

NAIST-IS-DT0061023

博士論文

マルチエージェント技術を用いた組合せ最適化問題に
対する近似アルゴリズム

村田 佳洋

2003年2月24日

奈良先端科学技術大学院大学
情報科学研究科 情報処理学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
博士(工学) 授与の要件として提出した博士論文である。

村田 佳洋

審査委員： 伊藤 実 教授
石井 信 教授
杉本 謙二 教授
安本 慶一 助教授

研究業績

● 査読付き投稿論文

- An Approximation Algorithm for the Task-Coalition Assignment Problem. Murata, Y. Ishihara, Y. Ito, M. The Institute of Electronics, Information and Communication Engineers, Vol.E85-D No.4 pp.685-693, 2002.
- エージェント指向自己適応遺伝アルゴリズム. 村田 佳洋, 柴田 直樹, 伊藤 実. 「情報処理学会論文誌:数理モデル化と応用」(TOM) に投稿, 採録決定

● 国際学会発表

- Agent Oriented Self Adaptive Genetic Algorithm. Murata, Y. Shibata, N. Yasumoto, K. and Ito, M. Communications and computer networks(CCN), November 4-6, 2002 cambridge, USA, pp. 348-353, 2002.

● 研究会発表

- 仕事に対するエージェント集合割り当て問題の近似アルゴリズム. 村田 佳洋, 石原 靖哲, 伊藤 実. 電子情報通信学会技術研究報告, COMP99-56 ~ 68, p.25-32, 1999 年 12 月 10 日 .
- エージェント指向自己適応遺伝アルゴリズム. 村田 佳洋, 柴田 直樹, 伊藤 実. 情報処理学会研究報告, Vol.2002, No.89, 2002-MPS-41, pp. 21-24, 2002.

● 卒業論文及び修士論文

- 島モデルを用いた熱力学的遺伝アルゴリズムの提案. 大阪府立大学卒業論文,1998 年 .
- 仕事に対するエージェント集合割り当て問題の近似アルゴリズム. 奈良先端科学技術大学院大学修士論文, 2000 年 .

マルチエージェント技術を用いた組合せ最適化問題に 対する近似アルゴリズム*

村田 佳洋

内容梗概

本論文は，筆者が奈良先端科学技術大学院大学情報科学研究科に在学中に行ったマルチエージェント技術を用いた近似アルゴリズムに関する研究をまとめたものである．

NP 困難のクラスに属する組合せ最適化問題は多項式時間で最適な解を導くことができないと予想されており，従来から，現実的な時間でより良い近似解を導くための様々な近似アルゴリズムが研究されている．一方，分散計算技術の発達により，自律した複数の計算機もしくはプロセスが協力して問題を解く，マルチエージェント技術が近年注目されるようになってきた．

本論文では，NP 困難のクラスに属する問題を解くためのマルチエージェント技術を用いた近似アルゴリズムに関する 2 つの研究を行った．最初に，仕事へのエージェント集合割り当て問題に対する近似アルゴリズムについて述べ，次に，より一般的な問題に対して，計算に必要なパラメータを自動的に適応するエージェント指向自己適応遺伝アルゴリズムについて述べている．

仕事割り当て問題としてのエージェントの協力問題の定式化の一つに，仕事へのエージェント集合割り当て問題 (Task-Coalition Assignment Problem, 以下 TCAP) がある．TCAP は Shehory らによって提案され，形式化が試みられている．本論文の 2 章では，まず TCAP の定義を示し，次に TCAP が最大クリーク問題を含むことを証明する．これにより，最適解の定数倍の近似解を求める多項

*奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文, NAIST-IS-DT0061023, 2003 年 2 月 24 日.

式時間近似アルゴリズムは、 $P=NP$ でない限り存在しないことが示される。次に、TCAP の特別な場合である 1 次元仕事へのエージェント集合割り当て問題 (1-TCAP) を以下のように定義する。1-TCAP は、エージェントの有限集合 A 、仕事の有限集合 T 、エージェントの能力値と仕事の仕事量を与える関数 \cdot を入力とし、エージェント集合が仕事の処理を試みた場合に得られる利益を与える関数 $f(C, t)$ ($C \subseteq A$) が与えられているときに、利益の総和を最大にする割り当てを求めめる問題である。1-TCAP では、エージェントと仕事はスカラー量で特徴づけられる。まず、subset sum 問題が 1-TCAP に帰着できることから 1-TCAP が NP 困難のクラスに属することを示す。次に、1-TCAP に対する多項式時間近似アルゴリズムを提案し、その近似解の値 v と最適解の値 v^* の比率 v^*/v が最悪の場合でも $9/4$ 未満であることを示す。

次に、本論文の 3 章において、遺伝アルゴリズムにおける解探索の際のパラメータを自動的に適応する、エージェント指向自己適応遺伝アルゴリズム (Agent oriented Self-Adaptive Genetic Algorithm, 以下 A-SAGA) を提案する。遺伝アルゴリズム (Genetic Algorithm, 以下 GA) は生物の進化を模した最適化アルゴリズムであり、広範な領域の問題に適用可能な手法として期待されている。しかし、GA の探索効率は与えられるパラメータ (突然変異率、交叉率など) の組み合わせによって大きく左右されるという問題があり、多くのパラメータの調整を人手で行うのは困難である。そこで、パラメータを自動的に調整する、様々な適応 GA が提案されている。しかし、従来の適応 GA のほとんどは 1 ないし 2 つのパラメータしか適応させられず、多数のパラメータを適応させる適応 GA であっても、そのほとんどが大きな計算量を必要としていた。A-SAGA は、合理的な計算量で、多数のパラメータを同時に適応させることができる。また、適応 GA の最大の目的は、パラメータ適応の労力を軽減することである。その観点から、適応 GA を用いるためには、必要な予備知識や事前定義が少なければ少ないほど望ましいと考えられる。一般に、GA を適用するためには、適応度の評価方法と遺伝的演算子 (交叉演算子や突然変異演算子) の定義を与えることが必要である。いくつかの適応 GA はそれに加えて多様性などに関する定義を必要とし、これらを利用して探索効率を高めている。しかし、これらを定義することは、問題によ

ては労力を必要とする．A-SAGA は，適応度の評価方法と遺伝的演算子の定義が与えられれば，各島に与えられるパラメータがどのような組合せでも適用可能であるという特徴を持つ．本手法の有効性を，メタ GA との探索効率の比較実験により評価し，4つのパラメータが合理的な計算量で同時に適応させられることを示す．また，エージェントの自律性による拡張性について考察する．

キーワード

エージェント, 近似アルゴリズム, 組合せ最適化問題, 近似可能性, 遺伝アルゴリズム

目次

1. はじめに	1
1.1 エージェントの協力問題の側面	1
1.2 本論文の構成	3
2. 仕事へのエージェント集合割り当て問題 (TCAP) と近似アルゴリズム	5
2.1 導入	5
2.2 TCAP の形式的定義	6
2.3 TCAP の近似可能性	7
2.4 1-TCAP の形式的定義	8
2.5 1-TCAP の NP 困難性	10
2.6 1-TCAP に対する近似アルゴリズム	12
2.7 本章の結び	29
3. エージェント指向自己適応遺伝アルゴリズム	31
3.1 序文	31
3.2 関連研究	31
3.3 提案手法	33
3.3.1 従来手法の問題点	33
3.3.2 提案手法における改良点	33
3.3.3 予備実験とそれに対する考察	36
3.3.4 エージェント指向自己適応遺伝アルゴリズム	36
3.4 実験	38
3.4.1 実験条件	38
3.4.2 実験結果	40
3.4.3 考察	41
3.5 エージェント指向による拡張性についての考察	44
3.6 本章の結び	45
4. おわりに	46

謝辭	47
參考文獻	48

目 次

1	荷物の運送問題	6
2	グラフ G	8
3	最大クリーク問題から TCAP への変換	8
4	1-TCAP の問題例	10
5	1-TCAP の解の例	10
6	降順連続当てはめアルゴリズム (SONFA)	12
7	SONFA の動作例	13
8	SONFA にとって最悪の場合のインスタンス	14
9	最悪の場合の SONFA の解	15
10	最悪の場合の最適解	15
11	2 章 4 節における表記法	16
12	エージェント a_l と仕事 t_k	16
13	A-SAGA エージェントアルゴリズム	35
14	A-SAGA アルゴリズム	35
15	A-SAGA 概念図	38
16	Rastrigin 個体適応度	40
17	TSP eil51 個体適応度	40
18	TSP eil51 個体適応度 (2)	40
19	Rastrigin p_m	40
20	Rastrigin e_p	41
21	TSP eil51 p_m	41
22	TSP eil51 e_p	41

表 目 次

1. はじめに

NP 困難のクラスに属する組合せ最適化問題は多項式時間で最適な解を導くことができないと予想されており，従来から，現実的な時間でより良い近似解を導くための様々な近似アルゴリズムが研究されている．

このような問題を解く上で単一の高性能なプロセッサを利用するよりも，廉価な汎用プロセッサを多数利用するほうがコスト的に有利である場合がある．多数のプロセッサを協調させて効率的に問題を解くために，分散計算技術が研究されている．制御が分散されるほど，それぞれの制御主体は個体としての性質を帯びてくる．1970年代から研究されている DAI(distributed artificial intelligence) は，それぞれの制御主体により大きな判断力と権限を与えることにより，全体としてうまく働くシステムを構築することを目的としている．エージェント技術はそれらの研究の一側面である．エージェントは自律して行動する主体であるが，どのような性質を持っていれば自律しているとみなすかについては，研究者によって意見が分かれている．そのため，エージェントに関する研究分野は広範囲にわたっている [17][45]．

1.1 エージェントの協力問題の側面

本節では，エージェントの協力問題の様々な側面について述べる．

分散計算機の連動アルゴリズムの設計は勘と経験とによるところが大きい．そこで計算機をエージェントと見なし，分散計算問題をエージェントの協力問題として定式化し性能を評価する研究がなされている．

[例 1] エージェントの協力問題の例として，宅配便の運送問題を示す．入力として宅配エージェントと仕事を与えられる．各エージェントは自分の利益の総和を最大化する事を目的とする利己的エージェントであり，利益は仕事を処理することで得ることができる．仕事は現在位置，荷物の量，目的地，処理した場合に得られる利益により特徴づけられる．

- スケジューリング問題としての側面

仕事を処理する順番により利益が変化する場合がある．このときの利益を

最大化するような順番を求める問題がスケジューリング問題である [15] [18] [43] . これはシステムがリアルタイム処理を行う際に特に重要となる . この問題には 2 つの系統がある . より大きなシステムのスケジュールをエージェントが決定する問題 [5] と , それぞれのエージェントが自分のスケジュールを決定する問題 [3] である .

- 集合分割問題としての側面

エージェントが互いに協力しあう集団のことを *coalition* と呼ぶ . エージェントの全体集合を *coalition* に分ける問題は集合分割問題であり , これを解くための手法がいくつか提案されている [35] [36] [41] .

- 仕事割り当て問題としての側面

各エージェントに仕事を与えられる場合と , 全体に仕事を与えられてそれをエージェント間で分配する場合がある . 後者の場合における分配方法を見つける問題が仕事割り当て問題である .

- ゲーム理論としての側面

エージェントがそれぞれ利己的である場合 , エージェントの協力問題を各エージェントをプレーヤーとした N 人ゲームとしてとらえ , 古典的ゲーム理論を利用してエージェント集団の振舞いを解析することができる [21] . また , エージェント同士の利害が衝突した場合 , 交渉することによって双方が同意できる意見を探す方法も研究されている [19] [22] [37] [38] .

また , エージェントの協力問題はシステムの規模によっても分類される [23] . 小規模なシステムでは他エージェントや環境の情報を集めることが容易であるため , エージェントは詳細な情報に基づく戦略を立てることができる . ところが大規模なシステムでは全エージェントが互いに通信することが通信量的に困難であるため , エージェントはほとんどの情報が未知なまま戦略を立てねばならない [2] [27] [39] .

