少による人手不足により「住宅建築・土木工事」でも生産性・作業効率の向上を図っている.　企業さんとのコンタクトを経て現状建築の現場で必要とされていることは各作業の完了に関する情報であることが分かった.

そこで，本研究では現場コミュニケーションアプリなどから得られる作業から次の作業までの. それら所要時間やその時に費やした費用のデータを基に, 時間と費用の面から建設日程計画の最適化を行う.

更に，その所要時間や費用は作業環境などの不確定 (ファジィ性) で不確実 (ランダム性) な要素を含むものである.. よって，本研究の目的は，ファジィ性・ランダム性を考慮した多目的日程計画最適化のモデルを考え，それを GA などの手法を使い並列分散で解くことにより，効果的な人員や費用の追加を補助することができる日程計画を作成することである.

2　多目的日程計画の概要

2.1　所要時間の最小化

本研究における最適な日程計画は「いかに無駄な人員や費用をかけずに作業を無理なく迅速に遂行できるか」を目的としている.

建設の工程計画には順序関係が存在している. 住宅建築における作業をプロジェクトネットワークとして可視化すると図 1 のようになる [2]. これをもとに，この工程における重要な作業 (プロジェクト全体を短縮するのに関わる作業) を導出する.

まず，一番時間がかかる作業の流れ (=クリティカルパス) を求める. クリティカルパスの短縮がプロジェクト全体の短縮に最も効果的であると考え，クリティカルパスの最小化を日程計画作成における一つの目的とした [3].

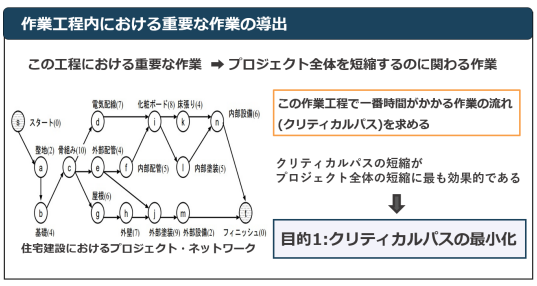

図 1　クリティカルパスの最小化

2.2　時間費用関数の導入による多目的最適化

二つ目の目的として追加費用の最小化を考える. しかし，所要時間を最小化するには作業員の追加 (費用の追加) が伴い，これら 2 つの目的はトレードオフの関係にある.

プロジェクト全体の作業時間の最小化を目的として, クリティカルパスの最小化を考えた. 更に, 二つ目の目的として追加費用も最小化したい. 所要時間を最小化するには, 作業員の追加 (費用の追加) が伴い, これら 2 つの目的はトレードオフの関係にある.

今回作業の所用時間と追加費用の関係は図 2 のような時間費用関数で表せると仮定する [3, 4]. 標準所要時間は追加費用をかけずに作業を達成できる時間とその費用に対応し，特急所要時間追加費用をかけることによって達成できる最短の時間とその費用に対応している

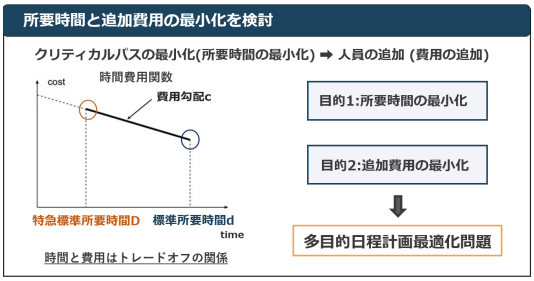

図 2　多目的日程計画最適化

2.3　作業履歴のビックデータを活用

現在，建築の現場では，現場での作業情報を管理・共有する為のアプリが導入されている [5]. そのアプリでは作業員の入退場 (作業時間) などが管理されている. このようなアプリから得られる作業履歴のビックデータからファジィ・ランダム変数や確率のパラメータの設定を行う.

