|  |
| --- |
| NSGA-Ⅲアルゴリズムを用いた  Web三層モデルの最適化に関する研究 |
|  |
| 齋藤篤志†1　棟朝雅晴†2 |
|  |
| **概要**：コンピュータリソースを仮想化し、ユーザーが利用したいときに利用したい分だけ利用することができるクラウドコンピューティングは、その利便性の高さから、個人から企業まで広く利用されるようになった。クラウドリソースを選択する幅が広がった一方で、ユーザーは自身の要求要件に最適なクラウドリソースを見つけ出す必要性に迫られるようになった。この問題の解決策として、ユーザーの要求に対して最適なプランを提示するクラウドブローカーサービスが求められている。本論文ではクラウドブローカーサービスに適した数理モデルを提案し、NSGA-3アルゴリズムを用いて実用的なクラウドブローカーサービスの実現を行う。 |
|  |
| **キーワード**：クラウドコンピューティング, 遺伝的アルゴリズム, NSGA-Ⅲ, Web三層モデル, クラウドブローカー |
|  |
| Research on Web Three-Tier Systems Optimization  Using NSGA-III algorithm |
|  |
| ATSUSHI SAITO†1　MUNETOMO MASAHARU†2 |
|  |
| ***Abstract***: Cloud Computing (vertualized network, server, storage, applycation and service) have been widly used among orginary people and enterprises. We can choose from a lot of cloud resources, but, on the other hand, we have to find better resource which suit our request. To solve this problem, the cloud broker service which can propose best plans for user request is needed. In this paper, we suggest the mathmatical model for cloud broker service, and achieve practical cloud broker service using NSGA-Ⅲ algorithm. |
|  |
| ***Keywords***: Cloud Computing, Genetic Algorithm, NSGA-Ⅲ, Web 3-tier model, Cloud Broker Service |

# はじめに

　クラウドコンピューティング技術の発達により、ユーザーは自前でコンピュータリソースを確保することなく、コンピュータリソースを手に入れることができるようになった。その利便性の高さから瞬く間にクラウドコンピューティングは浸透し、現在では数多くのクラウドベンダーからクラウド化されたコンピュータリソース（クラウドリソース）や、クラウドを用いたサービス（クラウドサービス）が提供されている。

ユーザーにとっては選択の幅が大きく広がった一方で、自身の要求要件に適するクラウドリソースやクラウドサービスを選択することは非常に難しい状況にあると推察される。なぜならば、自身のシステムを構築するために、システム内のコンポーネントを構成するクラウドリソースを、複数のクラウドベンダーが提供するクラウドリソースの一覧表から一つ一つ検討する必要があるためである。

このような状況下で、クラウドブローカーサービスが求められている。クラウドブローカーサービスは最適なクラウドリソースの選択や、複数のクラウドサービスの統合管理・運用をサポートするサービスのことである。

この論文では、ユーザーの要求を記述したユーザーリクエストを最適化エンジンに送り、最適化エンジンはクラウドの情報が格納されているデータベースから情報を利用して最適解を選択し、結果をユーザーに返すという一連の動作を行うクラウドブローカーサービスを想定している。

クラウドブローカーサービスの最適化エンジンが解くべき問題は制約充足付き多目的最適化問題である(詳細な検討は3節で行う)が、2節で議論するように、現状では実用に耐えるアルゴリズムを用いたクラウドブローカーは提案されていない。

そこで本論文では、進化的計算手法である遺伝的アルゴリズムの一種であるNSGA-Ⅲアルゴリズムを用いて、クラウドリソースの選択をサポートする実用的なクラウドブローカーサービスを提案する。NSGA-Ⅲアルゴリズムの簡潔な紹介を含む提案手法は4節で、実験は5節で、結論は6節で述べる。

# 関連研究 [[1]](#footnote-1)\*【\*の文字書式「隠し文字」】

クラウドリソース選択に関する研究では古典的最適化手法を用いた数多くの研究がある。この節では一部の典型的な研究を議論する。

まずsundarswaranらの研究がある。クラウドブローカーサービスの基本構造を示した上で、無数にあるクラウドリソースを符号化して管理する手法を述べている。この論文の主眼は効率的なインデックスと探索を行なうことであった。確かにCSP indexを用いてクラウドリソースをバイナリ符号化し、CSS-query algorithmでハミング距離を用いて探索をすることは高速であろう。しかし、ハミング距離が近いからといって、それが最適解である保証がないことが大きな問題であり、ユーザーが要求した要件を満たした解を提示できないと思われる。

gragらの研究ではCSMICが提案するService Mesurement Index (SMI)というクラウドサービス評価基準を用い、それぞれが階層的に示された7つの性質に基づいて評価を行なう。それぞれのサービスをSMIで評価した上で、階層分析法(AHP)を用いてインスタンスの選択を行なっている。AHPは階層構造を探索するために優れているため、SMIとともに用いる必然性がある。しかし、AHPによる選択は結局のところ、荷重和法の延長と言えるもので、パレート最適解の効率的な探索には不十分である。

