

ハイパーグラフ分割問題に対する分散遺伝的アルゴリズム

A Distributed GA Hybrid for Hypergraph Multi-Way Sectioning

上土井 陽子 若林 真一

Yoko Kamidoi Shin'ichi Wakabayashi

広島大学工学部

Faculty of Engineering, Hiroshima University

〒724 東広島市鏡山一丁目 4 番 1 号

4-1, Kagamiyama 1 chome, Higashi-Hiroshima 724 JAPAN

1. まえがき

ハイパーグラフの分割問題はサイズの制約条件を満たすカット数最小のハイパーグラフの分割を求める問題である。この問題は一般に NP 困難^[4]なことが知られており、現在までにいくつかのヒューリスティックアルゴリズムが提案されている。代表的な手法について以下に述べる。Kernighan と Lin は節点サイズが均一なグラフを 2 等分割する反復改良法に基づくヒューリスティックアルゴリズム（以下、KL 法と呼ぶ）^[12]を提案した。KL 法を基礎として、以後、多くのハイパーグラフ分割問題に対するヒューリスティックアルゴリズムが提案されている。KL 法では節点の対交換により解を改良し、改良されなくなったとき終了する。ハイパーグラフの節点数を m とするとき、KL 法の 1 回の適用の時間計算量は $O(m^2 \log m)$ である。KL 法を基礎として、以後、多くのハイパーグラフ分割問題に対するヒューリスティックアルゴリズムが提案されている。Fiduccia と Mattheyses は KL 法にセルゲインの概念を導入し、ハイパーグラフ 2 分割アルゴリズム（以下、FM 法と呼ぶ）^[3]に拡張した。セルゲインは節点を移動することによって減少する目的関数の値を表している。この手法の 1 回の適用あたりの時間計算量は入力ハイパーグラフの節点次数の総和に関し線形である。FM 法の拡張として Krishnamurthy は節点移動によって生じるゲインの変化を先読みするためのレベルゲインと呼ぶ新しい概念を導入したハイパーグラフ 2 分割手法^[13]を提案した。また、Sanchis は Krishnamurthy の手法を k (≥ 2) 分割手法^[15]に拡張した。この手法においても 1 回の適用の時間計算量は入力ハイパーグラフの節点次数の総和に関して線形である。Kahn は節点に重みを持たないハイパーグラフ 2 等分割問題に対する組立て法による手法^[8]を提案している。この方法は intersection graph と呼ばれる入力ハイパーグラフの双対グラフに基づいており、ハイパーグラフの節点の重みは考慮されていない。以上の手法はいずれも目的関数が減少する場合にのみ解の改良を行っているために、局所最適解に陥り易いという問題点がある。

局所最適解に陥ることを防ぐ一般的な方法の 1 つとして遺伝的アルゴリズムがある。この方法は遺伝子の進化のプロセスの概念を組合せ最適化問題に応用したものである^[5]。遺伝的アルゴリズムは主に解の交配、淘汰、突然変異のプロセスから構成されており、一般に解を文字列としてコード化し、コード上での解の交配により解の改良を試みる。コード化を行うことにより問題に依存しない汎用的な解法を構築でき

る。著者らはハイパーグラフ分割問題に対し、遺伝的アルゴリズムと高速ヒューリスティックアルゴリズムを組み合わせた手法^{[10][11]}を提案した。この手法では解のコード化を行わずに解の初期化、交配においてヒューリスティックアルゴリズムを用いており、大規模な入力に対しても実用的な計算時間で一般のヒューリスティックアルゴリズムでは求めにくい良質な解を求めることが示されている。本稿では、この遺伝的アルゴリズムに基づいたヒューリスティックアルゴリズムの分散アルゴリズム化について考察する。