エージェントの協力問題において想定されるエージェントの能力はさまざまである . ゲーム理論の要素の強い問題ではエージェントは人間と同程度の知恵を持つと仮定される . これは主に *open space* な環境が想定されている場合では , 人間が

特にその環境や他エージェントに対して有利であるようなエージェントを新たに設計して送り込むことができるためである．逆に検索ロボットによる WWW(World Wide Web) 上検索問題等では，エージェントのサイズが小さいことが重要になり，その能力は限定される．これはサイズが小さいほうが通信負荷が軽減されることによる．

以上のように，エージェントの協力問題は様々な側面を持っており，そのうちのいくつかは NP 困難のクラスに属する．NP 困難のクラスに属する組合せ最適化問題は多項式時間で最適な解を導くことができないと予想されており，従来から，現実的な時間でより良い近似解を導くための様々な近似アルゴリズムが研究されている．

1.2 本論文の構成

本論文では，NP 困難のクラスに属する問題を解くためのマルチエージェント技術を用いた近似アルゴリズムに関する 2 つの研究を行った．最初に，仕事へのエージェント集合割り当て問題に対する近似アルゴリズム [28] [29] [30] について述べし，次に，より一般的な問題に対して，計算に必要なパラメータを自動的に適応するエージェント指向自己適応遺伝アルゴリズム [31] [32] [33] について述べている．

仕事割り当て問題としてのエージェントの協力問題の定式化の一つに，仕事へのエージェント集合割り当て問題 (Task-Coalition Assignment Problem, 以下 TCAP) がある．TCAP は Shehory らによって提案され，形式化が試みられている．本論文の 2 章では，TCAP の概要を示した後 (2.1 節)，TCAP の定義を示し (2.2 節)，次に TCAP が最大クリーク問題を含むことを証明する (2.3 節)．これにより，最適解の定数倍の近似解を求める多項式時間近似アルゴリズムは， $P=NP$ でない限り存在しないことが示される．次に，TCAP の特別な場合である 1 次元仕事へのエージェント集合割り当て問題 (1-TCAP) を定義する (2.4 節)．1-TCAP は，エージェントの有限集合 A ，仕事の有限集合 T ，エージェントの能力値と仕事の仕事量を与える関数 τ を入力とし，エージェント集合が仕事の処理を試みた場合に得られる利益を与える関数 $f(C, t)$ ($C \subseteq A$) が与えられているときに，利益

の総和を最大にする割り当てを求める問題である。1-TCAP では、エージェントと仕事はスカラー量で特徴づけられる。まず、subset sum 問題が 1-TCAP に帰着できることから 1-TCAP が NP 困難のクラスに属することを示す (2.5 節)。次に、1-TCAP に対する多項式時間近似アルゴリズムを提案し、その近似解の値 v と最適解の値 v^* の比率 v^*/v が最悪の場合でも $9/4$ 未満であることを示す (2.6 節)。

次に、遺伝アルゴリズムにおける解探索の際のパラメータを自動的に適応する計算に必要なパラメータを自動的に適応する、エージェント指向自己適応遺伝アルゴリズム (Agent oriented Self-Adaptive Genetic Algorithm, 以下 A-SAGA) を提案する (3 章)。遺伝アルゴリズム (Genetic Algorithm, 以下 GA) は生物の進化を模した最適化アルゴリズムであり、広範な領域の問題に適用可能な手法として期待されている。しかし、GA の探索効率は与えられるパラメータ (突然変異率、交叉率など) の組み合わせによって大きく左右されるという問題があり、多くのパラメータの調整を人手で行うのは困難である。そこで、パラメータを自動的に調整する、様々な適応 GA が提案されている。しかし、従来の適応 GA のほとんどは 1 ないし 2 つのパラメータしか適応させられず、多数のパラメータを適応させる適応 GA であっても、そのほとんどが大きな計算量を必要としていた。A-SAGA は、合理的な計算量で、多数のパラメータを同時に適応させることができる。まず、適応 GA の関連研究について述べ (3.2 節)、提案手法について説明し (3.3 節)、本手法の有効性を、メタ GA との探索効率の比較実験により評価し、4 つのパラメータが合理的な計算量で同時に適応させられることを示す (3.4 節)。

2. 仕事へのエージェント集合割り当て問題 (TCAP) と近似アルゴリズム

2.1 導入

本章では，仕事へのエージェント集合割り当て問題 (TCAP) について行った研究について述べる．

集合分割問題かつ仕事割り当て問題としてのエージェントの協力問題の定式化の一つに仕事へのエージェント集合割り当て問題 (Task-Coalition Assignment Problem, 以下 TCAP) がある．

[例 2] TCAP の一例として荷物の運送問題を図 1 に示す．荷物を運送するという仕事と，仕事を処理するためのエージェントがそれぞれ複数与えられる．この例題では仕事として島への運送，ビルへの運送，家への運送の 3 つが与えられ，エージェントとして 2 隻の船，2 台の小型トラック，1 台の大型トラック，1 機のヘリコプターが与えられている．

仕事に対してこれらのエージェントから成る coalition を割り当て，処理できた場合には利益が得られる．利益は coalition と仕事の組合せによって変化する．例えば島への運送にトラックを用いることはできないし，トラックが通行できる場所へヘリコプターを用いるのは燃費や積載量の関係で好ましくないだろう．また積載量によっては 1 台のトラックでは処理できない場合があるかもしれない．

TCAP は Shehory らによって提案され，形式化が試みられている [40][42]．本章では，TCAP に具体的な制約を加えることにより，1 次元仕事へのエージェント集合割り当て問題 (1-TCAP) を定義する．1-TCAP ではエージェントと仕事はスカラー量で特徴づけられる．また，subset sum 問題が 1-TCAP に帰着できることから 1-TCAP が NP 困難であることを示す．そして，1-TCAP に対する多項式時間近似アルゴリズムを提案し， $9/4$ の近似比を達成することを示す (2.6 節)．ここで， α の近似比を達成するとは，すべてのインスタンスに対し，最低でも OPT/α の値の解をつくり出すことを示す．ここで OPT は最適解の値である．最後に結論及び今後の課題について述べる (2.7 節)．

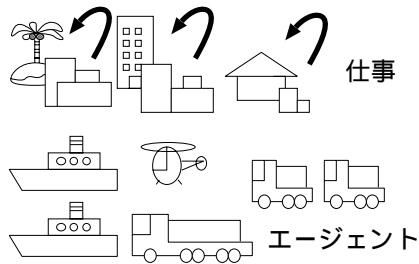


図 1 荷物の運送問題

2.2 TCAP の形式的定義

[定義 1] 仕事へのエージェント集合割り当て問題 (TCAP) を次のように定義する。まず、入力は以下の 3 つである。

- エージェントの有限集合 A .
- 仕事の有限集合 T .
- 利益関数 $f : 2^A \times T \rightarrow \mathbf{R}$. ここで \mathbf{R} は非負の実数の集合である。利益関数は coalition $C \subseteq A$ を仕事 t に割り当てた時の利益を返す関数である。ただし

$$- f(\emptyset, t) = 0 .$$

空の coalition を仕事に割り当てても、利益は 0 である。

$$- f(C, t) > 0 \text{ のとき, 任意の } C' \supseteq C \text{ について } f(C', t) \leq f(C, t) .$$

ある coalition C が仕事 t を処理できる場合、 C に別のエージェントを加えた C' で同じ仕事を処理すると利益が同じか下がる。これは大きな coalition になるほど通信等のコストが大きくなることを示している。

そして、目的は利益の総和

$$\sum_{t \in T} f(\sigma(t), t)$$

を最大化する割り当て $\sigma : T \rightarrow 2^A$ を求めることである。ただし異なる $t, t' \in T$ に対し $\sigma(t) \cap \sigma(t') = \emptyset$ でなければならない。すなわち、エージェントは複数の

coalition に所属することはできない。

2.3 TCAP の近似可能性

本節では TCAP の近似可能性について述べる。まず、TCAP が最大クリーク問題を含むことを示す。

[定理 1] 最大クリーク問題は TCAP の特別な場合である。

証明： $G = (V, E)$ に対し、TCAP のインスタンスを以下のように構成する。

- エージェント集合 $A = V$ ：グラフの各頂点をエージェントとみなす。
- 仕事集合 $T = \{t\}$ ：仕事は 1 つだけ存在するとする。
- 利益関数

$$f(C, t) = \begin{cases} |V - C| & V - C \text{ がクリークを形成するとき,} \\ 0 & \text{そうでないとき.} \end{cases}$$

このように構成された f は TCAP の定義中の利益関数についての条件を満たす。また、この問題を解くことは $G = (V, E)$ の最大クリークの頂点数を求めることと同義である。

図 3 に概念図を示す。仕事 t に対し黒い頂点のエージェントからなる coalition を割り当てたとする。このとき、残った白い頂点がクリークを形成するかどうかによって利益が定まる。クリークを形成する場合、そのクリークの頂点数が利益となる。逆にクリークを形成しない場合、利益は 0 となる。つまり、仕事 t に割り当てる頂点 (エージェント) を増やすほど残りの頂点数が減ることから、クリークを形成しやすくなると同時に形成したときの利益が減少する。これは TCAP の利益関数の条件を満たしている。図 3 の場合では残った 3 頂点がクリークを形成しているため、利益は 3 となる。

最大クリーク問題はクラス IV に属していることが知られている [16]。クラス IV とは、“ある定数 $\delta > 0$ が存在して、最適解の $\frac{1}{n^\delta}$ の近似解を求めることは NP

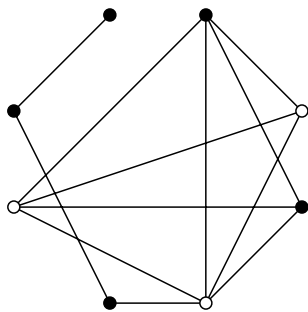


図 2 グラフ G

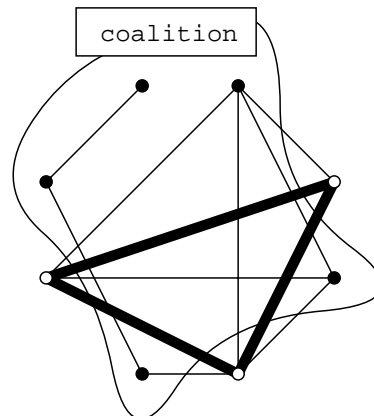


図 3 最大クリーク問題から TCAP への変換

困難である” ような最適化問題のクラスである (ただし n は問題のインスタンスのサイズ). したがって TCAP は少なくともクラス IV に属する.

近似解の利益の総和に対する最適解の利益の総和の比を近似比と呼ぶことにする. ただし, 近似解の利益の総和と最適解の利益の総和がともに 0 であるときは, 便宜上近似比を 1 と定める. この定義より, 近似比は必ず 1 以上になることができる. 上で述べたように, TCAP は少なくともクラス IV に属することから, 最悪の場合の近似比が定数である (問題のインスタンスのサイズに依存しない) ような多項式時間近似アルゴリズムは, $P=NP$ でない限り存在しない.

2.4 1-TCAP の形式的定義

TCAP のサブクラスとして 1 次元 TCAP (以下, 1-TCAP) を提案する. 1-TCAP では, エージェントは能力値で, 仕事は仕事量で特徴づけられる. 能力値と仕事量はともにスカラー量である. エージェントが coalition を形成したとき, その coalition の能力値は参加したエージェントの能力値の総和である. coalition の能力値が仕事の仕事量以上である場合, その仕事を処理できる. そのときの利益は, 仕事の仕事量に等しい. 以下に 1-TCAP を定義する.

[定義 2] 一次元仕事へのエージェント集合割り当て問題 (1-TCAP) を次のように定義する．まず，入力は以下の 3 つである．

- エージェントの有限集合 A ．
- 仕事の有限集合 T ．
- エージェントの能力値を与える関数 $\bar{a} : A \rightarrow \mathbb{R}$ ．ここで \mathbb{R} は非負の実数の集合である． \bar{a} ($a \in A$) はエージェント a が持つ能力値を表す ．
- 仕事の仕事量を与える関数 $\bar{t} : T \rightarrow \mathbb{R}$ ． \bar{t} ($t \in T$) は仕事 t が持つ仕事量を表す ．

ここで， $C \subseteq A$ に対し，coalition C の能力値を $\bar{C} = \sum_{a \in C} \bar{a}$ と定義する．利益関数 f は次のように定義される ．

$$f(C, t) = \begin{cases} 0 & \bar{C} < \bar{t} \text{ の場合 ,} \\ \bar{t} & \bar{C} \geq \bar{t} \text{ の場合 .} \end{cases}$$

そして，目的は利益の総和

$$\sum_{t \in T} f(\sigma(t), t)$$

を最大化する割り当て $\sigma : T \rightarrow 2^A$ を求めることである．ただし異なる $t, t' \in T$ に対し $\sigma(t) \cap \sigma(t') = \emptyset$ でなければならない ．

1-TCAP の例を図 4 に示す ．ここではエージェント集合 $A = \{a_1, a_2, \dots, a_5\}$ と仕事集合 $T = \{t_1, t_2, \dots, t_5\}$ が与えられている ．図において，エージェントと仕事の高さはそれぞれの能力値，仕事量の大きさを表している ．解の例を図 5 に示す ．ここでは $\sigma(t_1) = \{a_1\}, \sigma(t_2) = \{a_3\}, \sigma(t_3) = \emptyset, \sigma(t_4) = \{a_5\}, \sigma(t_5) = \{a_2, a_4\}$ と割り当てている ．すなわち， t_3 以外の仕事がすべて処理されており，利益の総和は $\bar{t}_1 + \bar{t}_2 + \bar{t}_4 + \bar{t}_5$ となる ．

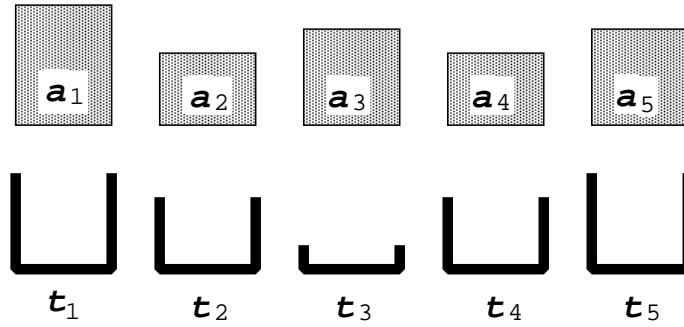


図 4 1-TCAP の問題例

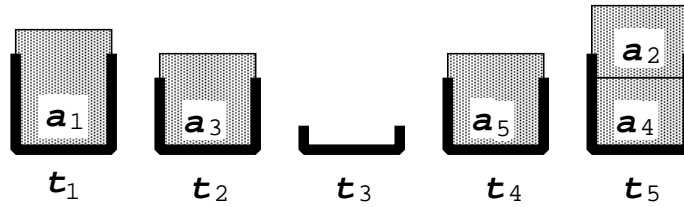


図 5 1-TCAP の解の例

2.5 1-TCAP の NP 困難性

本節では 1-TCAP が NP 困難であることを証明する．そのために，すでに NP 困難であることが知られている subset sum 問題 [7] が 1-TCAP に帰着できることを示す．

subset sum 問題とは，実数有限集合 Z と，実数 b が与えられた場合に， Z 内の数をうまく足し合わせてちょうど b になるような組合せが存在するかどうかを判定する問題である．

[定義 3] subset sum 問題は以下のように定義される．まず，入力は以下の 2 つである．

入力:

- $Z = \{\bar{z}_1, \bar{z}_2, \dots, \bar{z}_m\} \subset \mathbf{R}$.
- $b \in \mathbf{R}$.

目的は以下のような $C \subseteq Z$ が存在するかどうか判定することである .

$$b = \sum_{z \in C} \bar{z}$$

[定理 2] 1-TCAP は NP 困難である .

証明 : subset sum 問題が 1-TCAP に帰着できることを示すことにより , 本定理を証明する .

subset sum 問題のインスタンスが以下のように与えられたとする .

- $Z = \{\bar{z}_1, \bar{z}_2, \dots, \bar{z}_m\} \subset \mathbf{R}$.
- $b \in \mathbf{R}$.

