実数値遺伝的アルゴリズムによる
最適化問題分散並列解法

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.

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「最適な人員割り当て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. また、現場監督の大きな悩みとして「各大工さんのスケジュールの空き込み具合を把握したい」というものが挙がっている. 更に、人手不足のなかで、どの大工さんをどこの現場 (作業) へ振り分けるかは作業効率に大きく関わってくる要素であると考えられる.

そこで、本研究の目的として、現場コミュニケーションアプリなどから得られる各大工さんの作業情報をもとに、大工さんの作業ごとの熟練度 (単位時間あたりの作業量) を導出. それらの不確定性・不確実性を考慮した、建築工事人員割り当て問題の最適化を検討する. 更に、それを並列分散環境で解くことにより、高速化を実現し、リアルタイム性の高い人員割り当ての最適化を行うことを目的とする.

2 実数値 GA

遺伝的アルゴリズムの遺伝子が実数値である場合を実数値遺伝的アルゴリズムと呼ぶ. 遺伝的アルゴリズムとの違いがいくつかある. 基本的な動作は遺伝的アルゴリズムと同じだが、選択、交叉、突然変異の方法は遺伝子の表現を考慮して変更する必要がある. また、遺伝子が実数の場合、遺伝子の交換ではなく新たな値を生成するような交叉方法が必要になる. 代表的な選択の方法として MGG（MinimalGenerationGap)、交叉の方法としては BLXalpha などがある. このとき、交叉に突然変異要素が含まれているとする考えの元突然変異の操作を行う必要がない.

実数値 GA では、生成した遺伝子 (実数値の組み合わせ) を評価する (=評価関数を計算する) 事に時間がかかるいかに評価関数を使わず最適解を発見できるか、という所がアルゴリズムの評価に入ってくる

3 生存選択モデル

3.1 MGG(MinimalGenerationGap)

実数値 GA のための代表的な世代交代モデルの一つである.
(特徴)

[1] 初期収束や進化的停滞を効果的に回避可能 (多様性の維持に優れている)

[2] 二親交叉を対象に設計されている

MMG には、上記のような特徴があるが「二親交叉」は悪スケールで変数間依存のある問題において有効でない. よって、対策として多親交叉の適用を検討する.

（多親交叉と MGG の生存選択）

[1] 親個体群から 2 個ランダムに選択

[2] 子個体群と合わせ家族を作成

[3] 家族の中の最良個体 1 個と評価値ランキングに基づくルーレット選択で 1 個選択

[4] 家族に加えた親 2 個体と交代し次世代集合をつくる

3.2 JGG(JustGenerationGap)

MGG と同じく実数値 GA のための世代交代モデルの一つであるが、JGG は多親交叉に適した世代交代モデルとして提案されたものである.
（JGG の生存選択）

[1] 子個体群から評価値最良の μ 個体と交叉に参加した親個体群を交代し次世代集団

(MGG と JGG の違い)

[1] MGG は親個体が次世代に残る機会を与えている⇔ JGG では親個体は全て子個体に置き換えられる

[2] MGG は最良個体とルーレット選択により個体を選択⇔ JGG は評価値の良い個体のみ選択

[3] MGG は多親交叉を用でいる時も最大 2 個体交代⇔ JGG では交叉に参加した親個体は全て子個体と置き換えられる

従来研究の実数値 GA のフロンティア [3] より、JGG のほうが数十倍探索効率が良いという結果が得られている.MGG のルーレット選択は収束速度の低下・最終的な解の精度を悪化させる結果をまねくことから、ルーレット選択を行っていない JGG の方が探索効率が良いという結果になった. 他にも、J 交代個体数を増加させたことも探索効率の向上につながったと考えられている.

しかし、次世代に残す個体の選択で JGG では評価値の良い個体を選択する. よって、収束しやすい半面、局所解に陥る可能性がある. このことから、対象とする問題によっては注意する必要がある.

4 数値実験

信頼度や性能を確かめるため代表的ないくつかのベンチマーク問題に対して Simplex+JGG の組み合わせで実行してみた.

4.1 Rosenbrock 関数

Rosenbrock 関数は最適化の標準的なテスト関数であり点 (1,1) で一意に最小値 0 に到達する. この関数は曲線の深い谷に奥行きのない最小値をもつため、最小値の検索がアルゴリズムによっては困難な場合がある. Rosenbrock 関数を図 1 に示す.