一般に、遺伝的アルゴリズムを分散システム環境上で実現する場合、交配手続きを分散処理することで計算時間を短縮できると予測される。さらに、分散実行において解集合を積極的に多様化することで局所最適解に陥ることを防ごうとする試みがある。本稿では、著者らが既に提案している遺伝的ハイパーグラフk分割手法のクライアント・サーバモデル上での分散実行について議論する。まず、大規模な入力データに対する逐次アルゴリズムの解の性質を実験的に考察する。次に、考察に基づき、局所最適解への収束回避を目的とした分散アルゴリズム化を提案する。また、ワークステーションより構成される分散システム上に提案手法を実現し、逐次アルゴリズムとの比較による実験的評価を行う。

2. ハイパーグラフ分割問題

ハイパーグラフとカットに関する用語の定義、および、本稿で考察するハイパーグラフ分割問題の定式化を以下に示す。

[定義1] ハイパーグラフ $H=(V, E)$ は節点集合 $V=\{v_1, v_2, \dots, v_m\}$ とハイパー枝集合 $E=\{e_1, e_2, \dots, e_n\}$ よりなる。ここで各ハイパー枝 $e \in E$ は V の空でない部分集合である。もし、全ての $e \in E$ に対し、 $|e|=2$ なら H はグラフでもある。節点集合 V にその重みを表す関数が定義されていたならばハイパーグラフ $H=(V, E)$ は重み付きハイパーグラフであると言う。各節点 v の重みを v のサイズと呼び、 $size(v)$ で表す。節点の重みは全て正整数であると仮定する。□

[定義2] 重み付きハイパーグラフ $H=(V, E)$ に対し、自然数 k と関数 $P: V \rightarrow \{1, 2, \dots, k\}$ が与えられたとする。このとき、 $V_i=\{v \mid v \in V, P(v)=i\}$ ($1 \leq i \leq k$)で表される k 個の互いに素な節点部分集合を H の k 分割と定義する。そして、各ハイパー枝 $e \in E$ に対し、スパン関数 $q(e)=\{P(v) \mid v \in e\}$ を定義し、 $\{e \mid |q(e)| \geq 2, e \in E\}$ なるハイパー枝の集合を k 分割カットセットと言う。 k 分割カットセットのサイズを k 分割カットセットに属しているハイパー枝の数で表す。ハイパーグラフ H の最小コスト k 分割は $|\sum_{u \in V_i} size(u) - \sum_{v \in V_j} size(v)| \leq \max\{\max\{size(v) \mid v \in V\}, \beta \times W\}$, $1 \leq i, j \leq k$ を満たす最小コストの k 分割である。ここで、 W は節点の重みの総和を表し、 β は0.1以下の正の定数とする。□

3. 遺伝的ハイパーグラフk分割アルゴリズム

ここでは著者らが既に提案しているハイパーグラフk分割問題を解く遺伝的アルゴリズムに基づく手法^{[10][11]}について概説する。

3.1 遺伝的アルゴリズムの概要

一般に遺伝的アルゴリズムは遺伝子の進化の概念を組合せ最適化問題に応用した確率的アルゴリズムとして知られている^[5]。遺伝的アルゴリズムの特徴はコード化された複数の解を常に保持し、前世代の2個以上の解のコードを交配することで新しい解を生成することである。これは遺伝子の交配プロセスの概念に基づいている。様々な最適化問題を扱うため、遺伝的アルゴリズムは解のコード化を行い、各個別問題固有の情報とは独立に2個以上の解をランダムに交配する。しかし、解をコード化することによりサイズ

の大きな問題に対しては解の収束に多大な計算時間が必要であるなどの問題点があった。JonesとBeltramoは遺伝的アルゴリズムとgreedyな手法を組み合わせることにより汎用性は失うが一般的な遺伝的アルゴリズムより高速に良質な解を出力することを示した^[7]。

著者らは既にハイパーグラフk分割問題に対する遺伝的アルゴリズムに基づくヒューリスティックアルゴリズムを提案している^[10]。この手法は主に初期化、交配、淘汰、反復の4ステップにより構成されている。初期化ステップではp個の異なる解を著者らが提案したハイパーグラフ2分割手法^[9]の拡張であるヒューリスティックアルゴリズムHKS^[11]を用いて求める。交配プロセスでは前世代の2個の解を組み合わせ、2個の解の共通性質を受け継いだ新しい解を初期化ステップと同様な手法を用いて求める。淘汰のステップでは交配プロセスで求めた $p(p-1)/2$ 個の解から目的関数の良い解をp-m個、残りの解よりm個ランダムに選び次世代を構成する。ここではm個の良質でない解を含めることにより一般の遺伝的アルゴリズムにおける突然変異と同様に局所最適解に陥ることを回避することを期待している。

