

進化アルゴリズムによる MXF-MCF 問題の一解法

田口研究室

1 はじめに

今日の社会には、ネットワーク構造をもつシステムが数多く存在する。

しかし、ネットワーク上でのモノを配送するときのフローを最大にする最大フロー問題 (Maximum Flow problem) と、費用を最小にする最小コストフロー問題 (Minimum Cost Flow Problem) は数多くの制約条件を伴う多目的ネットワークモデルとして定式化され、NP 困難な組み合わせ最適化問題となる。これに対して進化アルゴリズムである遺伝的アルゴリズム (Genetic Algorithm) を適用することによって、組み合わせ最適化問題の一つである最大フロー/最小コストフロー問題 (以下:MXF-MCF 問題) に対して、有効な解の提示を可能とした。

過去の手法 [1] では探索可能な全てのフローを探索しているが、本研究ではフローを制限するデコーディング方法を用いることで、非常に制度の高い (コストが低い) 非劣解を求めることができる。また、ソリューションセットより優れた個体の割合の増加と各世代の非劣解の数の増加を図る。さらに、各フローに対するコストをさらに下げることが目指す。

2 MXF-MCF 問題

MXF-MCF 問題とは、MXF と MCF の二つの目的関数であるフローの最大化とコストの最小化を同時に満足させる問題である。 m 個のノード $i, j \in V$ と n 個のアーキ $(i, j \in A)$ を持つグラフ $G = (V, A)$ と G の各アーキに関してコストと容量が与えられているとき、ノード番号を $i, j = 1, 2, \dots, m$ として、 c_{ij} をアーキ (i, j) のコスト、 u_{ij} をアーキ (i, j) の容量の上限値、 f をフロー量、 x_{ij} をノード i からノード j までのフロー量、とすると一般に、最大フロー/最小コストネットワークモデルは次のように定式化される..

$$\max z_1 = f \quad (1)$$

$$\min z_2 = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \quad (2)$$

$$\text{s.t.} \quad \sum_{j \in \text{succ}(i)} x_{ij} - \sum_{j \in \text{pred}(i)} x_{ji} = \begin{cases} f & (i = 1) \\ 0 & (i = 2, 3, \dots, m-1) \\ -f & (i = m) \end{cases} \quad (3)$$

$$0 \leq x_{ij} \leq u_{ij}, \forall (i, j) \in A, 0 \leq f \leq 90 \quad (4)$$

ここで、 $\text{succ}(i)$ と $\text{pred}(i)$ はそれぞれノード i の後継ノードセットと前継ノードセットである。

3 優先順位ベース GA によるデコーディング法

最大フロー/最小コストネットワーク問題では、実行可能解は複数のパスからなる場合が多い。つまり、一つの染色体によって複数のパスを表現する必要がある。また、単純に複数のパスを組み合わせた染色体表現の場合、プ

ログラム作成が難解になってしまう。

3.1 エンコーディング法

[ルーチン 1:優先順位ベースの染色体生成 (エンコーディング処理)]

入力データ:ノードの数 m , 集団サイズ $popsize$

出力データ:染色体 v_k

ステップ 0: $i = 1$ から m まで $v_k(i) \leftarrow i$ を繰返す (1 から m までの数を順番に染色体に入れる)。

ステップ 1: $i = 1$ から $[m/2]$ までステップ 2~3 を繰返す。

ステップ 2: $j \leftarrow \text{random}[1, m], l \leftarrow \text{random}[1, m]$ 。

ステップ 3: $l \neq j$ であれば $v_k(j)$ の要素と $v_k(l)$ の要素を交換する $i \leftarrow i + 1$ 。

ステップ 4:ステップ 0 から 3 までを $k = popsize$ となるまで繰返す。

ステップ 5: 優先順位ベースの染色体 v_k を出力する。

3.2 デコーディング (提案手法)

ここで、複数パスを生成するデコーディング法を説明する。このデコーディング法は優先順位ベースエンコーディング法により生成された染色体から一つのパス P_k を生成し、ネットワークのフロー量を更新する。このとき、各染色体のフローに制限をつける f_{max} を与えることで、本来コストとフローの二目的線形問題を、一目的として扱うことができる。

[ルーチン 2:一つのパス生成]