これに対し , 1-TCAP のインスタンスを以下のように構成する .

- エージェント集合 $A = \{a_1, a_2, \dots, a_m\}$.
- 仕事集合 $T = \{t, t'\}$.
- 各エージェント $a_i \in A$ の能力値 $\bar{a}_i = \bar{z}_i$.
- 仕事 t の能力値 $\bar{t} = b$, 仕事 t' の能力値 $\bar{t}' = (\sum_{a \in A} \bar{a}) - b$.

この場合 ,

$$b = \sum_{a \in C} \bar{a}, C \subseteq A$$

なる C が存在するときかつそのときのみ利益の総和の最大値は $\bar{t} + \bar{t}'$ となる . よって subset sum 問題は 1-TCAP に帰着できる .

入力 :

- エージェントの有限集合 A
- 仕事の有限集合 T
- 能力値と仕事量を与える関数 :

出力 :

- 割り当て σ

1. A と T をそれぞれ能力値, 仕事量に従って降順に並べ替える .
並べ換えた結果をそれぞれ $a_1, a_2, \dots, a_m, t_1, t_2, \dots, t_n$ とする .
2. $j := 1$
3. for $i := 1$ to n do
 - 3.(a) $\bar{a}_j + \bar{a}_{j+1} + \dots + \bar{a}_k \geq \bar{t}_i$ となる最小の k を求める .
 - 3.(a) でそのような k が存在する場合
 $\sigma(t_i) := \{a_j, a_{j+1}, \dots, a_k\}$
 $j := k + 1$
 - 3.(a) でそのような k が存在しない場合
(すなわち $\bar{a}_j + \bar{a}_{j+1} + \dots + \bar{a}_m < \bar{t}_i$ の場合)
 $\sigma(t_i) := \emptyset$

図 6 降順連続当てはめアルゴリズム (SONFA)

2.6 1-TCAP に対する近似アルゴリズム

1-TCAP は定理 2 より NP 困難であるため, よい多項式時間近似アルゴリズムを開発することが重要である. 本章では, 1-TCAP に対する近似アルゴリズムとして降順連続当てはめアルゴリズム (Size Order Next Fit Algorithm, 以下 SONFA) を提案する. そのアルゴリズムを図 6 に示す.

図 7 に動作例を示す. この例では, $i = 1$ で $\sigma(t_1) = \{a_1, a_2\}$ と割り当て (SONFA の行 3.(a) で $k = 2$), $i = 2$ で t_2 には割り当てず (k が存在しない), $i = 3$ で $\sigma(t_3) = \{a_3, a_4\}$ ($k = 4$ で最後のエージェントまで割り当て終わった) と割り当てて

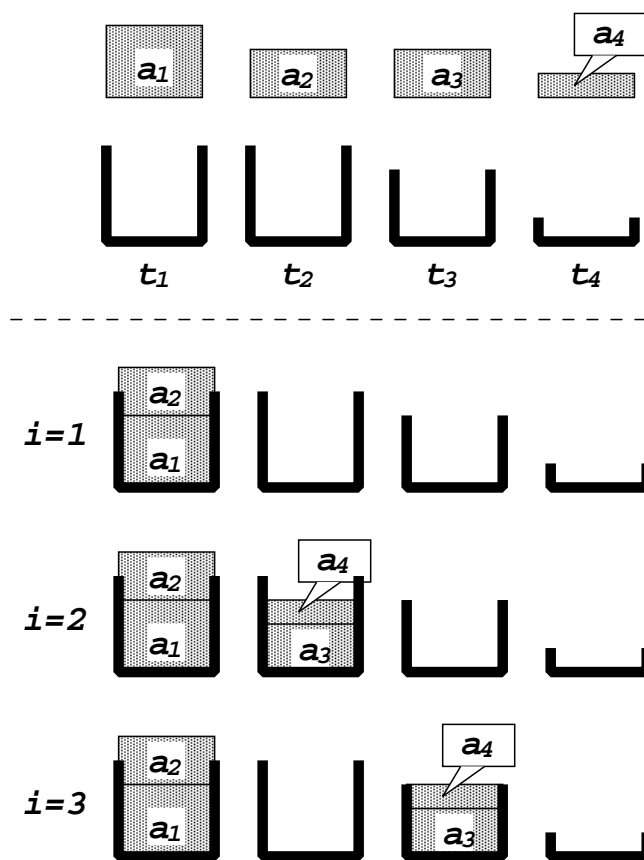


図 7 SONFA の動作例

いる .

これから , 我々は SONFA の性能を分析する . まず , 我々はいかなる $\epsilon < 9/4$ に対しても , SONFA が近似比 ϵ を達成できないことを示す .

[定理 3] いかなる $\epsilon < 9/4$ に対しても , SONFA が近似比 ϵ を達成できない .

証明: $\epsilon < 9/4$ としよう . ここで $\frac{9\Omega-3}{4\Omega+1} > \epsilon$ (すなわち $\Omega > \frac{3+\epsilon}{9-4\epsilon}$) であるような整数 Ω が存在したとする . 図 8 に示されるような 1-TCAP のインスタンスを考える . SONFA の解の利益の合計は $4\Omega + 1$ である (図 9 を見よ) . そして , 最適解の利益の合計は $9\Omega - 3$ である (図 10 を見よ) . Ω の定義により , SONFA は近似比 ϵ を達成しない .

$$\begin{aligned}
A &= \{a_1, a_2, a_3, a_4, a_5, a_6\}. \\
T &= \{t_1, t_2, t_3, t_4\}. \\
\bar{\cdot} &: \bar{a}_1 = \bar{a}_2 = \bar{a}_3 = 2\Omega, \\
&\bar{a}_4 = \bar{a}_5 = \bar{a}_6 = \Omega - 1, \\
&\bar{t}_1 = 4\Omega + 1, \\
&\bar{t}_2 = \bar{t}_3 = \bar{t}_4 = 3\Omega - 1.
\end{aligned}$$

図 8 SONFA にとって最悪の場合のインスタンス

次に、我々は SONFA が近似比 $9/4$ を達成することを示す。我々は図 11 に示す表記法を用いる。

可読性のために、 $\delta^S(T, A)$ と $\delta^*(T, A)$ はしばしば σ^S と σ^* とそれぞれ記述される。我々は σ^* が“意味のない割り当て”を含まないと仮定する。この仮定を用いても、我々の証明は一般性を失わない。“意味のない割り当て”とは、 $t \in T$, if $|\sigma^*(t)| > 0$ に対して $\bar{\sigma}^*(t) < \bar{t}$ であるような割り当てである。

T_{empty} を $|\sigma^S(t)| = 0$ であるような仕事 t の集合であるとする。 k を $\sigma^S(t_i) \neq \emptyset$ であるようなすべての i ($1 \leq i \leq k$) の中で最大の添字であるとする。この場合、 T_{empty} 内の仕事の最小の添字は $k+1$ である。ここで、 l を $\bigcup_{i=1}^k \sigma^S(t_i)$ 内のエージェントの中で最大の添字であるとする (図 12)。また、 $U(\delta^*(T, A)) = \{t \mid a_i \in \sigma^*(t) \text{ for some } i (1 \leq i \leq l)\}$, $T' = \{t_{k+2}, \dots, t_n\}$, $A' = \{a_{l+1}, \dots, a_m\}$ であるとする。

[補題 1] 以下の式が成り立つ:

$$v(\delta^*(T, A)) \leq \bar{U}(\delta^*(T, A)) + v(\delta^*(T', A')). \quad (1)$$

証明: $\bar{A}' < \bar{t}_{k+1}$ であるから、 A' は t_i ($i \leq k+1$) であるようないかなる仕事をも処理することができない。従って、

$$\begin{aligned}
v(\delta^*(T, A)) &\leq \bar{U}(\delta^*(T, A)) + v(\delta^*(T, A')) \\
&= \bar{U}(\delta^*(T, A)) + v(\delta^*(T', A')).
\end{aligned} \quad (2)$$

である。□

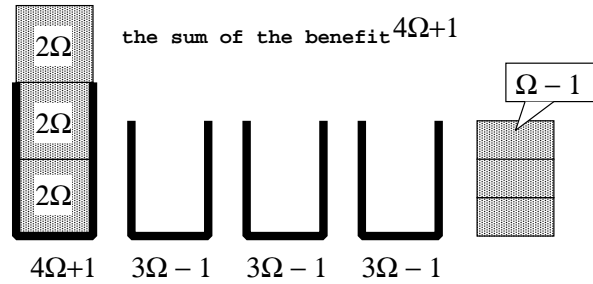


図 9 最悪の場合の SONFA の解

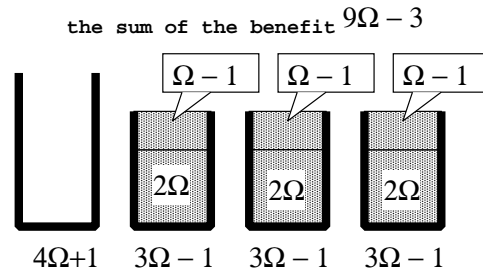


図 10 最悪の場合の最適解

[補題 2]

$$\frac{\bar{U}(\delta^*(T, A))}{v(\delta^S(T - T', A - A'))} < \frac{9}{4}. \quad (3)$$

であるとする . この場合 ,

$$\frac{v(\delta^*(T', A'))}{v(\delta^S(T', A'))} < \frac{9}{4} \quad (4)$$

もしくは

$$v(\delta^*(T', A')) = v(\delta^S(T', A')) = 0. \quad (5)$$

であるならば ,

$$\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} < \frac{9}{4}. \quad (6)$$

である .

- \bar{T} : $\sum_{t \in T} \bar{t}$.
 \bar{A} : $\sum_{a \in A} \bar{a}$.
 m : インスタンスにおける仕事数 .
 n : インスタンスにおけるエージェント数 .
 $\bar{\sigma}(t)$: $\sum_{a \in \sigma(t)} \bar{a}$.
 $\delta^S(T, A)$: インスタンス (T, A) に対する SONFA の出力の割り当て .
 $\delta^*(T, A)$: インスタンス (T, A) に対する最適解の割り当て .
 $v(\sigma)$: 割り当て σ の利益の合計 .

図 11 2 章 4 節における表記法

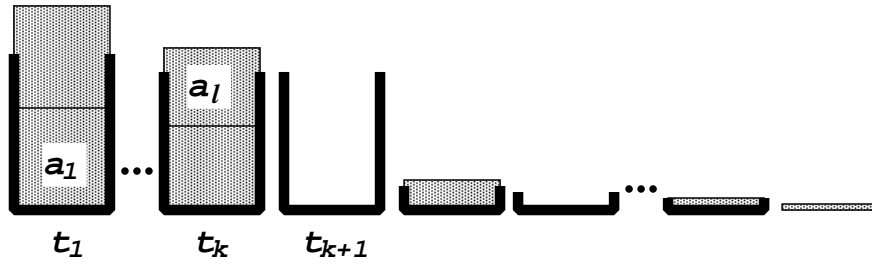


図 12 エージェント a_l と仕事 t_k

証明: SONFA の振舞いにより ,

$$\begin{aligned}
 v(\delta^S(T, A)) \\
 &= v(\delta^S(T - T', A - A')) + v(\delta^S(T', A')).
 \end{aligned} \tag{7}$$

従って , 補題 1 を用いて ,

$$\begin{aligned}
 &\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} \\
 &\leq \frac{\bar{U}(\delta^*(T, A)) + v(\delta^*(T', A'))}{v(\delta^S(T - T', A - A')) + v(\delta^S(T', A'))} \\
 &< \frac{9}{4}.
 \end{aligned} \tag{8}$$

である .

□

[定理 4] SONFA は , 近似比 $9/4$ を達成する . つまり , いかなるインスタンス (T, A) に対しても

$$\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} < \frac{9}{4} \quad (9)$$

であるか , さもなくば

$$v(\delta^*(T, A)) = v(\delta^S(T, A)) = 0 \quad (10)$$

である .

証明: 補題 2 に基づいて , 我々は $|T_{\text{empty}}|$ に関する帰納法によりこの定理を証明する .

基底段階として , 我々は式 (9) もしくは式 (10) が成り立つことを示す . 最初に , すべての i ($1 \leq i \leq n$) に対して $n \geq 1$ かつ $|\sigma^S(t_i)| \geq 1$ であるような場合を考える . この場合においては , すべての仕事が処理される . 従って ,

$$\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} = 1 < \frac{9}{4}. \quad (11)$$

一方 , もし $T = \emptyset$ であれば , 全く仕事は処理されない . 従って ,

$$v(\delta^*(T, A)) = v(\delta^S(T, A)) = 0. \quad (12)$$

である . 帰納段階として , 我々は表 1 が成り立つことを示す .

表 1 定理 4 の証明の帰納段階

	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k \geq 4$
$l = 1, 2$	補題 4	補題 5	補題 5	補題 5	補題 3
$l = 3, 4$	補題 4	補題 6	補題 5	補題 5	補題 3
$l = 5, 6$	補題 4	補題 6	補題 7	補題 5	補題 3
$l \geq 7$	補題 4	補題 6	補題 7	補題 8	補題 3

[補題 3] $k \geq 4$ であれば ,

$$\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} < \frac{9}{4}. \quad (13)$$

である .

証明: 1-TCAP の定義により ,

$$v(\delta^*(T, A)) \leq \sum_{j=1}^m \bar{a}_j = \sum_{j=1}^l \bar{a}_j + \bar{A}'. \quad (14)$$

である . SONFA の振舞いと k の定義により ,

$$v(\delta^S(T, A)) \geq \sum_{i=1}^k \bar{t}_i, \quad (15)$$

$$\bar{A}' < \bar{t}_{k+1}. \quad (16)$$

である . $\bar{t}_1 \geq \bar{t}_2 \geq \dots \geq \bar{t}_n$ であるから ,

$$\frac{\bar{t}_{k+1}}{\sum_{i=1}^k \bar{t}_i} \leq \frac{1}{k}. \quad (17)$$

である .