ハイパーグラフk分割問題に対する遺伝的アルゴリズムGHKS (Genetic Algorithm Hybrid for Hypergraph k-way Section)を以下に示す。

[アルゴリズムGHKS]

Step1: パラメータp, R, mを入力する。 $\gamma = 0$ 。

Step2: アルゴリズムHKSを適用し、p個の初期解を求め、解集合Pを作成する。

Step3: 解集合Pの全ての非順序対に対しアルゴリズムHKSを適用し交配する。

Step4: Step3で求められた $p(p-1)/2$ の解よりp-m個の目的関数の良い解と残った解からランダムにm個選ぶ。選ばれたp個の解により集合Pを更新し、次世代を構成する。

Step5: 終了条件を満たしていれば終了。そうでなければ、Step3へ。

紙面の都合上、解の交配手法等のアルゴリズムの詳細は文献[11]に譲り、ここでは省略する。

4. 分散アルゴリズム化

4. 1 目的

遺伝的アルゴリズムの分散システム上での実現において、最も基本的と思われる手法は交配手続きを分散アルゴリズム化することである。著者らは文献[11]でクライアント・サーバモデル上での遺伝的アルゴリズムにおける交配手続きの分散実行を試みた。得られたシミュレーション実験結果より入力データの規模が大きい場合、通信によるオーバーヘッドも無視できなくなり何らかの通信データ量の削減が必要になることがわかった。

一般に、遺伝的アルゴリズムの分散アルゴリズム化では、各プロセスが独立に遺伝的アルゴリズムを実行した場合、各プロセスがそれぞれ独自の解集合を保つため、独立した環境を構成していると見なすことができる。このとき、プロセス間で解の情報を通信することで各環境を変化させ、局所最適解に陥ることを防ごうとする手法が提案されており、その有効性が示されている。

遺伝的アルゴリズムの分散アルゴリズム化の一手法として、環境変化の概念の導入^{[2][6]}が挙げられる。これらの手法では、分散実行時に問題となった通信によるオーバーヘッドを小さくし、かつ、効率良く良質解を得るための解の通信方法が導入されている。これらの手法は、良質な解の情報、または解集合中の解とは異なる性質を持つ解を解集合中に意図的に混入することが局所最適解に陥ることを防ぐために有効であるという予測に基づいている。

しかし、本研究で考察の対象としているヒューリスティックアルゴリズムと遺伝的アルゴリズムを組み

合わせたHybrid手法では、人口数（アルゴリズム実行時に常に保持している解の数）が小さいため、通信された解が解集合に及ぼす影響が大きく、プロセス間で異なった環境を保持することが困難である。よって、提案Hybrid手法に応じた分散アルゴリズム化を考察する必要がある。

本稿では、始めにHybrid手法の性質に関する実験的考察を行う。次に、考察に基づいた分散アルゴリズム化を提案する。

4. 2 アルゴリズム逐次実行時の性質

遺伝的アルゴリズムに基づく提案ハイパーグラフk分割手法（アルゴリズムGHKS）の性質を実験的に考察する。シミュレーション実験より、アルゴリズムGHKSは解の質が各世代を構成する解の数（人口数）に依存することが分かった。この状況は入力ハイパーグラフの規模により変化する。小規模のデータに対しては、人口数の解の質に対する影響は比較的小さいが、データの規模が大きい程、人口数の大きなアルゴリズムの適用で、良質な解が得られている。例えば、人口数5と人口数15の場合を比較すると、20回の試行の最小のカット数で20%以上、平均カット数で35%以上人口数15の試行の方が良質な解を得ている。しかし、小規模なデータに対してはあまり差がないことから、人口数が多ければ多い程いいという仮説は成り立たず、データの規模に応じて人口数の適切な値が有ると思われる。すなわち、データの規模が大きい程、十分良質な解を得るのに必要な人口数は多くなると予測される。