入力データ:ノードの数 m , 染色体 v_k , ノード i の集合 S_i

出力データ:経路の集合 P_k

ステップ 0: 開始ノード $i \leftarrow 1, P_k \leftarrow \phi$ 。

ステップ 1: ノード i の集合 $S_i \neq \phi$ であれば、ステップ 2 を繰返し、そうでないならステップ 3 に進む。

ステップ 2: 高い優先順位を持つノードの選択。ノード i の集合 $S_i = \{j | (i, j) \in A\}$ の経路を探索し、生成した優先順位ベース染色体によって、高い優先順位を持ち $u_{ij} > 0$ のノード l を選び、 $P_k \leftarrow P_k \cup \{x_{ij}\}$ を設定する。もし選ばない場合は終了する。選ばれたら、 $i \leftarrow l$ とする。

ステップ 3: 経路 P_k を出力し、終端ノード $g \leftarrow l$

[ルーチン 3:すべて実行可能なパス生成]

ここで、経路のコストを c_l^k , フロー量を $f_l^k \forall l \in L_k$ とする。

入力データ:染色体 v_k , ネットワークデータ (A) , ノード i の集合 S_i , 終端ノード a , 集団サイズ $popsize$, フロー制限 f_{max}

出力データ:パスの数 L_k , 各染色体のコスト $c(v_k)$ とフロー量 $f(v_k)$

ステップ 0: $l \leftarrow 0$, 初期経路 $x_{ij} \leftarrow 0, \forall (i, j) \in A, \tilde{u}_{ij} \leftarrow u_{ij}$

ステップ 1: 集合 $S_1 \neq \phi$ もしくは、 $f_1^k < f_{max}$ ならば $l \leftarrow l + 1$ しステップ 6 まで繰返す。そうでないとき、または終端ノード a に到達できないならば、ステップ 7 へ進む

ステップ 2: 染色体 v_k と集合 S_i を用いて、一つのパス P_l^k をルーチン 2 によって生成し、パス P_l^k の終了ノード g を求める。

ステップ 3: パス P_l^k の終了ノード $g = a$ であれば、ステップ 4 へ進む、そうでないなら、 x_{ia} によって、 $S_i \leftarrow S_i - \{a\} \forall i$ を更新し、ステップ 1 に戻る。

ステップ 4: 生成したパス P_l^k のコスト c_k とフロー量 f_k を計算する.

$$f_l^k \leftarrow f_{l-1}^k + \min\{\tilde{u}_{ij}|(i, j) \in P_l^k\} \quad (5)$$

$$c_l^k \leftarrow c_{l-1}^k + \sum_{i=1}^m \sum_{j=1}^m c_{ij}(f_l^k - f_{l-1}^k) \quad (6)$$

ステップ 5: 次式より, 各 P_l^k のアークの容量 \tilde{u}_{ij} と x_{ij} を更新する.

$$\tilde{u}_{ij} \leftarrow \tilde{u}_{ij} - (f_l^k - f_{l-1}^k)\forall(i, j) \in P_l^k$$

$$x_{ij} \leftarrow x_{ij} + (f_l^k - f_{l-1}^k)\forall(i, j) \in P_l^k$$

ステップ 6: 各アークにおいて次の条件を満たしているかチェックを行う.

$$\tilde{u}_{ij} = 0, \forall(i, j) \in P_l^k$$

条件を満たすノード j がある場合, ノード j を通るパスは生成できない. そこで以下のように S_i を更新する.

$$S_i \leftarrow S_i - \{j\}, \forall(i, j) \in P_l^k \ \& \ \tilde{u}_{ij} = 0$$

ステップ 7: $c(v_k) \leftarrow c_l^k, f(v_k) \leftarrow f_l^k$
 ステップ 8: $k \leftarrow k + 1$ とし $k = \text{popsize}$ となるまでステップ 0, ステップ 1 を繰り返す.
 ステップ 9: パスの数 $L_k \leftarrow l - 1$, パス $P_k \in L_k$ のコスト $c(v_k)$ とフロー量 $f(v_k)$ を出力する.

3.3 遺伝的操作

交叉: ここでは重みマッピング交叉法を使用する.
 突然変異: ここでは, エンコーディング処理と全く同じ方法で生成した染色体を挿入する.
 選択: ここでは高速優越ソートを用いてランク 1 の染色体だけは次世代に残し, それ以外のものをランダムで選択する.

3.4 パレート最適解集合の更新

遺伝的アルゴリズムを多目的最適化問題に適用する場合は, 通常の解集合とは別のパレート最適解集合を持ち, 各世代の解集合によってパレート最適解集合を更新することになる.

4 評価

[参照ソリューションセット S^*]

文献 [1] より得られている参照ソリューションセットを用いる. 比較を行うすべてのアルゴリズムより得られた非劣解集合のことを参照ソリューションセット S^* と呼び, この集合 S^* と各手法で得られたパレート最適解の集合を S_j とし, それらと比較することで, 各アルゴリズムで得られた非劣解集合が全体に対してどの程度の精度やばらつきがあるか, 非劣解集合であるかを評価することができる.

