

分散制約最適化問題のための解集合を非集中管理する進化的計算手法の検討

松井俊浩 †

名古屋工業大学 †

1 はじめに

分散制約最適化問題 [1] はマルチエージェントシステムにおける協調問題解決の基礎として研究されている。分散制約最適化問題の非厳密解法として、進化的計算に基づく解法 AED [2] が提案されているが、各エージェントが系全体のエージェントの変数値の割り当てを含む完全解からなる個体群を持つため、各エージェントは無関係なエージェントの変数値を得ずに問題を解決するという、分散制約最適化問題の動機の一つを満足しない。そこで、本研究では、各エージェントが無関係な変数値を取得せずに、各個体群に含まれる自身の変数値を保管する方式の解法について検討する。

2 従来手法

2.1 分散制約最適化問題

分散制約最適化問題はエージェントの集合 A 、変数の集合 X 、変数の値域の集合 D 、制約のコスト関数の集合 F により表される。変数 $x_i \in X$ はエージェント $a_i \in A$ の意思決定や状態を表す。変数 x_i は離散値の値域 $D_i \in D$ に含まれる値を取る。非負整数値のコスト関数 $f_{i,j}(x_i, x_j) \in F$ により x_i と x_j の値が評価される。全てのコスト関数値の合計を最小化する変数値の割り当てが最適解である。エージェント a_i は x_i の値を探索できる。制約密度が比較的高く、多数の変数からなる問題の解決には準最適解を得る非厳密解法が必用である。

2.2 進化的計算に基づく解法 AED

AED [2] は、進化的計算に基づく、分散制約最適化問題の解法であり、制約グラフ上の近傍エージェント間および単一の幅優先探索 (BFS) 木上の情報交換を伴う同期型の分散アルゴリズムである。AED の疑似コードを図 1 に示す。この処理では、全変数への値の割り当てからなる最適解の候補の集合である個体群および各個体 I に付随するコスト値 $I.fitness$ を操作する。BFS 木に基づくボトムアップおよびトップダウンな前処理により、全ての変数についての初期解の個体群と各評価値が各エージェント a_i の P_{a_i} として共有される。各

```

1 Construct a BFS tree on a constraint graph.
2 Share an initial set of individuals  $P_{a_i}$  by a protocol on the BFS tree.
3  $Itr \leftarrow 0$ .
4 until  $Itr$  is less than a cutoff cycle do begin
5    $P_{selected} \leftarrow$  a set of  $|N_i| \times ER$  individuals sampled from  $P_{a_i}$  allowing
     to select the same elements.
6    $P_{new} \leftarrow \{P_{new}^{n_1}, \dots, P_{new}^{n_{|N_i|}}\}$  consisting of sets of the same size
     generated from  $P_{selected}$  by partitioning its elements.
7   for  $n_j$  in  $N_i$  do begin
8     Update individuals in  $P_{new}^{n_j}$  by sampling each assignment to  $a_i$ 's
       variable.
9     Send  $P_{new}^{n_j}$  to  $n_j$ .
10  end
11 for  $P_{new}^{n_k}$  received from  $n_k$  in  $N_j$  do begin
12   Update individuals in  $P_{new}^{n_k}$  by selecting each best assignment to
      $a_i$ 's variable.
13   Return  $P_{new}^{n_k}$  to  $n_k$ .
14 end
15 for  $P_{new}^{n_j}$  returned from  $n_j$  in  $N_j$  do begin
16    $P_{a_i} \leftarrow P_{a_i} \cup P_{new}^{n_j}$ .
17 end
18  $B \leftarrow \operatorname{argmin}_{I \in P_{a_i}} I.fitness$ .
19 Update and commit the globally best solution using  $B$  by a protocol
   on the BFS tree executing in background.
20  $P_{a_i} \leftarrow$  a set of  $|N_i| \times ER$  individuals sampled from  $P_{a_i}$  disallowing
   to select the same elements.
21 if  $Itr \bmod MI = 0$  then begin
22   for  $n_j$  in  $N_i$  do begin
23     Send a set of  $ER$  individuals, which is sampled from  $P_{a_i}$ 
       disallowing to select the same elements, to  $n_j$ .
24   end
25   for  $P_{migrated}^{n_k}$  received from  $n_k$  in  $N_j$  do begin
26      $P_{a_i} \leftarrow P_{a_i} \cup P_{migrated}^{n_k}$ .
27   end
28 end
29  $Itr \leftarrow Itr + 1$ .
30 end

```

図 1: AED (agent a_i)

エージェントは以降の処理を反復し、自身が持つ個体群を局所的に更新する。まず、エージェント a_i は制約で関係する各近傍エージェント $a_j \in N_i$ に渡す個体群 P_{new}^j を P_{a_i} からサンプリングし、 P_{new}^j の各個体の自変数値をサンプリングし更新する。 P_{new}^j は近傍エージェント a_j に渡され、 a_j は各個体の自変数値を最良の評価値となるように更新し、返却する。返却された個体群は P_{a_i} に結合され、そこから最良解 B が選択される。解法の背後で動作する分散スナップショットアルゴリズムにより、大域的な最良解が更新され全エージェントで同期される。 P_{a_i} はサイズを維持するために再度サンプリングされる。個体群の多様性を維持するために、マイグレーション間隔 MI の反復ごとに、各エージェントは近傍エージェントから個体群の一部を取り込む。詳細については文献 [2] を参考にされたい。