最初に , すべての i ($1 \leq i \leq k$) に対して $\bar{\sigma}^S(t_i) < 2\bar{t}_i$ であるような場合を考える . この場合では ,

$$\sum_{j=1}^l \bar{a}_j < \sum_{i=1}^k 2\bar{t}_i = 2 \sum_{i=1}^k \bar{t}_i. \quad (18)$$

である . 従って , 式 (14)–(18) により ,

$$\begin{aligned} \frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &\leq \frac{\sum_{j=1}^l \bar{a}_j + \bar{A}'}{\sum_{i=1}^k \bar{t}_i} \\ &< \frac{2 \sum_{i=1}^k \bar{t}_i + \bar{t}_{k+1}}{\sum_{i=1}^k \bar{t}_i} \\ &= 2 + \frac{\bar{t}_{k+1}}{\sum_{i=1}^k \bar{t}_i} \\ &\leq 2 + \frac{1}{k}. \end{aligned} \quad (19)$$

である．ここで， $k \geq 4$ の場合，式 (13) が成り立つ．

次に， $\bar{\sigma}^S(t_i) \geq 2\bar{t}_i$ であるような i ($1 \leq i \leq k$) が存在する場合を考える． o を $\bar{\sigma}^S(t_i) \geq 2\bar{t}_i$ であるような最大の添字 i ($1 \leq i \leq k$) であるとする．また，我々は以下の表記法を用いる：

$$T_1 = \{t_1, \dots, t_o\}, A_1 = \bigcup_{t \in T_1} \sigma^S(t),$$

$$T_2 = \{t_{o+1}, \dots, t_k\}, A_2 = \bigcup_{t \in T_2} \sigma^S(t).$$

インスタンス $(T - T_1, A)$ の最適解の利益は $v(\delta^*(T, A)) - \bar{T}_1$ によって上界が与えられることに注意せよ．そこで，

$$v(\delta^*(T, A)) \leq \bar{T}_1 + v(\delta^*(T - T_1, A)). \quad (20)$$

である． A_1 内の各エージェントは $T - T_1$ 内の各仕事をそれぞれ単独で処理するから，

$$\begin{aligned} v(\delta^*(T - T_1, A)) &\leq v(\delta^*(T - T_1, A_1)) \\ &\quad + v(\delta^*(T - T_1, A_2 \cup A')). \end{aligned} \quad (21)$$

である．まず，式 (21) 右辺の第一項， $v(\delta^*(T - T_1, A_1))$ を考える．我々は以下の式を得る．

$$v(\delta^*(T - T_1, A_1)) \leq |A_1| \bar{t}_o, \quad (22)$$

なぜならば，

$$\bar{a} > \max_{t \in T - T_1} \bar{t} \quad (\forall a \in A_1), \quad (23)$$

$$\bar{t}_o \geq \max_{t \in T - T_1} \bar{t}. \quad (24)$$

我々は各 t_i ($i \leq o$) に対して以下の式を示す．

$$\bar{t}_i \geq |\sigma^S(t_i)| \bar{t}_o, \quad (25)$$

$\delta^S(T, A)$ において T_1 内の仕事は処理されているから， $|\sigma^S(t_i)| \geq 1$ が成り立つ．もし $|\sigma^S(t_i)| = 1$ であれば，明らかに式 (25) が成り立つ． $|\sigma^S(t_i)| \geq 2$ の場合，SONFA の振舞いにより，

$$\begin{aligned} \bar{t}_i &> \bar{\sigma}^S(t_i) - (\min_{a \in \sigma^S(t_i)} \bar{a}) \\ &\geq (|\sigma^S(t_i)| - 1)(\min_{a \in \sigma^S(t_i)} \bar{a}). \end{aligned} \quad (26)$$

である． $\bar{\sigma}^S(t_i)$ の定義により，

$$\bar{\sigma}^S(t_i) \geq |\sigma^S(t_i)|(\min_{a \in \sigma^S(t_i)} \bar{a}). \quad (27)$$

である．式 (26) と (27) により，

$$\begin{aligned} |\sigma^S(t_i)|\bar{t}_i &> |\sigma^S(t_i)|\bar{\sigma}^S(t_i) - |\sigma^S(t_i)|(\min_{a \in \sigma^S(t_i)} \bar{a}) \\ &\geq |\sigma^S(t_i)|\bar{\sigma}^S(t_i) - \bar{\sigma}^S(t_i) \\ &= (|\sigma^S(t_i)| - 1)\bar{\sigma}^S(t_i), \end{aligned}$$

$$\frac{|\sigma^S(t_i)|}{|\sigma^S(t_i)| - 1} \bar{t}_i > \bar{\sigma}^S(t_i). \quad (28)$$

である．もし $|\sigma^S(t_o)| \geq 2$ であれば，式 (28) は仮定 $\bar{\sigma}^S(t_o) \geq 2\bar{t}_o$ に矛盾する．そこで， $|\sigma^S(t_o)| = 1$ であり，従って $\min_{a \in \sigma^S(t_o)} \bar{a} = \bar{\sigma}^S(t_o)$ である． A_1 と t_o の定義により， $\min_{a \in A_1} \bar{a} = \min_{a \in \sigma^S(t_o)} \bar{a}$ である．従って，

$$\begin{aligned} \min_{a \in \sigma^S(t_i)} \bar{a} &\geq \min_{a \in A_1} \bar{a} \\ &= \min_{a \in \sigma^S(t_o)} \bar{a} = \bar{\sigma}^S(t_o) \\ &\geq 2\bar{t}_o. \end{aligned} \quad (29)$$

である．式 (26) と (29)，そして仮定 $|\sigma^S(t_i)| \geq 2$ により，

$$\begin{aligned} \bar{t}_i &> (|\sigma^S(t_i)| - 1)(\min_{a \in \sigma^S(t_i)} \bar{a}) \\ &\geq (|\sigma^S(t_i)| - 1)2\bar{t}_o \\ &\geq |\sigma^S(t_i)|\bar{t}_o. \end{aligned} \quad (30)$$

である．従って，式 (25) が成り立つ．式 (25) により，

$$\bar{T}_1 = \sum_{i=1}^o \bar{t}_i \geq \sum_{i=1}^o |\sigma^S(t_i)|\bar{t}_o = |A_1|\bar{t}_o. \quad (31)$$

である．従って，

$$v(\delta^*(T - T_1, A_1)) \leq |A_1|\bar{t}_o \leq \bar{T}_1. \quad (32)$$

である .

次に , 式 (21) 右辺の第二項 , $v(\delta^*(T - T_1, A_2 \cup A'))$ についてを考える . 明らかに , これは $\bar{A}_2 + \bar{A}'$ より大きくない . $\bar{A}_2 < 2\bar{T}_2$ と $\bar{A}' < \bar{t}_{k+1}$ から ,

$$v(\delta^*(T - T_1, A_2 \cup A')) < 2\bar{T}_2 + \bar{t}_{k+1}. \quad (33)$$

である . 式 (20), (21), (32) と (33) によって ,

$$\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} < \frac{\bar{T}_1 + (\bar{T}_1 + 2\bar{T}_2 + \bar{t}_{k+1})}{\sum_{i=1}^k \bar{t}_i}. \quad (34)$$

である . 式 (14) と $\bar{T}_1 + \bar{T}_2 = \sum_{i=1}^k \bar{t}_i$ によって ,

$$\begin{aligned} \frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &< \frac{2 \sum_{i=1}^k \bar{t}_i + \bar{t}_{k+1}}{\sum_{i=1}^k \bar{t}_i} = 2 + \frac{\bar{t}_{k+1}}{\sum_{i=1}^k \bar{t}_i} \\ &\leq 2 + \frac{1}{k}. \end{aligned} \quad (35)$$

である . 従って , $k \geq 4$ の場合 , 式 (13) が成り立つ . □

[補題 4] 式 (4) もしくは式 (5) が成り立つとする . もし , $k = 0$ であれば , 式 (9) もしくは式 (10) が成り立つ .

証明: $k = 0$ であるから , 我々は $\bar{A} < \bar{t}_1$ を得る . 従って , $\sigma^S(t_1) = \sigma^*(t_1) = \emptyset$ である . このことから

$$v(\delta^*(T, A)) = v(\delta^*(T', A)), \quad (36)$$

$$v(\delta^S(T, A)) = v(\delta^S(T', A)). \quad (37)$$

従って , この補題は証明される .

[補題 5] 式 (4) もしくは式 (5) が成り立つとする . もし $1 \leq k \leq 3$ かつ $l < (9/4)k$ であれば ,

$$\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} < \frac{9}{4}. \quad (38)$$

である .

証明: U の定義により ,

$$|U(\delta^*(T, A))| \leq l \leq \left\lfloor \frac{9}{4}k \right\rfloor. \quad (39)$$

である . 従って ,

$$\begin{aligned} \frac{\bar{U}(\delta^*(T, A))}{v(\delta^S(T - T', A - A'))} &\leq \frac{\bar{t}_1 + \cdots + \bar{t}_{\lfloor \frac{9}{4}k \rfloor}}{\bar{t}_1 + \cdots + \bar{t}_k} \\ &< \frac{9}{4}. \end{aligned} \quad (40)$$

である . 補題 2 により , この補題は証明される .

[補題 6] 式 (4) もしくは式 (5) が成り立つとする . もし $k = 1$ かつ $l \geq 3$ であれば ,

$$\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} < \frac{9}{4}. \quad (41)$$

である .

証明: 最初に , $\bar{A}' < (\frac{9}{4} - \frac{l}{l-1})\bar{t}_1 + v(\delta^S(T', A'))$ であるとする . $v(\delta^*(T, A)) \leq \bar{A} < \frac{l}{l-1}\bar{t}_1 + \bar{A}'$ であるから ,

$$\begin{aligned} \frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &\leq \frac{\frac{l}{l-1}\bar{t}_1 + \bar{A}'}{\bar{t}_1 + v(\delta^S(T', A'))} \\ &< \frac{\frac{9}{4}\bar{t}_1 + v(\delta^S(T', A'))}{\bar{t}_1 + v(\delta^S(T', A'))} \\ &\leq \frac{9}{4}. \end{aligned} \quad (42)$$

である . 次に ,

$$\bar{A}' \geq (\frac{9}{4} - \frac{l}{l-1})\bar{t}_1 + v(\delta^S(T', A')). \quad (43)$$

である場合を考える . 補題 2 により , この場合 ,

$$\frac{\bar{U}(\delta^*(T, A))}{v(\delta^S(T - T', A - A'))} < \frac{9}{4}. \quad (44)$$

の式が成り立つことを示してこの補題を証明する．そのために，

$$\frac{\bar{U}(\delta^*(T, A))}{v(\delta^S(T - T', A - A'))} \geq \frac{9}{4}, \quad (45)$$

であると仮定して矛盾を導く．仮定 (45) から， $|U(\delta^*(T, A))| \geq 3$ である．なぜならば， $\delta^S(T - T', A - A')(t_1) \neq \emptyset$ であるからである． $t \in \{t_3, \dots, t_n\}$ を $U(\delta^*(T, A))$ において三番目に大きな仕事量を持つ仕事であるとする．すると，

$$\bar{t} \leq \frac{\bar{U}(\delta^*(T, A))}{3}. \quad (46)$$

である．

$$\bar{U}(\delta^*(T, A)) \leq \bar{A} < \frac{l}{l-1}\bar{t}_1 + \bar{A}' \quad (47)$$

と仮定 (43) を用いて，

$$\begin{aligned} \bar{t} &< \frac{l}{3(l-1)}\bar{t}_1 + \frac{\bar{A}'}{3} \\ &\leq \frac{l}{3(l-1)} \cdot \frac{\bar{A}' - v(\delta^S(T', A'))}{\frac{9}{4} - \frac{l}{l-1}} + \frac{\bar{A}'}{3} \\ &= \frac{l(\bar{A}' - v(\delta^S(T', A')))}{\frac{15l}{4} - \frac{27}{4}} + \frac{\bar{A}'}{3} \\ &= \frac{9(l-1)\bar{A}' - 4lv(\delta^S(T', A'))}{15l - 27} \\ &= \frac{9(l-1)}{15l - 27}\bar{A}' - \frac{4lv(\delta^S(T', A'))}{15l - 27}. \end{aligned} \quad (48)$$

を導くことができる． $l \geq 3$ の場合，式 $\frac{9(l-1)}{15l-27} \leq 1$ と $\frac{4lv(\delta^S(T', A'))}{15l-27} \geq 0$ が成り立つ．ゆえに， $\bar{t} < \bar{A}'$ であり，従って A' は t を処理できる． $t \in T'$ から，

$$\bar{t} \leq v(\delta^S(T', A')). \quad (49)$$

である． U の定義から， $|U(\delta^*(T, A))| \leq l$ で $U(\delta^*(T, A)) \subseteq T$ である．従って，

$$\bar{U}(\delta^*(T, A)) \leq \sum_{i=1}^l \bar{t}_i \leq \bar{t}_1 + \bar{t}_2 + (l-2)\bar{t}_3. \quad (50)$$

である． t は $U(\delta^*(T, A))$ において三番目に仕事量の大きな仕事であるから，

$$\bar{U}(\delta^*(T, A)) \leq \bar{t}_1 + \bar{t}_2 + (l-2)\bar{t}. \quad (51)$$

である．仮定 (45) と式 (51) から，

$$\begin{aligned} \frac{\bar{t}_1 + \bar{t}_2 + (l-2)\bar{t}}{\bar{t}_1} &\geq \frac{\bar{U}(\delta^*(T, A))}{v(\delta^S(T - T', A - A'))} \geq \frac{9}{4}, \\ \bar{t}_1 + \bar{t}_1 + (l-2)\bar{t} &\geq \frac{9}{4}\bar{t}_1, \\ \bar{t} &\geq \frac{1}{4(l-2)}\bar{t}_1. \end{aligned} \quad (52)$$

である．一方，仮定 (43) から，

$$\begin{aligned} \left(\frac{9}{4} - \frac{l}{l-1}\right)\bar{t}_1 + v(\delta^S(T', A')) &\leq \bar{A}' \\ &< \bar{t}_2 \\ &\leq \bar{t}_1, \end{aligned} \quad (53)$$

であり，従って，

$$v(\delta^S(T', A')) < \left(\frac{l}{l-1} - \frac{5}{4}\right)\bar{t}_1. \quad (54)$$

である．式 (49), (52) と (54) から，

$$\begin{aligned} \frac{1}{4(l-2)}\bar{t}_1 &\leq \bar{t} \\ &\leq v(\delta^S(T', A')) \\ &< \left(\frac{l}{l-1} - \frac{5}{4}\right)\bar{t}_1. \end{aligned} \quad (55)$$

である．しかしこの式は矛盾している．なぜならば $l \geq 3$ の場合に

$$\begin{aligned} &\left(\frac{1}{4(l-2)}\right) - \left(\frac{l}{l-1} - \frac{5}{4}\right) \\ &= \frac{1}{4} \left(\frac{1}{l-2} - \frac{4l}{l-1} + 5\right) \\ &= \frac{1}{4} \frac{l-1-4l^2+8l+5l^2-15l+10}{(l-1)(l-2)} \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{4} \frac{l^2 - 6l + 9}{(l-1)(l-2)} \\
&= \frac{1}{4} \frac{(l-3)^2}{(l-1)(l-2)} \geq 0,
\end{aligned} \tag{56}$$

となるためである .