3　ファジィ・ランダム多目的最適化

3.1　ファジィ・ランダム変数化

投入資源による所要時間の変化の不確定性・不確実性を表現するため, 本研究では時間費用関数にファジィ・ランダム変数を用いる. ファジィ・ランダム変数とは，ファジィ性とランダム性の両方を表現することができる変数のことである. ファジィ性とランダム性について描いたものを図 3 に示す.

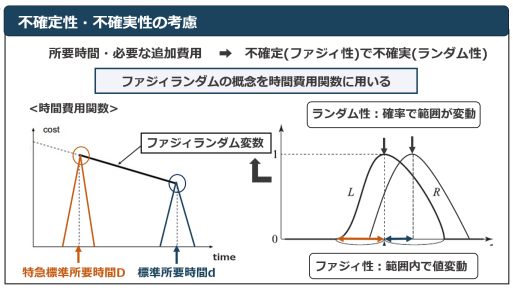

図 3　ファジィ・ランダム変数のメンバシップ関数

ファジィ性とは曖昧性のことである. 値が確定せず, ある範囲で変動するため, 区間値で表すことができる. ランダム性とはその区間が確率で変化することである.

3.2　モデルの定式化

クリティカルパスの最小化と費用の最小化を目的関数としたファジィ・ランダム多目的日程最適化問題を定式化したものを図 4 に示す [4].

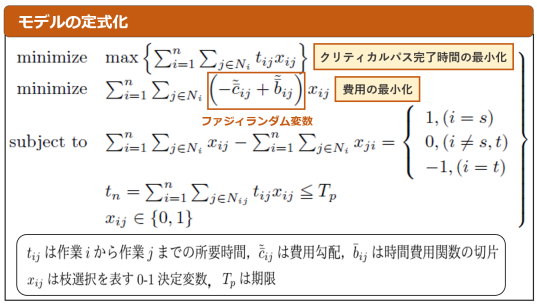

図 4　モデルの定式化

4　等価確定問題への変換

しかし，ファジィ・ランダム変数を含む式をそのまま取り扱うことは困難なため，確率計画問題から多目的計画問題へ等価変換する必要がある. そこで，可能性測度と確率最大化モデルに基づき式を等価な. 問題へ置き換えた [6, 7, 8]. ファジィ性を含む原問題に対して，図 5 のようにファジィ目標を導入する.

$\tilde{c}_{ij}, \tilde{b}_{ij}$ の各要素は図 5 にあるメンバシップ関数により特性づけられるファジィランダム変数を要素とする変数ベクトルである. 目的関数の係数は L-R ファイジィ数において, 中心, 広がりのパラメータが確率変数となっているファジィランダム変数であるため, 各目的関数は, 拡張原理に基づく L-R ファジィ数の演算により, 図 5 のメンバシップ関数で特徴づけられている.

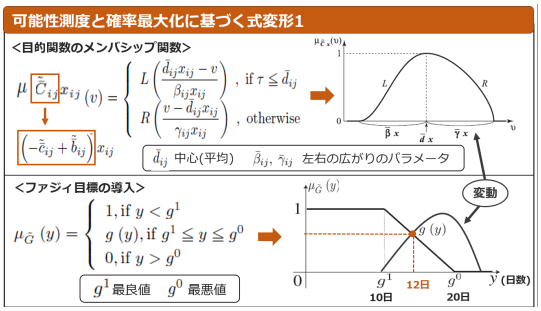

図 5　目的関数のメンバシップ関数とファジィ目標の導入

さらに, 目的関数に対して, ファジィ目標 $\tilde{G}_l$ を導入し, 目的関数のメンバシップ関数 $\mu_{\tilde{c}_{ij}x_{ij}}(y)$ を可能性分布とみなすとき, その分布の下でファジィ目標 $\tilde{G}_l$ が満たされる可能性の度合い $\pi_{\tilde{c}_{ij}x_{ij}}(\tilde{G}_l)$ は、可能性測度を用いて図 6 のように与えられる

ここでは, 問題における $\pi_{\tilde{c}_{ij}x_{ij}}(\tilde{G}_l)$ の最大化,$\pi_{\tilde{c}_{ij}x_{ij}}(\tilde{G}_l)$ の値がある一定値 h_l(満足基準値) 以上となる確率を最大化するという確率最大化モデルに基づき，ファジィ目標が満たされる可能性の度合いを最大化する問題として変形することで，可能性測度の導入によりファジィ性を確定的に取り扱う.