なぜ、小さい人口数でのアルゴリズムの試行が既知である最小値と比較して悪い解に収束するかを考察するため、MCNC（Microelectronics Center of North Carolina）の大規模なベンチマークデータであるInd2（節点数12142、ハイパー枝数13419）に対し、各世代での解の類似度の平均値、および、最小カット数を比較する。ここで、類似度とは、2つの分割を重ね合わせたときの最大共有節点サイズと節点サイズの総和の比で表す。例えば、2分割の場合、図1に示すように2つの分割を重ね合わせたとき最大4個の

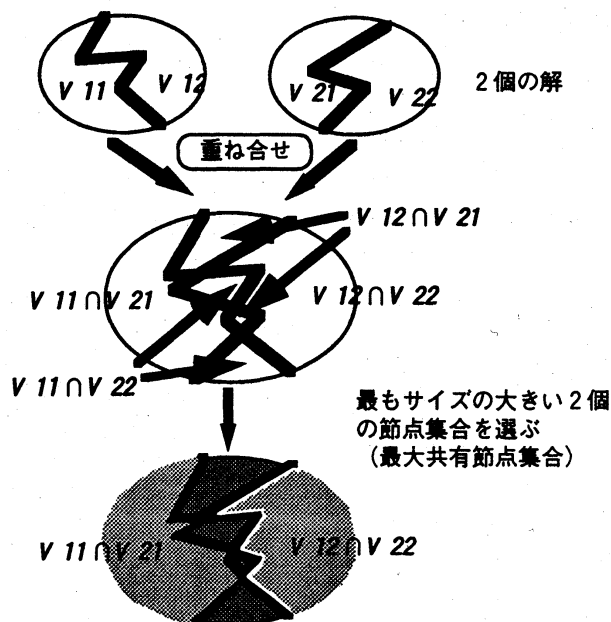


図1. 2つの分割の類似度（2分割の場合）

部分ハイパーグラフに分割できるが、サイズの大きな2個の部分ハイパーグラフが $V_{1R} \cap V_{2R}$, $V_{1L} \cap V_{2L}$ に誘導されるハイパーグラフだとする。このとき、2つのカットの類似度は以下の値で表される。

$$\frac{\text{size}(V_{1R} \cap V_{2R}) + \text{size}(V_{1L} \cap V_{2L})}{\text{size}(V)} \times 100 [\%]$$

20個のランダムな初期状態に対し、 $p=5$, $R=10$, $m=2$ と $p=15$, $R=10$, $m=2$ でアルゴリズムを実行した。各世代ごとに20個のカット数の平均、最小、類似度の平均に関し比較した結果を表1に示す。ここで、改良度は1つ前の世代のカット数に対する現世代のカット数の改良された割合を示す。実験結果より、小さい人口数のアルゴリズムの試行では、第2, 3世代より後の世代ではほとんど解が改良されておらず、一方、人口数が大きいと第5, 6世代まで解の改良が続いていることが分かる。また、類似度に関しても人口数が大きい場合、いくぶんゆっくりと減少していることがわかる。これより、世代を構成している人口の多様性が良質な解を導くための必要条件の1つであるといえる。

人口の多様性を増加させるため、環境変化の概念を導入することが有効であると予測される。そこで、人口数5で分散環境上で4個のアルゴリズムを並列に実行し、解の通信を行う実験を試みたが、人口数15で得られた解と同程度の質の解は得られなかった。これは、分散実行しても、1つのアルゴリズムで改良された解は類似している、もしくは、同程度の解の質であるため解も類似していることより、それらの解を少数組み合わせても改良は見込めない等が原因として挙げられる。

また、多様性の他に、人口数が大きい程、淘汰は厳しいものとなることがいえる。交配手続きにより構成される解の数は人口数の二乗オーダーであるので人口数が多いほど次世代に残れる解の数の構成される解の総数に対する比率は小さくなり、淘汰の基準が厳しくなり良質な解しか次世代に残れないことになる。