以上のことから , 式 (43) が成り立つ場合は , 式 (44) が成り立つ . よってこの補題は証明される .

[補題 7] 式 (4) もしくは式 (5) が成り立つ . もし $k = 2$ かつ $l \geq 5$ であれば ,

$$\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} < \frac{9}{4}. \tag{57}$$

である .

証明: $l_1 = |\sigma^S(t_1)|$ かつ $l_2 = |\sigma^S(t_2)|$. であるとする . $l_1 + l_2 = l$ であることに注意せよ . 我々は以下の 3 つの場合について考える .

- $l_1 = 1$ の場合 . この場合には ,

$$\begin{aligned}
\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &< \frac{\bar{t}_1 + \frac{l_2}{l_2-1}\bar{t}_2 + \bar{A}'}{\bar{t}_1 + \bar{t}_2} \\
&< \frac{\bar{t}_1 + \frac{l_2}{l_2-1}\bar{t}_2 + \bar{t}_1}{\bar{t}_1 + \bar{t}_2} \\
&< \frac{9}{4}.
\end{aligned} \tag{58}$$

である .

- $l_2 = 1$ の場合 . $\bar{t}_1 > \bar{a}_1 + \dots + \bar{a}_{l_1-1}$ かつ $\bar{a}_l \geq \bar{t}_2$ であるから , 我々は式 $\bar{t}_1 > (l_1 - 1)\bar{t}_2$ を得る . $\bar{U}(\delta^*(T, A)) \leq \bar{t}_1 + (l - 1)\bar{t}_2$ であるから ,

$$\begin{aligned}
\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &\leq \frac{\bar{t}_1 + (l - 1)\bar{t}_2 + \bar{A}'}{\bar{t}_1 + \bar{t}_2} \\
&< \frac{2\bar{t}_1 + \bar{t}_2 + \bar{A}'}{\bar{t}_1 + \bar{t}_2} \\
&< \frac{9}{4}.
\end{aligned} \tag{59}$$

である .

• $l_1, l_2 \geq 2$ の場合 . 我々は以下の式を仮定する .

$$\begin{aligned} \bar{A}' &\geq \left(\frac{9}{4} - \frac{l_1}{l_1 - 1}\right) \bar{t}_1 + \left(\frac{9}{4} - \frac{l_2}{l_2 - 1}\right) \bar{t}_2 \\ &\quad + v(\delta^S(T', A')). \end{aligned} \quad (60)$$

$\bar{A}' < \bar{t}_2$ を用いて ,

$$\begin{aligned} v(\delta^S(T', A')) &< \left(\frac{l_1}{l_1 - 1} - \frac{9}{4}\right) \bar{t}_1 + \left(\frac{l_2}{l_2 - 1} - \frac{9}{4} + 1\right) \bar{t}_2 \\ &= \left(\frac{1}{l_1 - 1} - \frac{5}{4}\right) \bar{t}_1 + \left(\frac{1}{l_2 - 1} - \frac{1}{4}\right) \bar{t}_2 \\ &\leq \left(\frac{1}{l_1 - 1} + \frac{1}{l_2 - 1} - \frac{3}{2}\right) \bar{t}_1 \leq 0. \end{aligned} \quad (61)$$

である . これは矛盾している . だから ,

$$\begin{aligned} \bar{A}' &< \left(\frac{9}{4} - \frac{l_1}{l_1 - 1}\right) \bar{t}_1 + \left(\frac{9}{4} - \frac{l_2}{l_2 - 1}\right) \bar{t}_2 \\ &\quad + v(\delta^S(T', A')). \end{aligned} \quad (62)$$

である . 従って ,

$$\begin{aligned} \frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &\leq \frac{\frac{l_1}{l_1 - 1} \bar{t}_1 + \frac{l_2}{l_2 - 1} \bar{t}_2 + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + v(\delta^S(T', A'))} \\ &< \frac{\frac{9}{4} \bar{t}_1 + \frac{9}{4} \bar{t}_2 + v(\delta^S(T', A'))}{\bar{t}_1 + \bar{t}_2 + v(\delta^S(T', A'))} \\ &\leq \frac{9}{4}. \end{aligned} \quad (63)$$

である .

□

[補題 8] 式 (4) もしくは式 (5) が成り立つとする . もし $k = 3$ かつ $l \geq 7$ であるならば ,

$$\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} < \frac{9}{4}. \quad (64)$$

である .

証明: $l_1 = |\sigma^S(t_1)|$, $l_2 = |\sigma^S(t_2)|$ かつ $l_3 = |\sigma^S(t_3)|$ であるとする . $l_1 + l_2 + l_3 = l$ であることに注意せよ .

- $l_1 = l_2 = 1$ である場合 . この場合では ,

$$\begin{aligned} \frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &< \frac{\bar{t}_1 + \bar{t}_2 + \frac{l_3}{l_3-1}\bar{t}_3 + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\ &< \frac{9}{4}. \end{aligned} \quad (65)$$

である .

- $l_1 = l_3 = 1$ である場合 . $\bar{t}_2 > (l_2 - 1)\bar{t}_3$ であり $\bar{U}(\delta^*(T, A)) \leq \bar{t}_1 + \bar{t}_2 + (l - 2)\bar{t}_3$ であることから ,

$$\begin{aligned} \frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &\leq \frac{\bar{t}_1 + \bar{t}_2 + (l - 2)\bar{t}_3 + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\ &< \frac{\bar{t}_1 + 2\bar{t}_2 + \bar{t}_3 + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} < \frac{9}{4}. \end{aligned} \quad (66)$$

である .

- $l_1 = 1$ で $l_2, l_3 \geq 2$ である場合 . この場合には ,

$$\begin{aligned} \frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &< \frac{\bar{t}_1 + \frac{l_2}{l_2-1}\bar{t}_2 + \frac{l_3}{l_3-1}\bar{t}_3 + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\ &< \frac{9}{4}. \end{aligned} \quad (67)$$

である .

- $l_2 = l_3 = 1$ である場合 . $\bar{t}_1 > (l_1 - 1)\bar{t}_3$ であり $\bar{U}(\delta^*(T, A)) \leq \bar{t}_1 + \bar{t}_2 + (l - 2)\bar{t}_3$ であることから ,

$$\begin{aligned} \frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &\leq \frac{\bar{t}_1 + \bar{t}_2 + (l - 2)\bar{t}_3 + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\ &< \frac{2\bar{t}_1 + \bar{t}_2 + \bar{t}_3 + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\ &< \frac{9}{4}. \end{aligned} \quad (68)$$

である .

- $l_2 = 1$ かつ $l_1, l_3 \geq 2$ である場合 . $\bar{t}_1 > (l_1 - 1)\bar{t}_2$ から ,

$$\begin{aligned}
\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &< \frac{\bar{t}_1 + (l_1 + 1)\bar{t}_2 + \frac{l_3}{l_3-1}\bar{t}_3 + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\
&< \frac{2\bar{t}_1 + 2\bar{t}_2 + \frac{l_3}{l_3-1}\bar{t}_3 + \bar{t}_3}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\
&= 2 + \frac{\bar{t}_3}{(l_3 - 1)(\bar{t}_1 + \bar{t}_2 + \bar{t}_3)} \\
&< 2 + \frac{\bar{t}_3}{(l_3 - 1)(l_1\bar{t}_2 + \bar{t}_3)} \\
&< \frac{9}{4}.
\end{aligned} \tag{69}$$

である .

- $l_3 = 1$ かつ $l_1, l_2 \geq 2$ である場合 . $\sigma^S(t_3) = \{a_l\}$ であることから , 我々は式 $\bar{a}_l \geq \bar{t}_3 > \bar{A}'$, $\bar{t}_1 > (l_1 - 1)\bar{a}_l$, そして $\bar{t}_2 > (l_2 - 1)\bar{a}_l$ を得る . 従って ,

$$\begin{aligned}
\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} &< \frac{\frac{l_1}{l_1-1}\bar{t}_1 + \frac{l_2}{l_2-1}\bar{t}_2 + \bar{a}_l + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\
&< 1 + \frac{\frac{1}{l_1-1}\bar{t}_1 + \frac{1}{l_2-1}\bar{t}_2 + \bar{a}_l}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\
&< 2 + \frac{\bar{a}_l}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3} \\
&< 2 + \frac{\bar{a}_l}{(l_1 + l_2 - 2)\bar{a}_l + \bar{t}_3} \\
&< \frac{9}{4}.
\end{aligned} \tag{70}$$

である .

- $l_1, l_2, l_3 \geq 2$ の場合 . 我々は以下の式を仮定する .

$$\begin{aligned}
\bar{A}' &\geq \left(\frac{9}{4} - \frac{l_1}{l_1 - 1}\right)\bar{t}_1 + \left(\frac{9}{4} - \frac{l_2}{l_2 - 1}\right)\bar{t}_2 \\
&\quad + \left(\frac{9}{4} - \frac{l_3}{l_3 - 1}\right)\bar{t}_3 + v(\delta^S(T', A')).
\end{aligned} \tag{71}$$

式 $\bar{A}' < \bar{t}_3$ を用いて ,

$$v(\delta^S(T', A'))$$

$$\begin{aligned}
&< \left(\frac{l_1}{l_1-1} - \frac{9}{4} \right) \bar{t}_1 + \left(\frac{l_2}{l_2-1} - \frac{9}{4} \right) \bar{t}_2 \\
&\quad + \left(\frac{l_3}{l_3-1} - \frac{9}{4} + 1 \right) \bar{t}_3 \\
&= \left(\frac{1}{l_1-1} - \frac{5}{4} \right) \bar{t}_1 + \left(\frac{1}{l_2-1} - \frac{5}{4} \right) \bar{t}_2 \\
&\quad + \left(\frac{1}{l_3-1} - \frac{1}{4} \right) \bar{t}_3 \\
&\leq \left(\frac{1}{l_1-1} + \frac{1}{l_2-1} + \frac{1}{l_3-1} - \frac{11}{4} \right) \bar{t}_1 \\
&< 0.
\end{aligned} \tag{72}$$

である．式 (72) は $v(\delta^S(T', A')) \geq 0$ と矛盾している．だから，

$$\begin{aligned}
\bar{A}' &< \left(\frac{9}{4} - \frac{l_1}{l_1-1} \right) \bar{t}_1 + \left(\frac{9}{4} - \frac{l_2}{l_2-1} \right) \bar{t}_2 \\
&\quad + \left(\frac{9}{4} - \frac{l_3}{l_3-1} \right) \bar{t}_3 + v(\delta^S(T', A')).
\end{aligned} \tag{73}$$

である．従って，

$$\begin{aligned}
&\frac{v(\delta^*(T, A))}{v(\delta^S(T, A))} \\
&\leq \frac{\frac{l_1}{l_1-1} \bar{t}_1 + \frac{l_2}{l_2-1} \bar{t}_2 + \frac{l_3}{l_3-1} \bar{t}_3 + \bar{A}'}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3 + v(\delta^S(T', A'))} \\
&< \frac{\frac{9}{4} \bar{t}_1 + \frac{9}{4} \bar{t}_2 + \frac{9}{4} \bar{t}_3 + v(\delta^S(T', A'))}{\bar{t}_1 + \bar{t}_2 + \bar{t}_3 + v(\delta^S(T', A'))} \\
&\leq \frac{9}{4}.
\end{aligned} \tag{74}$$

である．

□

2.7 本章の結び

本章では，扱いやすい TCAP のサブクラスである 1-TCAP を定義した．1-TCAP に対する多項式時間の近似アルゴリズム SONFA を提案し，それが $9/4$ の

近似比を達成することを示し, しかしいかなる $\epsilon < 9/4$ に対しても近似比 ϵ を達成しないことを示した. 今後の課題として, シミュレーションによって 1-TCAP の性能を評価することが挙げられる.