[得られた非劣解の数 $|S_j|$]

$|S_j|$ は各手法により得られた非劣解集合の数であり, これを評価値とする.

[優越個体割合 $R_{NDS} S_j$]

次式より算出される優越個体割合 $R_{NDS} S_j$.

$$R_{NDS}(S_j) = \frac{|S_j - \{x \in S_j | \exists y \in S^*, y < x\}|}{|S^*|} \quad (7)$$

[参照ソリューションセットとの距離 $D1_R$]

誤差 $D1_R$ は, 参照ソリューションセットとの距離であり, 次式により算出される.

$$D1_R = \frac{1}{|S^*|} \sum_{y \in S^*} \min\{d_{xy} | x \in S_j\} \quad (8)$$

ここで, d_{xy} は求められた解集合 S_j にある求められた解 x から集合 S^* にあるパレート最適解 y までの距離である.

5 数値実験

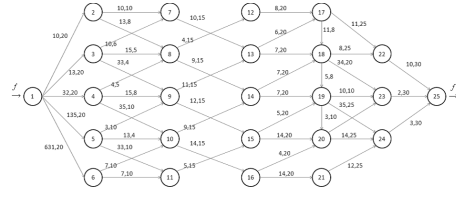


図 1 数値実験を行う問題

25 個のノードと 49 個のアークを持つネットワークの数値例 (Munakata, T., & D. J. Hashier) [2] に基づいて, 数値実験を行う (図 1).

今回の選択方法で用いた方法では, 優先度や適応度を用いていないため, 遺伝的ではなく進化アルゴリズムとしている. そのためルーレット選択やトップシス法を用いた遺伝的な選択方法も同じく数値実験を行う. それによって, 今回提案した手法と比較してみる.

また, 今回の実験で使用した, 参照ソリューションセットとの距離 $D1_R$ を求めて参照ソリューションセットとの距離が短ければ良い解としてきたが, 非劣解が参照ソリューションセットよりも優越な解であった場合, $D1_R$ の値が大きくなってしまふ. そこで, 今回の問題は第一目的の最大化と第二目的の最小化が目的関数のため, トップシス法により評価を行うことで, 参照ソリューションセットとの距離 $D1_R$ を求めるよりも正確に評価することができる. 紙面の都合上実験結果は発表当日に示す. [トップシス法]

この手法は各世代生成された非劣解より得られる理想点を利用し, その理想点に非劣解がどれだけ離れているかを計算する.

非劣解から得られる 4 個の端点 (maximum: $Z_1^{(*)}$, minimum: $Z_1^{(-)}$, maximum: $Z_2^{(-)}$, minimum: $Z_2^{(*)}$) を次のように定義する. ただし, q は生成した非劣解の数とする.

$$Z_1^{(*)} \leftarrow \max : z_1^1, z_2^1, z_3^1, \dots, z_q^1 \quad (9)$$

$$Z_1^{(-)} \leftarrow \min : z_1^1, z_2^1, z_3^1, \dots, z_q^1 \quad (10)$$

$$Z_2^{(-)} \leftarrow \max : z_1^2, z_2^2, z_3^2, \dots, z_q^2 \quad (11)$$

$$Z_2^{(*)} \leftarrow \min : z_1^2, z_2^2, z_3^2, \dots, z_q^2 \quad (12)$$

6 おわりに

本研究では, MXF-MCF 問題において GA を用いる解法を提案した. また, GA における評価方法を提案した. 数値実験においては次のような結果が得られた.

- 優越個体割合, 非劣解の数は染色体数 5000 において, 提案手法が全ての手法に対して上回った.
- 参照ソリューションセットへの距離 $D1_R$ は上回らなかったものもあるが, 優越個体割合が大きいことから, 優越な解が多数存在していることが明らかになった.
- 解の選択方法は優越ソートで求めた値が高い数値を得られた.

ソリューションセットに対して, 全ての個体集団が優越であるとはいえなかったが, V25/A49 の平均解とソリューションセットとの差が小さかったことから, 今回提案した手法の有効性を確認した.

参考文献

- [1] 玄光男・林林『ネットワークモデルと多目的 GA』, 共立出版, 2008.
- [2] Munakata, T. and D. J. Hashier: "A genetic algorithm applied to the maximum flow problem", *Proc. of 5th Int. Conf. Genetic Algorithms*, pp.488-493, 1993.