表1. アルゴリズムGHKS (2分割, Ind2の場合) の $p=5$ と $p=15$ の試行の比較

(a) $p=5$ の試行

世 代	1	2	3	4	5
類似度 [%]	71	80	90	94	97
カット数 (最小)	456	415	404	373	352
カット数 (平均)	691	587	532	514	501
改良度 [%]	-	15	9	3	3

(b) $p=15$ の試行

世 代	1	2	3	4	5
類似度 [%]	70	80	88	93	95
カット数 (最小)	334	331	309	281	276
カット数 (平均)	567	445	389	369	356
改良度 [%]	-	22	13	6	4

4. 3 分散遺伝的アルゴリズム

4. 2でのアルゴリズムGHKSの逐次実行に対する実験的考察より、厳しい選択基準で解を選び、良質な解集合を構成すること、および、解集合の多様性を保つことの2つが良質な解を得るための重要な要素ではないかと考えた。以上の議論に基づき、クライアント・サーバモデル上での分散遺伝的アルゴリズムを提案する。サーバ・プロセス、クライアント・プロセスの手続きを以下に示す。

[アルゴリズム DGHKS]

[クライアント・プロセスの手続き]

Step1: パラメータ値 p , R , m を入力する. $\gamma = 0$.

Step2: アルゴリズム HKS を適用し, p 個の初期解を求め, 解集合 P を作成する.

Step3: 解集合 P の全ての非順序対に対しアルゴリズム HKS を適用し交配する.

Step4: Step3で求められた $p(p-1)/2$ の解より $p-m$ 個までの目的関数の良い解と残った解からランダムに m 個選ぶ. 選ばれた p 個の解により集合 P を更新し, 次世代を構成する.

Step5: 終了条件を満たしていればサーバ・プロセスに得られた最良解を送って終了. そうでなければ, Step3へ.

[サーバ・プロセスの手続き]

Step1: パラメータ値 p , R , m を入力する. $\gamma = 0$.

Step2: クライアント・プロセスより受け取った p 個の解より解集合 P を作成する.

Step3: 解集合 P の全ての非順序対に対しアルゴリズム HKS を適用し交配する.

Step4: Step3で求められた $p(p-1)/2$ の解より $p-m$ 個までの目的関数の良い解と残った解からランダムに m 個選ぶ. 選ばれた p 個の解により集合 P を更新し, 次世代を構成する.

Step5: 終了条件を満たしていれば得られた最良解を出力し終了. そうでなければ, Step3へ.

提案分散遺伝的アルゴリズムにおいても入力データの規模により十分良質な解を得るために必要な人口数が増加することが予測されるが, 分散実行による計算時間の短縮率も増加すると考えられるので, 分散実行の有効性は保持されると予想できる.

4. 4 実験結果

4. 3 で提案した手法を SPARCstation 2, SPARCstation IPX 各 1 台, SPARCstation ELC 2 台, 計 4 台のワークステーションより構成される分散システム上に C 言語を用いて実現した. 通信方法としてはソケットを使用した. 実験データとしては Ind2 を用いた. 表 2 にアルゴリズム DGHKS の実験結果を示す. ここで, アルゴリズム DGHKS のパラメータはクライアント・プロセスでは $p=10$, $m=2$, $R=2$, サーバ・プロセスでは $p=10$, $m=2$, $R=10$ とした. 実験は 20 個のランダムな初期状態から始めた. 表 2 では, サーバ・プロセスにおける各世代ごとの 20 個のカット数の最小と平均, 及び, 改良度を示す. 実験結果より, アルゴリズム DGHKS は多くの世代にわたり解を改良でき, 逐次アルゴリズム GHKS においてパラメータを $p=15$, $m=2$, $R=10$ とした場合の平均カット数を 6.7% 改良した.