よって，問題は図 6 のような確率計画問題となる. 最後に，確率計画問題を等価な多目的計画問題に変形していくと，問題は非凸非線形 0-1 計画問題となる.

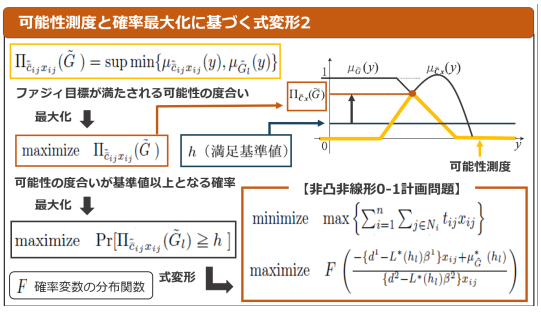

図 6　確率最大化モデルに基づいた変換

6　RaspberryPi を用いた MPI 並列計算

6.1　MPI の概要

本研究では，7 節までで考えたモデルを， 複数の RaspberryPi と MPI を用いて解く.

一般的なコンピュータを使って高速な処理を実現する場合には 1 つの目的を複数のコンピュータで分散・並列処理することになる. この実行基盤の 1 つが MPI （Message Passing Interface）と呼ばれる仕様・規格で，MPICH, Open MPI など複数の実装系が存在している.

Task と呼ばれる独立したアドレス空間で動作するプロセスを CPU コア/スレッド毎に生成し，このプロセス間で MPI 標準規格で定義された関数を使ってデータを分割・移動（Message Passing）しながら並列・協調動作を実現している.

6.2　並列環境の構築

ノード間並列を行う前に，NFS とパスワードなしで，ssh 接続ができる状態にする必要がある.ssh の認証方式としてはよく使われるパスワード認証の他に公開鍵認証というものがあり，公開鍵方式では「公開鍵」と「秘密鍵」の 2 種類の鍵を使って認証を行うことができる.

秘密鍵はログインする側 (ローカル側) の PC に置いておき，公開鍵はログインされる側 (リモート側) の PC に置く必要がある. しかし，秘密鍵と公開鍵は 2 つで 1 組なので，公開鍵認証を使用する際には，どちらかの PC で秘密鍵と公開鍵を一緒に生成し，もう片方の PC に一方の鍵を移すという作業をしなければならない.

今回は，ノード間並列を行う準備として，RaspberryPi(ラズパイ) 8 台分 (master, slave2... slave8) に対して，各ラズパイ同士が，公開鍵認証を用いた ssh 接続ができる環境を整えた.

7　Mpich を用いた並列計算テスト

7.1　Mpich を用いた Hello World

簡単なデモを行う前に，パスワードなしでの SSH 通信を可能にした各 Raspberry Pi に Mpich をインストルールした. また，master の作業ディレクトリを各 slave のディレクトリに共有できるように環境設定を行った.

Mpich を用いるための環境設定を行った後，Raspberry Pi8 台で Hello World を表示させる並列実行のテストを行った. 実行結果を図 7 に示す.