3. エージェント指向自己適応遺伝アルゴリズム

3.1 序文

本章では、エージェント指向の手法により、適応 GA の一つであるメタ GA と島モデル GA の一つである環境分散型並列 GA[26] (いずれも後述) を合わせ、パラメータを同時に適応させつつ探索を行うエージェント指向自己適応遺伝アルゴリズム (Agent oriented Self-Adaptive Genetic Algorithm, 以下 A-SAGA) を提案する。A-SAGA は、適応度の評価方法と遺伝的演算子の定義が与えられれば、各島に与えられるパラメータがどのような組合せでも適用可能であるという特徴を持つ。本手法の有効性を、メタ GA との探索効率の比較実験により評価し、4 つのパラメータが合理的な計算量で同時に適応させられることを示す。

3.2 関連研究

Goldberg らは、個体数や交叉率などのパラメータの最適値を与えるために、one max problem に GA を適用した際に有効に働くパラメータの範囲を理論的に計算し [12][14]、その範囲を示すコントロールマップと呼ばれるグラフを描いている。また、いくつかの選択手法の特性を分析している [13]。しかし、これらの分析結果を利用して具体的なパラメータの値を得るためには、対象となる問題の特性を改めて定式化する必要があり、問題によってはこの定式化が困難な場合がある。この困難と我々の研究の関連性については 3.4.3 節において改めて考察する。

Bäck による適応 GA[1] では、各個体の突然変異率がそれぞれの遺伝子上にコーディングされている。これにより、良い突然変異率を持った個体が生き残ることが期待される。しかし、高い突然変異率を持つ個体は致死個体となる可能性が高く、終盤には低い突然変異率を持つ個体ばかりになってしまう傾向がある。そのため、終盤の探索力が低くなることが報告されている [10]。

Espinoza らによる適応 GA[8] では、個体それぞれが局所探索を行える。また、個体の生殖などの計算量と局所探索の計算量の割合を適応させ、評価回数毎の探索効率の改善がなされている。

Krink らの適応 GA[24] では、格子状空間の座標に応じて交叉率と突然変異率

が定っており，この空間内に配置された個体が，より望ましいパラメータの組合わせに対応する座標を目指して移動する．更に，同じ格子の中に同時に存在できる個体数が制限されているため，個体の座標がある程度収束した後も交叉率と突然変異率の多様性が確保される．

メタ GA は，GA のパラメータの組合わせ（以後，パラメータベクトル）を GA によって探索する手法である．しかし，メタ GA では，通常の GA の実行，パラメータベクトルの吟味，再度の通常の GA の実行という過程を繰返すため，評価回数が極端に大きくなる傾向があり，全体の計算量に多大な影響を及ぼす．例えば，100 個体 100 世代の通常 GA は 100×100 の評価回数を必要とするが，メタ GA は，通常 GA の 1 度の実行で 1 回分の評価を行うので，10 個体 20 世代のメタ GA を実行すると， $10 \times 20 \times 100 \times 100 = 2,000,000$ の評価回数を必要とする．

Kee らによるメタ GA の改良法 [20] では，実際の問題に対して探索を行う前に，トレーニングのための事前探索を行う．事前探索において，まず，個体群の状態をいくつかの指標により分類する．次に，各個体群に対し，数十種類のパラメータベクトルに関して探索効率を調査する．これにより，個体群の状態に応じてどのパラメータの組合わせの探索効率が高いかを調べる．実際の問題の探索では，個体群の状態に応じて，トレーニングにおいて最も探索効率が高かったパラメータベクトルを使用する．

並列 GA の一つとして島モデル GA (Island GA, 以下 IGA) [4] がある．IGA は，並列に実行させたいいくつかの GA を島とみなし，一部の個体を別の島に移民することにより探索過程の情報を共有し，共同探索を行う．三木らの環境分散型並列 GA [26] では，IGA の各島に対してそれぞれ異なったパラメータベクトルを与えることにより，良いパラメータベクトルが与えられた島において良い解を見つけることが期待できる．ただし，パラメータベクトルは利用者が与えるので，これは適応 GA ではない．

3.3 提案手法

3.3.1 従来手法の問題点

従来の適応 GA のほとんどは 1 ないし 2 つのパラメータしか適応させられず、多数のパラメータを適応させる適応 GA であっても、そのほとんどが大きな計算量を必要としていた。また、適応 GA の最大の目的は、パラメータ適応の労力を軽減することである。その観点から、適応 GA を用いるためには、必要な予備知識や事前定義が少なければ少ないほど望ましいと考えられる。一般に、GA を適用するためには、適応度の評価方法と遺伝的演算子の定義を与えることが必要である。いくつかの適応 GA はそれに加えて多様性などに関する定義を必要とし、これらを利用して探索効率を高めている。しかし、これらを定義することは、問題によっては労力を必要とする。

メタ GA は適応度の評価方法と遺伝的演算子の定義を与えるだけで多数のパラメータを同時に適応させることができるが、計算量が大きいという欠点があった。この原因は、1 つのパラメータベクトルを評価するために 1 度 GA を実行しなければならない点にある。評価するパラメータベクトルの数が多いほど、メタ GA は良いパラメータベクトルを見つける可能性が高まる。同じ計算量でパラメータベクトルをより多く評価しようとする、GA 1 回毎の計算量を減らさなければならない。しかし、その計算量を減らすと GA は十分な探索を行うことができず、良い解を導くことが難しい。更に、計算量を減らした GA に適したパラメータベクトルと元の GA に適したパラメータベクトルが一致するとは限らないという問題がある。

3.3.2 提案手法における改良点

我々は、メタ GA と環境分散型並列 GA を組み合わせ、A-SAGA の前身であるアルゴリズムを開発した。以降、このアルゴリズムを A-SAGA β と呼ぶ。単純にメタ GA を IGA 化した場合には、パラメータベクトル 1 つを評価するために GA を 1 度実行する必要がある。A-SAGA β では、IGA における各島を 1 つのエージェントとみなし、それぞれに異なるパラメータベクトルを与えることにより複

数のパラメータベクトルを同時に適応させる．このことにより，システム全体において1度GAを実行する毎に複数のパラメータベクトルを同時に評価することができる．環境分散型並列GAの特徴から，A-SAGA β においてそれぞれのエージェントの計算量が少なくても，システム全体として良い解を導くことが期待できる．区別のために，各個体の持つ適応度のことを個体適応度，エージェントの評価のために与えられる適応度のことをエージェント適応度と呼ぶ．

メタGAでは，個体群の中で最良の個体適応度を持つ個体（以後，エリート個体と呼ぶ）の適応度をパラメータベクトルの評価として与えている．しかし，A-SAGAでは，同じ手法を用いてもうまく働かない．その理由は，各エージェントの持つエリート個体がまわりのエージェントからの移民に依存して与えられ，パラメータベクトルに対する評価が独立に与えられないためである．A-SAGA β では，エージェント適応度はエリート個体の個体適応度の上昇値の累積値で評価する．ただし，エリート個体の個体適応度は，より良い個体適応度を持つ個体が他の島から移民してきた場合にも上昇するが，この上昇はそのエージェント自身の探索効率の高さによるものではないので，移民による個体適応度の上昇はエージェント適応度に反映させない．

A-SAGAでは，各エージェントは，システム全体の利得の直接的な最大化を目的とするのではなく，それぞれの得る利得の最大化を目的とする点に自律性を見出している．また，複数の利己的なエージェントを用いてシステム全体の利得の最大化を試みる点から，A-SAGAはマルチエージェントシステムであるとみなしている．この自律性を生かした将来の拡張性については，3.5節で述べる．

A-SAGA β では，エージェントにパラメータベクトル $\mathbf{v}_\beta = (n, p_m, p_c, l)$ が与えられる．ここで，

n : 個体数

p_m : 突然変異率

p_c : 交叉率

l : 線形スケール係数

である．各エージェントは通常のGAにより探索を行う．エージェント毎に一定の評価回数が与えられるが， n が異なるため，繰返し回数も異なる．

1. 個体群受け取り
2. 個体群評価
3. 個体群選択
4. 個体群交叉
5. 個体群突然変異
6. 個体群評価
7. 累積評価回数が規定値に達していれば個体移民
8. 累積評価回数が規定値に達していなければ 3. へ
9. 個体群引き渡し

図 13 A-SAGA エージェントアルゴリズム

1. エージェント群の初期化
2. 個体群の初期化
3. 次の時代のエージェントへの個体群引き継ぎ
4. エージェント群による探索
5. 最後の時代でなければ 3. へ
6. エージェント群選択
7. エージェント群交叉
8. エージェント群突然変異
9. 繰返し回数が規定値に達していなければ 2. へ

図 14 A-SAGA アルゴリズム

3.3.3 予備実験とそれに対する考察

A-SAGA β の評価実験を行ったが、期待していたほどの性能は得られなかった (3.4.2節)。

探索の序盤において探索効率が高いエージェントを序盤高効率エージェント、探索の終盤において探索効率が高いエージェントを終盤高効率エージェントと呼ぶ。A-SAGA β において、この両者が混在した場合と、環境分散型並列 GA と同様の効果が働き、移民を通じて助け合うことにより全体として良い探索を行うことができる。しかし、より個体適応度が高いエリート個体が移民によって得られた場合には、その個体適応度の上昇値をエージェント適応度に換算しない。従って、序盤高効率エージェントと終盤高効率エージェントが混在した場合、探索の序盤には序盤高効率エージェントだけが、終盤には終盤高効率エージェントだけが個体適応度を得る傾向がある。一般に、GA において探索効率の高いパラメータは探索過程において変化する [20]。また、探索の序盤において世代毎の適応度の上昇値は大きいですが、終盤においてこれは小さくなる。従って、A-SAGA β では、エージェント適応度を比較したとき、序盤高効率エージェントの方が終盤高効率エージェントよりも大きなエージェント適応度を得る傾向がある。

A-SAGA β では、エージェント適応度によってエージェント、及び、それが持つパラメータベクトルを評価している。その結果、低いエージェント適応度しか持たない終盤高効率エージェントは淘汰され、序盤高効率エージェントばかりになってしまう。このことからシステム全体の性能が低下し、期待した性能が得られなかったと考えられる。

3.3.4 エージェント指向自己適応遺伝アルゴリズム

前節の考察を踏まえて、A-SAGA β の探索過程を分割し、それぞれ別のエージェント群により探索するエージェント指向自己適応遺伝アルゴリズム (A-SAGA) を開発した。ここで、探索過程を分割する単位を時代と呼ぶ。これにより、探索過程に応じた適応が可能となる。A-SAGA の各エージェントのアルゴリズムを図 13 に、A-SAGA 全体のアルゴリズムを図 14 に示す。これらのアルゴリズム内におけるループを区別するため、図 13 において項目 3. から 8. までは 1 度繰返すこ

とを 1 エージェント内世代，図 14において項目 3. から 9. までを 1 度繰返すことを 1A-SAGA 世代と呼ぶ．

A-SAGA では，通常のメタ GA のパラメータ以外に以下のパラメータを与える．

E_m : 1A-SAGA 世代毎の評価回数

G_i : 1A-SAGA 世代毎の時代数

N_a : 1 時代毎のエージェント数

A-SAGA の各エージェントに対してパラメータベクトル $\mathbf{v} = (e_p, p_m, p_c, l)$ を与える．

e_p : 1 エージェント内世代毎の評価回数

p_m : 突然変異率

p_c : 交叉率

l : 線形スケールリング係数

各時代には別のエージェント群が配置される．最初の時代のエージェントだけが個体群の初期化を行い，その個体群が次の時代のエージェントに引き継がれる（図 15）．A-SAGA β ではパラメータにより個体数が変化したが，A-SAGA では常に最大の個体数が与えられる．ただし，1 エージェント内世代毎に評価されるのは e_p 個体のみで，各エージェントに与えられる評価回数の合計は $E_m/G_i N_a$ で一定である．

A-SAGA β において， E_m の評価回数と N_a のエージェントを用いて探索する場合，各エージェントにはそれぞれ E_m/N_a の評価回数を与えられる．A-SAGA においてこれをさらに G_i の時代に分割する場合，各時代には N_a のエージェントが配置され，各エージェントにはそれぞれ $E_m/N_a \cdot G_i$ の評価回数を与えられる．

ここでエージェントは自分の配置された時代だけを認識し，次の時代のエージェントに個体群を引き渡す以外には他の時代のエージェントに干渉することはない．また，全体をメタ的に解釈するエージェントは存在せず，すべてのエージェントは同じ能力を持っている．別の時代に配置されたエージェント同士は隔離され，時代毎に別々のメタ GA を用いてパラメータ探索を行う．このことから，序盤高効率エージェントと終盤高効率エージェントが比較されることはない．同時に時代の数と同じだけのメタ GA を動作させることになるが，エージェント適応度を与えるために必要となる評価回数の合計は同じであるために，評価のためにかか

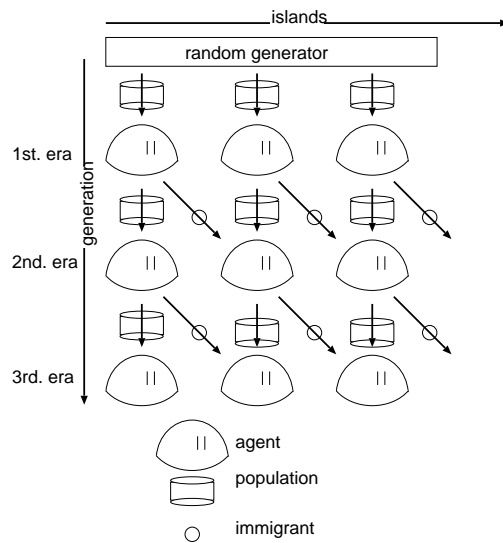


図 15 A-SAGA 概念図

る計算量も同じである。

3.4 実験

3.4.1 実験条件

A-SAGA の探索効率を評価するために、式 (75) に示す Rastrigin 関数と 51 都市巡回セールスマン問題 (Traveling Salesman Problem, 以下 TSP) eil51[44] を用いて実験を行った。いずれも最小化問題である。ただし、10 変数の Rastrigin 関数を用い、各変数を 16 ビットのグレイコードで表現した。TSP において、突然変異は 2 都市の交換を用いた。最適解の個体適応度は、Rastrigin 関数が 0、TSP eil51 が 426 であることが知られている。

$$\begin{aligned}
 f(x_1, x_2, \dots, x_n) &= 10n + \sum_i^n (x_i^2 - 10\cos(2\pi x_i)) \\
 -5.12 < x_i &\leq 5.12
 \end{aligned} \tag{75}$$

探索効率の向上を計測するために、それぞれの A-SAGA 世代で求められたエリート個体の個体適応度を計測した（実験 1）。20A-SAGA 世代まで 2000 試行の実験を行った。なお、定義から $G_i = 1$ のときの A-SAGA は A-SAGA β と同じ性能となる。実験では、以下のパラメータを用いた。