表 2. アルゴリズム DGHKS の実験結果

世 代	1	2	3	4	5	6	7	8	9	10
カット数 (最小)	560	442	371	345	330	299	294	290	287	279
カット数 (平均)	735	602	513	470	388	350	331	330	324	320
改良度 [%]	-	18	15	8	17	10	5	0	2	1

5. あとがき

本稿では遺伝的アルゴリズムに基づいたハイパーグラフk分割手法の分散アルゴリズム化を考察した。解集合を多様化し、局所最適解への収束の回避を試みた。提案手法を分散システム上に実現し、実験的評価を行った。実験の結果、提案した分散遺伝的アルゴリズムは良質な解を短い計算時間で得るために有効であることが分かった。今後、さらにアルゴリズムの改良と計算機実験による評価を行う予定である。

謝辞 日頃、熱心なご指導を賜ります広島大学工学部教授 吉田典可先生および助手 小出哲士先生に深謝致します。また、実験を行うにあたってご協力頂いた本学学部生 岸本善久君に感謝します。

本研究の成果の一部は文部省科学研究費補助金一般研究(C) (課題番号05680274) による。

文献

- [1] A.V.Aho, J.E.Hopcroft and J.D.Ullman : "Data Structures and Algorithms," Addison-Wesley (1983).
- [2] J. P. Cohoon, S. U. Hegde, W. N. Martin and D. S. Richards : " Distributed genetic algorithms for the floorplan design problem, " IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 10, No. 4, pp. 483 - 491 (1991).
- [3] C.M.Fiduccia and R.M.Mattheyses : "A linear-time heuristic for improving network partitions," Proc. 19th ACM/IEEE Design Automation Conference, pp.175 - 181 (1982).
- [4] M.R.Garey and D.S.Johnson : "Computers and Intractability : A Guide to the Theory of NP- Completeness," W. H. Freeman (1979).
- [5] D.E.Goldberg : "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley (1989).
- [6] J.J.Grefenstette : "Genetic algorithms for changing environments," in "Parallel Problem Solving from Nature 2," R. Manner and B.Manderick (Editors), pp.137-144, Elsevier Science Publishers B.V. (1992).
- [7] R. Jones and M.A.Beltramo : "Solving partitioning problems with genetic algorithms," Proc. 4th International Conf. on Genetic Algorithms, pp.442- 449 (1991).
- [8] A.B.Kahng : "Fast hypergraph partition," Proc. 26th ACM/IEEE Design Automation Conf., pp. 762-766 (1989).
- [9] Y.Kamidoi, S.Wakabayashi and N. Yoshida : "An efficient hypergraph bisection algorithm for partitioning VLSI circuits," IEICE Trans. Fundamentals, Vol. E75-A, No.10, pp.1272-1279 (1992).
- [10] Y.Kamidoi, S.Wakabayashi and N. Yoshida : "An efficient GA hybrid for hypergraph bisection with application to VLSI placement," Proc. IEEE Asia-Pacific Conf. on Circuits and Systems 1992, pp.369 - 401 (1992).
- [11] 上土井, 若林, 吉田: "ハイパーグラフをk分割する遺伝的分散アルゴリズムとその実験的評価," 信学技報, COMP92-97 (1993).
- [12] B.W.Kernighan and S.Lin : "An efficient heuristic procedure for partitioning graphs," Bell System Technical Journal, 49(2), pp.291-307 (1970).
- [13] B.Krishnamurthy : "An improved min-cut algorithm for partitioning VLSI networks," IEEE Trans. on Computers, Vol.C-33, No.5, pp.438-446 (1984).
- [14] T.Lengauer : "Combinatorial Algorithms for Integrated Circuit Layout," Wiley (1990).
- [15] L.A.Sanchis : "Multiple-way network partitioning," IEEE Trans. on Computers, Vol. 38, No. 1, pp. 62-81 (1989).
- [16] H.M. Voigt, I.S. Koref and J.Born : "Hierarchically structured distributed genetic algorithms," in "Parallel Problem Solving from Nature 2," R.Manner and B.Manderick (Editors), pp.155 -163, Elsevier Science Publishers B.V. (1992).