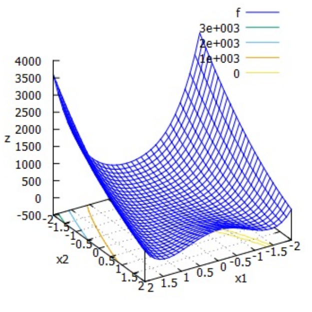

図 1　Rosenbrock 関数

(特徴) 単峰性関数　ベンチマーク関数の代表
(数式)

$$f(x_1 \cdots x_{n-1}) = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \tag{1}$$

(探索範囲と最適解)

$$-5 \leqq \ x_i \leqq \ 5 \qquad (2)$$

$$f_{min}(1,\cdots,1) = 0 \qquad (3)$$

4.2 Rastrigin 関数

Rastrigin 関数は非線形マルチモーダル関数の典型的な例である.Rastrigin 関数を図 2 に示す.

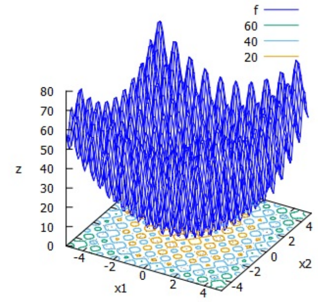

図 2 Rastrigin 関数

(特徴) 多峰性関数 非常に多くの局所解をもつ
(数式)

$$f(x_1 \cdots x_n) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i)) \qquad (4)$$

(探索範囲と最適解)

$$-5.12 \leqq \ x_i \leqq \ 5.12 \qquad (5)$$

$$f_{min}(0,\cdots,0) = 0 \qquad (6)$$

5 実行結果

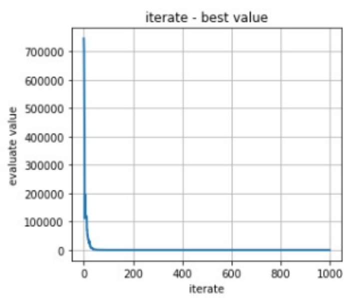

図 3 Rosenbrock 関数の評価値の推移

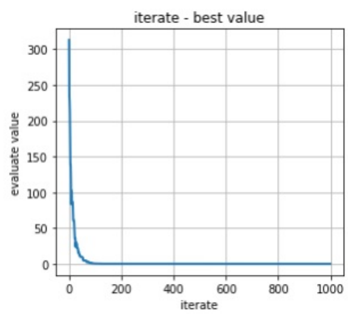

図 4 Rastrigin 関数の評価値の推移

出力として、それぞれの関数を Simplex+JGG の組み合わせで探索した場合の評価値の推移と遺伝子の推移の表示し、更に、処理時間の計算を行った. デモ 2 に関して、世代と遺伝子の分布の変化から、世代が進むにつれ遺伝子の値が 0(最適解) に近づいていく動きを見ることができた. よって、ｓ implex+JGG の組み合わせは多峰性関数の非常に多くの局所解をもつ問題に対して有効であることが分かった.

5.1 並列分散処理

今回デモ 1 +デモ 2 の処理時間を 1 台で処理した場合と 3 台で処理した場合での比較を行った. それぞれの処理時間を図 5 に示す.

1台	3台
83.3070330619812秒	86.17149519920349秒
83.46961188316345秒	83.38053011894226秒
83.16080045700073秒	85.08097243309021秒

図 5 処理時間

(考察)
・1 台処理より 3 台処理の方が時間がかかっている
・分散処理を行った方は処理時間が安定していない

3 台での処理の方が時間がかかる原因として考えられるのは、このプログラム自体が分散処理することをあまり考慮していないことであると考えられる.

6 まとめ

今回は、本研究で対象とする問題が実数値を扱う問題になると考え実数値 GA という手法について調査を行った. 更に、実数値 GA の JGG と Simplex 交叉を組み合わせた手法を用いて、2 つのベンチ—マーク関数の解探索を行った. 結果、ｓ implex+JGG の組み合わせは多峰性関数の非常に多くの局所解をもつ問題に対して有効であることが分かった. しかし、今回デモ数値実験に用いたプログラムは探索効率に関してはあまり考慮されておらず、並列分散を行ってもの処理時間を短縮することが出来なかった.

6.1 今後の課題

今後の課題は、NSGA-や今回調べた実数値 GA の遺伝子操作を参考に、既存のファジイランダムを考慮した多目的最適化問題を解くシステムの開発をおこなうことである. 更に、それらの問題に対して、ラズベリーパイ上での分散処理を行う.

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] 最適化アルゴリズムを評価するベンチマーク関数まとめ https://qiita.com/tomitomi3/items/d4318bf7afbc1c835dda

[3] 実数値 GA のフロンティア

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