E_m : 20000

G_i : 1, 10, 20

N_a : 10

比較対象として、同じ評価回数を与えたメタ GA に対して同様の計測を行った。筆者らの知る限り、適応度の評価方法と遺伝的演算子の定義を与えるだけで多数のパラメータを同時に適応させることができるという特徴を持つ適応 GA は、メタ GA だけである。

メタ GA は複数のパラメータを同時に適応させることができるが、パラメータを適応させるためのパラメータを手作業で与えなければならない。公正な比較を行うために、メタ GA に与えるパラメータは、あらかじめ手作業により最適化して効率を高めた。このメタ GA を調整メタ GA と呼ぶ。調整メタ GA は 10 回 IGA を実行させる毎にそれらの結果を元に交叉や突然変異を行う。調整メタ GA に与えるパラメータを選ぶために、以下の中から総当たりで調べ、最も探索効率の良かったものの周辺を更に手作業で調査した。

- メタ GA の突然変位率: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1 の 11 段階
- メタ GA の交叉率: 0, 0.2, 0.4, 0.6, 0.8, 1 の 6 段階
- メタ GA の線形スケーリング係数: 1, 2, 3, 4, 5, 6, 7, 10, 20, 50, 100, 200, 500 の 13 段階

A-SAGA は、探索過程を時代で分割することによって、探索過程に応じたパラメータベクトルの適応がなされることを狙っている。これを確かめるため、20A-SAGA 世代までパラメータベクトルを適応させたとき、エージェントが時代毎にどのようなパラメータベクトルを持っているかを計測した（実験 2）。

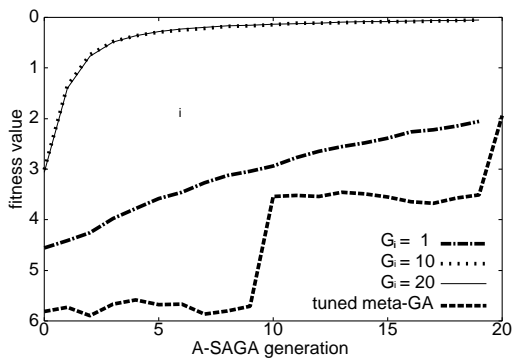


図 16 Rastrigin 個体適応度

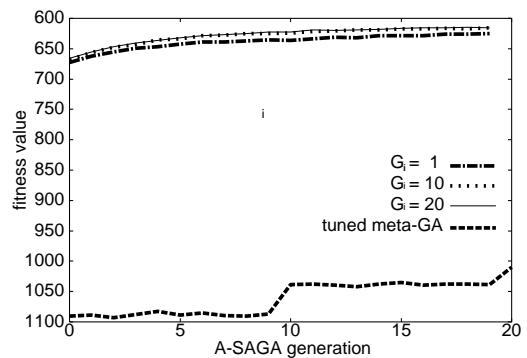


図 17 TSP eil51 個体適応度

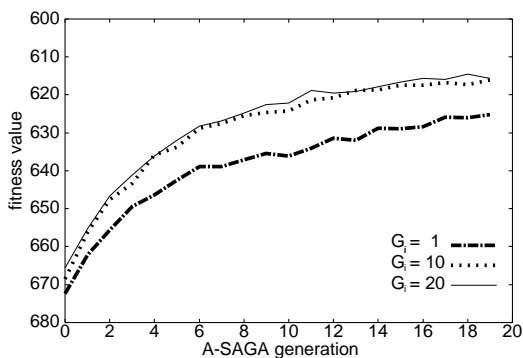


図 18 TSP eil51 個体適応度 (2)

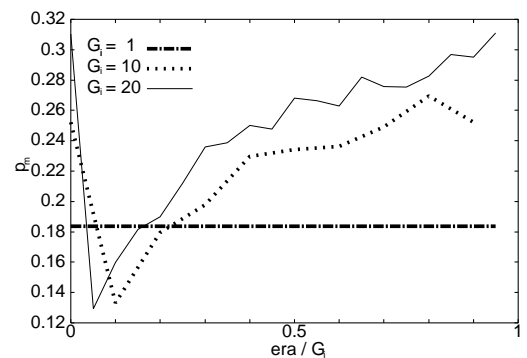


図 19 Rastrigin p_m

3.4.2 実験結果

図 16, 17 はパラメータ適応による探索効率の上昇を示すグラフである (実験 1)。これらのグラフにおいて、 $G_i = 10, 20$ の結果は、ほぼ重なっている。横軸が A-SAGA 世代、縦軸が各 A-SAGA 世代におけるエリート個体の個体適応度である。調整メタ GA の 1 世代は 10A-SAGA 世代に相当する。図 18 は、図 17 における $G_i = 1, 10, 20$ の結果の違いを明確にするために、A-SAGA の結果のみを抜粋したものである。図 19, 20, 21, 22 は、20A-SAGA 世代目において得られた各エージェントのパラメータの平均を示す (実験 2)。横軸は時代の推移である。時代毎に平均をとっているため、 $G_i = 1$ のときは値が一定である。図 19, 21 は p_m 、図 20, 22 は e_p のパラメータの平均を示す。

図 16, 19, 20 は Rastrigin 関数による実験結果であり、図 17, 21, 22 は TSP eil51

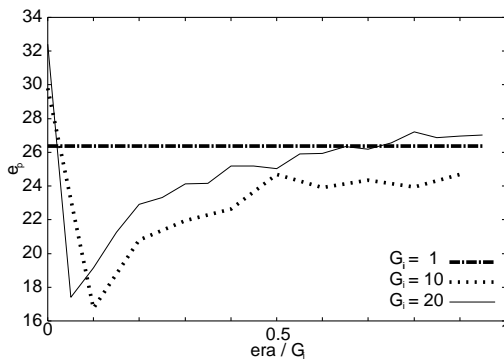


図 20 Rastrigin e_p

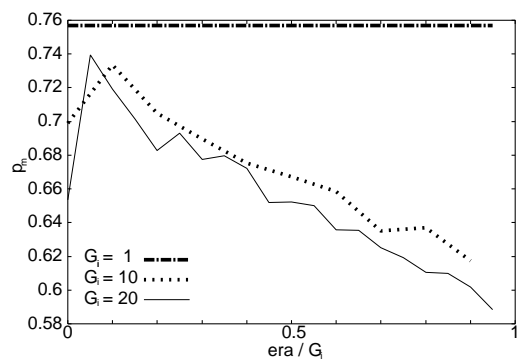


図 21 TSP eil51 p_m

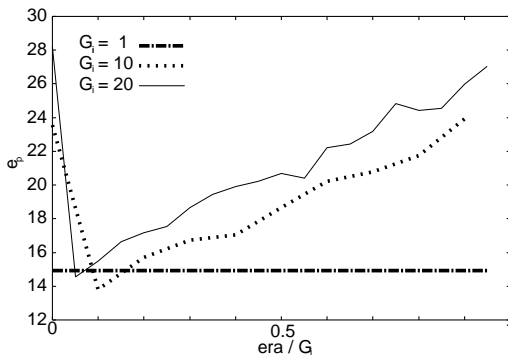


図 22 TSP eil51 e_p

による実験結果である。

3.4.3 考察

- 実験 1 に関する考察

図 16 から、いずれの時代においても A-SAGA は、A-SAGA 世代を経る毎に探索効率が向上することが分かる。特に、 $G_i = 10, 20$ の場合に探索効率が低い(これらの結果はグラフ上で重なっている)。しかし、 $G_i = 1$ の場合、20A-SAGA 世代目で調整メタ GA に逆転されている。このことから、時代で分割することが有効に働いていると考えられる。調整メタ GA に与えたパラメータは、突然変異率が 0.01、交叉率が 1、線形スケール係数が 100 であった。線形スケール係数が高い理由は、調整メタ GA では 2

世代分しかパラメータベクトルの探索が行えないからであると考えられる。ここで、A-SAGA 世代が 0、つまり、全くパラメータベクトルを適応させていない場合においても調整メタ GA より A-SAGA の方が性能が良く、また、A-SAGA においては G_i が大きい方が探索効率が良い。この理由として、最初にランダムな環境が与えられるため、その中でたまたま良いパラメータを持つエージェントが存在する機会は、エージェントの数（ここでは $G_i \times N_a$ ）が多いほうが多く、そのため環境分散型並列 GA と同様の効果が出ていると考えられる。

図 17 から、A-SAGA はいずれの場合においても調整メタ GA に比べて探索効率が低い。最適解の個体適応度 426 には及ばないが、人間がパラメータ調整を行った予備実験の IGA が示した個体適応度 602 に近い値を示している。

図 18 から、 $G_i = 10, 20$ の場合の方が $G_i = 1$ の場合よりも探索効率が低いことが分かる。Rastrigin 関数の場合と同様、時代で分割することが有効に働いていると考えられる。

- 実験 2 に関する考察

図 19 では、 $G_i = 10, 20$ のとき、探索が進むにつれて突然変異率が上昇している。この理由は、局所解から抜け出すために、探索終盤には高い突然変異率が必要であるためと考えられる。一方、Bäck の適応 GA では、探索終盤には突然変異率が極めて低くなっており、探索効率が悪くなっていた [10]。

図 21 では、探索が進むにつれて突然変異率が低下している。この理由は、ここで用いている突然変異オペレータがランダムで選ばれた 2 都市の交換であることが原因と考えられる。この突然変異オペレータで経路を改善できる個体は序盤には豊富であるが、一旦、この経路の改善がなされた場合、同じ 2 都市を再度交換した場合は必ず経路の改悪になる。従って、この突然変異オペレータで改善すればするほど改善できる個体が減少し、それにつれて必要な突然変異率も低下していると考えられる。

図 19 と図 21 を比べると、パラメータ適応の様子が異なる。このことから、性質の違う問題に対しても、A-SAGA はパラメータを適応させることによ

り探索効率を高めていると考えられる。

図 20 と図 22 は良く似た傾向を示している。 e_p は大きな初期値から急減し、その後徐々に上昇している。初期値が大きい理由は、初期個体群はランダムに与えられるために、少数の個体だけを用いて探索するよりも、多数の個体の中から有望な個体を探した方が効率が良いためと考えられる。初期値から e_p が急減する理由は、そこで見つけられた最良の個体の周辺のみを繰返し探索するためと考えられる。その後 e_p が上昇する理由は、最良個体周辺の探索だけでは局所解に陥ってしまうため、その後は徐々に他の個体を取り込むことによって多様性を確保し、初期収束を避けているためと考えられる。

- その他の考察

メタ GA では、適切なパラメータベクトルを用いて短く探索したときよりも、悪いパラメータベクトルを用いているが長く探索したときの方が良い結果が得られる場合がある。例えば、図 17 において、 $E_m \times$ A-SAGA 世代数 = $20,000 \times 20 = 400,000$ の評価回数を費やして、平均 615 の個体適応度を得た。しかし、 $E_m = 40,000$ とすると、たった 1A-SAGA 世代で平均 600 弱の個体適応度が得られた。A-SAGA の改善法としては、 E_m を自動的に決定するアルゴリズムの開発が挙げられる。

A-SAGA は、広範な問題に適用可能な適応 GA であるが、問題の性質がよくわかっている場合には、それらの性質を生かすことができる専門的な手法のほうが高い性能を示す傾向がある。したがって、A-SAGA は性質の良くわかっていない問題に適した手法であると言える。

我々は、実験により得られたパラメータを、Goldberg らのコントロールマップ [14] の概念に当てはめて評価するため、手作業によりコントロールマップに関する具体的な数値を得ようと試みた。しかし、10 変数 Rastrigin 関数に島モデル GA を適用して評価実験を行ったところ、コントロールマップに関する数値を得ることができなかった。具体的には、GA では淘汰圧が強すぎると cross-competitive (淘汰圧が高過ぎて多様性が失われてしまう

状態)になってしまうと述べられているが、最適なスケーリング係数(淘汰圧を決定する)が非常に大きな値になり、値を大きくすることによって性能が下がる境界を観測できなかった。このことにはいくつかの要因があると考えられる。

- Goldbergらの研究では島モデルでないGAが分析されていたが、我々は島モデルGAを用いた。この場合では、それぞれの島において局所解に収束したとしても、異なる局地解に収束したのであれば相互作用を通じて最適解を導く場合があると考えられる。
- Goldbergらの研究では、最適解を見つけられるか否かだけに注目し、局所最適解の存在しないone max problemが用いられていた。しかし、我々は、準最適解であっても適応度が十分高ければそれを評価し、どれだけ高い適応度の平均値を得ることができるかを評価していた。つまり、評価基準が異なっているため、単純な比較ができない。

以上のことから、文献 [14] のモデルは我々の実験結果に当てはまらないと考えられる。

3.5 エージェント指向による拡張性についての考察

現在のA-SAGAでは、個体群の受渡しや移民の交換、パラメータ情報のやりとりなどは固定であり、これらをパラメータとして自動的に適応させることができない。以下では、エージェントがこれら相互作用をいかに利用するかを能動的に決定できるようにすることにより、自動的に適応させるための拡張性について考察する。

我々は、エージェント間での競売システムを導入して個体群を分配する方法を考えている。

各時代のエージェントは最初に、受け取る個体群を競売により購入する。そして個体群を進化させ、次の時代のエージェントのために、競売により個体群を売却する。この購入価格と売却価格の差額を、各エージェントの利得とする。エージェントは、A-SAGAの反復過程を通じて、進化後の個体群をいくらで売却でき

るかの予想を立て、その範囲内で購入価格を競り上げる。ただし、最初の時代のエージェントには外的なシステムが無料で初期個体群を与え、最後の時代のエージェントからは外的なシステムが個体群を買い上げるものとする。このシステムにより、各エージェントは、個体群を進化させるよう動機づけられる。ただし、エージェントが、将来性は低いが高価な個体群を購入することによりリスクを減らす戦略を取る可能性があり、そのようなエージェントばかりになってしまうと、競売価格の下落により、その前の時代のエージェントが個体群を進化させる動機を失ってしまうおそれがある。これをどう抑えるかが効率性を実現するための鍵になると考えている。