A Study on Evolutionary Algorithm with Decentralized Management of Solution Sets for Distributed Constraint Optimization Problem
†Toshihiro Matsui · Nagoya Institute of Technology

3 提案手法

3.1 解集合の分散配置

全エージェントの個体群を同一の変数ごとに分割し、対応する変数を持つエージェントに分散して格納する。各エージェントは全ての個体と評価値を格納するが、各個体の自変数値しか格納しない。自身の評価に必要な変数の値は反復ごとに取得して管理するが、それ以外の変数は知らない。反復ごとに管理する変数の範囲を同期範囲と呼び、自身、近傍エージェント、その近傍エージェントの変数値を含む。それ以外の範囲を非同期範囲と呼ぶ。これらの個体群を管理するために、各個体に、所有するエージェントの識別子、同期範囲および非同期範囲の識別子を与える。個体群は変数値ではなく、識別子により管理される。反復ごとの管理では、各個体の同期範囲の識別子のみが更新され、非同期範囲の識別子は継承される。多様性の減少により継承されなかった非同期範囲の識別子は破棄される。

3.2 非集中型マイグレーション

マイグレーションでは非同期範囲を含む個体群の移動が必要となるため、個体群の移動の情報を BFS 木に沿って系全体に伝搬する。伝搬には $2(\text{BFS 木の高さ}) - 1$ の反復が必要であり、マイグレーション間隔 MI の最小値となる。各エージェントは個体群を 2 世代にわたって格納し、古い個体から新しい個体に自変数値を継承し、新しい個体の非同期範囲の識別子を更新する。継承されなかった個体は破棄される。これにより非同期範囲について格納される個体数が圧縮される。

3.3 最良解を同期するスナップショットアルゴリズム

最良解の同期では、最良解の情報は個体の各識別子に基づいて管理され伝搬される。各ノードは選択すべき自変数値を識別子から把握する。タイムスタンプを管理し、マイグレーションの開始時点よりも前に発生したスナップショットが伝搬したときは格納されている古い個体の情報を参照する。マイグレーションとスナップショットアルゴリズムの伝搬の送受信のタイミングを適切に調整することにより、同一の反復回数内に伝搬が完了する。

3.4 近傍関係の暴露の緩和

提案手法は変数値を隠ぺいするが、マイグレーションの伝搬において、エージェント間の関係を暴露する。そこで、非同期範囲の個体群を全エージェントで共通とし、予測が容易ではない一意な識別子により管理する。しかし、最良解の同期の際に、最良解を評価した

表 1: 解品質と実行コスト

prb.	alg.	fit.	max. sz. agt. store	time [s]
$d = 3$ $c = 250$	aed	8974.6	8050	40
	ds	8975.1	9998	61
	dssn	8975.1	10001	69
$d = 5$ $c = 150$	aed	4007	5650	30
	ds	4014	5768	28
	dssn	4014	5546	29

エージェントの識別子と、新たな識別子の対応が暴露される。そこで、最良解を評価したエージェントの識別子も同様の識別子により管理する。各識別子は各処理の反復ごとに更新され、必要な期間のみ存在するため、十分な桁数の疑似乱数を用いたときに衝突する可能性はごく小さく、衝突を検出できる。

4 評価

従来手法 aed と、提案手法のうち近傍関係の暴露を緩和しない ds および緩和する dssn を実験により評価した。aed のパラメータを同一とし、マイグレーション間隔 MI のみ aed では 5、提案手法では許容される最小値とした。変数/エージェント数 50、変数の値域のサイズ d 、制約数 c 、一様分布に基づく $[1, 100]$ のランダムな整数コスト値の関数の例題における、1000 回の反復の結果を表 1 に示す。コスト値 fit. は概ね同様であるが、提案手法の MI が平均 6 から 6.9 と比較的大きいことが影響した。解法の実行コストは、制約密度が高い $c = 250$ の問題では提案手法のオーバヘッドの影響が比較的大きく現れた一方で、制約密度 3 の $c = 150$ の場合では従来手法と同等以下であった。各エージェントが格納する解の変数値の個数の最大値 max. sz. agt. store については、提案手法はマイグレーションのために 2 世代の個体群を格納するが、非同期範囲の解を圧縮した効果により従来手法の 2 倍よりも十分に小さかった。

5 まとめ

分散制約最適化問題の進化的計算に基づく非厳密解法 AED に基づき、各エージェントが無関係な変数値を取得せずに、各個体群に含まれる自身の変数値を保管する方式の解法を提案した。

謝辞 本研究の一部は JSPS 科研費 JP19K12117 の助成による。

参考文献

- [1] Fioretto et al. Distributed constraint optimization problems and applications: A survey. *JAIR*, Vol. 61, pp. 623–698, 2018.
- [2] Mahmud et al. AED: An Anytime Evolutionary DCOP Algorithm. In *AAMAS*, pp. 825–833, 2020.