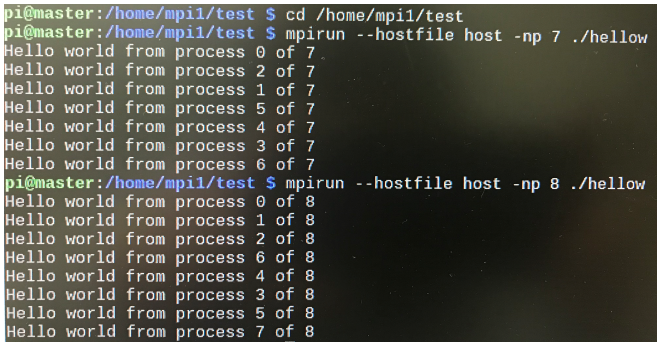

図 7　Raspberry Pi8 台で Hello World を表示

7.2　Mpich を用いた円周率計算

7.1 節で，Raspberry Pi8 台で Hello World を表示させる並列処理の確認ができた. 並列 GA のテストを行う前に，まず，簡単な計算を並列分散処理し，かかった時間を台数ごとに比較してみる. 今回は円周率の計算を 1 台 2 台、4 台、8 台で比較してみた. 実行結果を図 8 に示す. 結果をみると，1 台での処理時間自体が 0.0125(sec) しかかかっておらず，台数を増やしても，増やした分の通信時間の影響が出て，処理時間が短くならなかったと考えられる. 今後，処理に時間がかかる計算をしたときに，高速化されなかった場合は，アルゴリズムを見直す必要がある.

台数	処理時間[sec]
1台	0.0125
2台	0.0142
4台	0.0316
8台	0.0380

図 8　Raspberry Pi 複数台で円周率計算

7.3　島 GA を用いた巡回セールスマン問題 (TSP 問題)

TSP 問題とは，多くの都市と各都市間の移動コストが与えられたとき，全ての都市を 1 度だけ回って戻って来るルートのうち，コストが最

小なものを求める問題のことである. この問題は都市が増えるほど，コストが最小となるものを求めるのが困難になる.

今回は Mpich を使った並列処理テスト実行として，並列遺伝的アルゴリズム (並列 GA) でこの TSP 問題の最適解を求める.MPI への実装とアルゴリズムの流れを下記に示す. 遺伝的操作としては,「部分的交叉」「突然変異」「トーナメント選択」「移住」を行っている.

[1] 各 RaspberryPi にてマップファイルを読み込む

[2] master にて初期集団の生成・配布を行う

[3] 各 slave が初期集団を受け取る

[4] 各 RaspberryPi にて遺伝的操作を行う

各並列数 (1,2,4,8 台) で 3 回ずつ実行させ結果を比較する. 台数ごとの最適解，収束の早さと実行時間をまとめたものを図 9 に示す.

台数	最短距離[km]	処理時間
1台	222.29	2分46秒
2台	215.93	3分03秒
4台	206.43	2分52秒
8台	218.32	2分48秒

図 9 Raspberry Pi 複数台で TSP 問題を解いた結果

図 9 より，2 台以降数を増やしたとき実行時間が短くなっていることが分かる. 1 台の実行時間が短い理由としては，1 台の処理のみ各マシンとの間で移住を行わないため，複数台のときより，実行時間が短くなっていると考えられる. しかし，初期段階で局所解に陥ることを防ぎ，最適解に近づける可能性を高めるためには，移住を行った方が良いため，精度と処理時間の両方を考慮して考えると，台数を増やした方が良い.

8 おわりに

今回は，Mpich をインストールした後の環境設定と，簡単な並列計算のデモを行い速度や精度を比較できた. 今後の課題は，今回の並列 GA を用いた多目的最適化問題を実装すること，とファジィ・ランダムなどのパラメータから等価確定変換した状態の目的関数を設定できるようにすることである.

参考文献

[1] 国内人口推移が、2030 年の「働く」にどのような影響を及ぼすか https://www.recruit-ms.co.jp/research/2030/report/trend1.html

[2] 第 2 章 ネットワーク構造－つながりを見る http://www.econ.tohoku.ac.jp/ ksuzuki/teaching/ch2.pdf

[3] “建設工事における総括工程計画モデルの開発研究”

[4] 飯田耕司, “不確実性への挑戦・意思決定分析の理論”

[5] 現場コミュニケーションアプリ「Kizuku(キズク)」

[6] Hideki Katagiri , Interactive multiobjective fuzzy random linear programming: Maximization of possibility and probability

[7] M. Sakawa, I. Nishizaki, H. Katagiri, Fuzzy Stochastic Multiobjective Programming, Springer, 2011

[8] 椎名孝之, “確率計画法”, 朝倉書店, 2015