また同様に、パラメータのやりとりにも競売システムを導入し、優秀なパラメータを持ったエージェントがより多くの利得を得られるようにするメカニズムの導入も考えられる。

A-SAGA では、現在のところ、すべてのエージェントが同じ能力を持っているものと仮定している。しかし A-SAGA は島モデル GA の構造を持っており、その能力を生かすべく並列化を行った際には、異なる性能を持つ計算機にエージェントが割り当てられるかもしれない。このように能力の対称性が失われた際に A-SAGA がどう働き、またどのような拡張が必要かについてさらなる考察が必要であると考えられる。

3.6 本章の結び

本章では、エージェント指向の手法によりメタ GA と環境分散型並列 GA を組合わせた A-SAGA を提案した。A-SAGA は探索過程に応じて、既存のほとんどの適応 GA を上回る 4 つのパラメータを同時に比較的少ない計算量で適応させることができた。今後の課題として、エージェントに対する GA 以外の進化アルゴリズムの利用、他のエージェントの行動も視野に入れて知的に振舞うエージェントを開発などが挙げられる。

4. おわりに

本論文では，NP 困難のクラスに属する問題を解くためのマルチエージェント技術を用いた近似アルゴリズムに関する研究を行った．

まず，仕事割り当て問題としてのエージェントの協力問題の定式化の一つである TCAP の定義を示し，次に TCAP が最大クリーク問題を含むことを証明することを通じて，TCAP の最適解の定数倍の近似解を求める多項式時間近似アルゴリズムは， $P=NP$ でない限り存在しないことを示した．次に，TCAP の特別な場合である 1 次元仕事へのエージェント集合割り当て問題 (1-TCAP) を定義し，subset sum 問題が 1-TCAP に帰着できることを証明することを通じて，1-TCAP が NP 困難のクラスに属することを示した．また，1-TCAP に対する多項式時間近似アルゴリズムを提案し，その近似比が最悪の場合でも $9/4$ 未満であることを示した．

次に，より一般的な問題に対して，計算に必要なパラメータを自動的に適応する A-SAGA を提案し，その有効性を，メタ GA との探索効率の比較実験により評価し，4 つのパラメータが合理的な計算量で同時に適応させられることを示した．この結果から，A-SAGA は性質の良くわかっていない問題に対して効果的であると考えられる．GA は，ブラックボックス的な組合せ最適化問題に対して適用する手法として有用であるとみなされており，この特性は GA にとって重要であると考えられる．

今後の課題として，複数の組合せ最適化問題をエージェント群に与えたときに，エージェントが自発的に状況を判断し，coalition を形成し，最適化問題を解くようなアルゴリズムの開発が挙げられる．このアルゴリズムにより，性質の異なる問題をエージェントが能動的に分類して解くシステムや，複数のユーザからの多彩な計算要求に応答する分散システムの開発ができると考えられる．

謝辞

本研究を進めるにあたって，研究の指針から指導まで一切のご教授を賜わりました伊藤実教授に厚く御礼を申し上げます．また，本研究をまとめるにあたり，有益なご助言を与えてくださいました石井信教授，並びに杉本謙二教授に厚く御礼を申し上げます．

筆者が奈良先端科学技術大学院大学情報科学研究科在籍中に，ご指導，ご教授くださいました植村俊亮教授，小山正樹教授，鹿野 清宏教授，砂原秀樹教授，千原國宏教授，西谷紘一教授，松本 健一教授，松本 裕治教授，山口 英教授，山下裕助教授，吉川正俊助教授に厚く御礼を申し上げます．

本論文を執筆，また A-SAGA に関する研究をするにあたり，終始熱心に指導してくださった柴田直樹助手に厚く御礼を申し上げます．

本論文を執筆，また英語の論文を執筆するにあたり，熱心に指導してくださった安本慶一助教授に厚く御礼を申し上げます．

1-TCAP に関する修士論文を執筆，また研究をするにあたり，終始熱心に指導してくださった大阪大学情報科学研究科の石原靖哲助教授に厚く御礼を申し上げます．

1-TCAP に関する修士論文を執筆するにあたり，熱心に指導してくださった藤原秀雄教授に厚く御礼を申し上げます．

和歌山大学情報科学研究科の中西隆一助教授は鬼籍に入られましたが，ご生前中，1-TCAP に関する研究をするにあたり，熱心に指導していただきました．厚く御礼を申し上げるとともに，故中西隆一助教授のご冥福をお祈りいたします．

また，A-SAGA に関する実験を行うにあたって多大な貢献をしてくださった，本学博士前期課程 1 年の高島栄一君に厚く御礼を申し上げます．最後に適切な助言から研究室の生活まで，日頃お世話になった伊藤研究室の皆様にも厚く御礼申し上げます．

参考文献

- [1] Bäck, T. Self-adaptation in genetic algorithms. In F.J.Varela, P. B., editor, Proceedings of 1st European Conference on Artificial Life, pp. 263–271, (1992).
- [2] Bull, L., Fogarty, T. C. and Snaith, M. Evolution in Multi-agent Systems: Evolving Communicating Classifier Systems for Gait in a Quadrupedal Robot. Proceedings of the Sixth international conference on Genetic Algorithms. pp. 382–388, (1995).
- [3] Bulkhard, H. D. Fairness and control in multi-agent systems. Theoretical computer science. Elsevier Science. Vol.189 No.1, pp. 109–127, (1997).
- [4] Cant’u-Paz, E. A Survey of Parallel Genetic Algorithms. Calculateurs Pralleles, Reseaux et Systems Repartis, Paris: Hermes, 10(2):pp. 141–171, (1998).
- [5] Badell M., Nougues J.M., and Puigjaner L. Integrated on line production and financial scheduling with intelligent autonomous agent based information system. European symposium on computer aided process engineering. Computers and chemical engineering. Pergamon press. Vol.22 suppl., pp. 271–277, (1998).
- [6] Coello, C. A. and Pulido, G. T. Multiobjective Optimization using a Micro Genetic Algorithms. Proceedings of the Genetic and Evolutionary Computation Conference. July 7-11, 2001. Sanfrancisco, California. pp. 274–282, (2001).
- [7] Cormen T.H., Leisrson C.E., and Rivest R.L. Introduction to algorithms. The MIT electrical engineering and computer science series. Mcgraw-hill book company, (1989).
- [8] Espinoza, F., Minsker, B.S. and Goldberg, D.: *A Self-Adaptive Hybrid Genetic Algorithm*, Proceedings of the Genetic and Evolutionary Computation

- Conference, ,San Francisco, Morgan Kaufmann Publishers, p. 759, (2001).
- [9] French, R. L. B. and Damaper, R. I. Evolving a Nervous System of Spiking Neurons for Behavior Robot. Proceedings of the Genetic and Evolutionary Computation Conference. July 7-11, 2001. Sanfrancisco, California. pp 1099–1106, (2001).
- [10] Glicman, M. R. and Sycara, K.: Reasons for Premature Convergence of Self-Adapting Mutation Rates. Proceedings of the Congress on Evolutionary Computation, pp. 62–69, (2000).
- [11] Goldberg, D. Genetic Algorithm in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA, (1989).
- [12] Goldberg, D. Sizing populations for serial and parallel genetic algorithms, In J. Davis Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, pp. 70–79, (1989).
- [13] Goldberg, D. and Deb, K. A comparative analysis of selection schemes used in genetic algorithms. In Gregory J.E. Rawlins, editor, Foundations of Genetic Algorithms, Morgan Kaufmann Publishers, pp. 69–93, (1991).
- [14] Goldberg, D., Deb, K. and Dirk, T. Toward a better understanding of mixing in genetic algorithms. Journal of the Society of Instrument and Control Engineers, 32(1):pp. 10–16, (1993).
- [15] Fujita, S. and Yamashita, M. Approximation Algorithm for Multiprocessor Scheduling Problems. IEICE trans. inf. and syst., vol. E83-D, No.3, March, pp. 503–509, (2000).
- [16] Hochbaum D.S. Ed. Approximation algorithms for NP-hard problems. PWS publishing company, (1997).
- [17] Huhns, M. N. and Singh, M. P. Ed. Readings in Agents, Morgan Kaufmann Publishers, Inc. San Francisco, California, (1998).

- [18] Ishii, H. and Tada, M. Approximation Algorithms for Scheduling Problems. IEICE trans. inf. and syst., vol. E83-D, No.3, March, pp. 496–502, (2000).
- [19] Itoh T. and Shintani T. An approach to multi-agent based scheduling system using a coalition formation. Industrial and engineering applications of artificial intelligence and expert systems : IEA/AIE 96 : proceedings of the ninth international conference, Fukuoka, Japan, June 4-7, 1996. edited by Takushi Tanaka, Setsuo Ohsuga, Moonis Ali. Gordon and Breach, p. 780, (1996).
- [20] Kee, E., Airey, S. and Cye, W.: *An Adaptive Genetic Algorithm*, Proceedings of the Genetic and Evolutionary Computation Conference, pp 391–397, (2001).
- [21] Koller D. and Pfeffer A. Representations and solutions for game theoretic problem. Artificial intelligence : an international journal. North-Holland. Vol.94 No.1, pp. 167–215, (1997).
- [22] Kraus S., Wilkenfeld J., and Zlotkin G. Multiagent negotiation under time constraints. Artificial intelligence : an international journal. North-holland. Vol.75 No.2, pp. 297–345, (1995).
- [23] Kraus S. Negotiation and cooperation in multi-agent environments. Artificial intelligence : an international journal. North-holland. Vol.94 No.1, pp. 79–97, (1997).
- [24] Krink, T. and Ursem, R. K. Parameter Control Using the Agent Based Patchwork Model. Proceedings of the Congress on Evolutionary Computation, pp. 77–83, (2000).
- [25] Lobo., F. G. The parameter-less genetic algorithm: rational and automated parameter selection for simplified genetic algorithm operation. PhD thesis, University of Lisbon, Portugal, (2000).

- [26] Miki, M., Hiroyasu, K., Kaneko, M and Hatanaka, I. A Parallel Genetic Algorithm with Distributed Environment Scheme. IEEE Proceedings of Systems, Man and Cybernetics Conference SMC'99, pp. 695–700, (1999).
- [27] Mundhe, M. and Sen, S. Evolving agent societies that avoid social dilemmas. Proceedings of the Genetic and Evolutionary Computation Conference. July 10-12, 2000. Las Vegas, Nevada. pp. 809–816, (2000).
- [28] 村田 佳洋, 石原 靖哲, 伊藤 実. 仕事に対するエージェント集合割り当て問題の近似アルゴリズム. 電子情報通信学会技術研究報告, COMP99-56 ~ 68, p.25–32, (1999).
- [29] 村田 佳洋. 仕事に対するエージェント集合割り当て問題の近似アルゴリズム. 奈良先端化学技術大学院大学修士論文, (2000).
- [30] Murata, U. Ishihara, Y. Ito, M. An Approximation Algorithm for the Task-Coalition Assignment Problem. The Institute of Electronics, Information and Communication Engineers, Vol.E85-D No.4 pp. 685–693, (2002).
- [31] 村田 佳洋, 柴田 直樹, 伊藤 実. エージェント指向自己適応遺伝アルゴリズム. 情報処理学会研究報告, Vol.2002, No.89, 2002-MPS-41, pp. 21–24, (2002).
- [32] Murata, Y. Shibata, N. Yasumoto, K. and Ito, M. Agent Oriented Self Adaptive Genetic Algorithm Communications and computer networks(CCN). November 4-6, 2002 cambridge, USA, pp. 348–353, (2002).
- [33] 村田 佳洋, 柴田 直樹, 伊藤 実. エージェント指向自己適応遺伝アルゴリズム. 「情報処理学会論文誌:数理モデル化と応用」(TOM) に投稿, 採録決定.
- [34] Ochoa, G., Harvey, I. and Buxton, H. Optimal Mutation rates and Selection Pressure in Genetic Algorithm. Proceedings of the Genetic and Evolutionary Computation Conference. July 10-12, 2000. Las Vegas, Nevada. pp. 315–322, (2000).

- [35] Sandholm T.W. and Lesser V.R. Coalitions among computationally bounded agents. *Artificial intelligence : an international journal*. North-holland. Vol.94 No.1, pp. 99–137, (1997).
- [36] Sandholm, T., Larson, K., Andersson, M., Shehory, O. and Tohme, T. Coalition structure generation with worst case guarantees. *Artificial Intelligence* 111, pp. 209-238, (1999).
- [37] Sandholm, T. and Zhou, Y. Surplus equivalence of leveled commitment contracts. *Fourth International Conference on MultiAgents Systems(ICMAS-2000)*. pp. 247–254, (2000).
- [38] Sandholm, T. and Ferrandon, V. Safe Exchange Planner. *Fourth International Conference on MultiAgents Systems(ICMAS-2000)*. pp. 247–254, (2000).
- [39] Sato, H. and Namatame, A. Co-evolution in Social Interactions. *Congress on Evolutionary Computation 2001*. May 27-30, 2001. COEX, Seoul, Korea. vol. 2, pp. 1109–1114, (2001).
- [40] Shehory O. and Kraus S. Task allocation via coalition formation among autonomous agents. In *Proc. of IJCAI-95*, pp. 651–661, (1995).
- [41] Shehory, O. and Kraus, S. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. *International Conference on MultiAgents Systems(ICMAS-96)*. pp. 330–337, (1996).
- [42] Shehory O. and Kraus S. Methods for task allocation via agent coalition formation. *Artificial intelligence : an international journal*. North-holland. Vol.101 No.1, pp. 165–200, (1998).
- [43] Shimomura, M., Kubo, M. and Mohri, H. The Development of Software Components for Solving the Vehicle Routing and Facility Location Problems. *IEICE trans. inf. and syst.*, vol. E83-D, No.3, March, pp. 510–518, (2000).

- [44] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>
- [45] Weiss, G. Ed. *Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence*. The MIT Press Cambridge, Massachusetts London, England, (1999).
- [46] Yenn, T. S., Lee, K. H. and Yang, Y. S. Function approximations by compiling neural networks and genetic programming trees with oblique decision trees. *Artificial; Intelligence in Engineering* 13, pp. 223–239, (1999).