このように古典的手法におけるクラウドリソース選択問題では、複数のパレート最適解を効率的に得ることが難しく、それは数多くのクラウドリソースを探索する上で計算上のボトルネックとなるだろう。その上、単体のクラウドリソースを選択するだけでは実用的ではなく、ユーザーが要求するシステム全体の最適化を目指さなければならない。システム全体を最適化するためには、単体のクラウドリソースを最適化して組み合わせるだけでは不十分であろう。次に挙げる二つの論文はシステム全体の最適化を指向するものである。

川勝らの研究では、世界的な計算機資源に焦点を当て、三層アーキテクチャの最適化を目指しており、最適化アルゴリズムとしてNSGA-Ⅱアルゴリズムを用いている。この論文の主眼は、データセンター内の計算機資源を最適化することであり、当然ながらクラウドブローカーサービスにおいて主眼となるユーザー目線での選択基準で目的関数は設定されていない。また、NSGA-Ⅱアルゴリズムを用いており、実験にかかった時間には触れていないが、目的関数の数が増加した場合に混雑度の計算で相当な時間がかかることが予想される。本論文ではNSGA-Ⅲアルゴリズムを用いて、目的関数の数の増加した場合にも実用可能な時間で処理できる点で優位性がある。

pawlukらの研究ではクラウドブローカーサービスのアーキテクチャや構築方法について、ほとんどの紙幅が割かれているが、クラウドブローカーサービスを実装し、最適なアーキテクチャを。ただし、実験では荷重和法が用いられており、効率的な最適解の探索ができているとは言いがたい。また、目的関数も一つから二つしか用いられておらず、試験的な実装であり、ユーザーの本当に求めるサービスが提案できないだろう。しかし、アーキテクチャを最適化し提示している点で上記二つの研究に比べて実用的であるといえる。

# 数理モデル [[2]](#endnote-1)\*【\*の文字書式「隠し文字」】

　クラウドブローカーシステムが解くべき問題を制約条件つき多目的最適化問題として定義する。実用的なクラウドブローカーシステムにおいては、ユーザーからの要求や、クラウドを提供する企業によって複数の制約がつくことが普通であり、最適化をする際には考慮せざるを得ない。例えばユーザーからの要求では、クラウドを設置する地域に関して、自国外に置くことに抵抗があることもあるだろう。このような場合、仮に最適解が国外に設置することであっても、ユーザーに国外に設置するプランを提示すべきではない。また、クラウドを選択する場合、クラウドを選択する基準がコストや仮想マシンのパフォーマンス、耐障害性など、様々であるため、それは多目的最適化問題になる。よってクラウドブローカーシステムの数理モデルは制約条件つき多目的最適化問題である。

数式で表すならば、変数空間の次元をn、目的関数の個数をmとし、制約充足付き多目的最適化問題の解を 、 、可能領域を とすると、以下のように記述される。

また可能領域は一般に個の制約条件を満たす領域として以下のように定義できる。

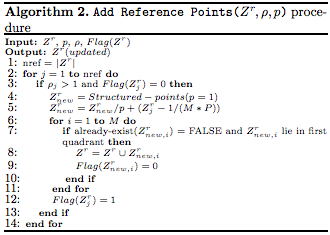
# 提案手法 [[3]](#endnote-2)\*【\*の文字書式「隠し文字」】

**4.1　NSGA-Ⅲアルゴリズム**

　NSGA-Ⅲアルゴリズムは、NSGA-Ⅱアルゴリズムをさらに改善した遺伝的アルゴリズムの一種である。そもそも遺伝的アルゴリズムとは生物の進化の過程を参考にして作られた計算アルゴリズムであり、進化計算分野の中でも最も広く使われている。遺伝的アルゴリズムでは遺伝子を模した個体群を用意し、一世代経るごとに遺伝子の選択、交叉、突然変異を繰り返し、評価関数により適応した適応度の高い個体群が淘汰されずに生き残る。

遺伝的アルゴリズムの利点として、問題に固有で設定が必要なパラメータを持たない点がある。その代わりに交叉、突然変異が起こる確率である遺伝パラメータが存在するが、固有パラメーターを設定するよりも容易である。 もう一つの利点として、解空間が不明の場合でも適用できる点が挙げられる。 問題によっては、解空間がいくつかに分断されている可能性もあるが、その ような場合でも遺伝的アルゴリズムは探索が可能である。さらにもう一つ利点を挙げるならば、複数の個体が同時に解空間を探索する点が挙げられる。 これにより、効率的なパレート最適解の探索を可能にしている。

NSGA-Ⅲアルゴリズムでは、多目的最適化問題を解くために開発されたNSGA-Ⅱアルゴリズムから高速非優越ソートとエリート主義を引き継ぎ、NSGA-Ⅱアルゴリズムで計算のボトルネックになっていた混雑度の計算は、参照点を用いた選択に変更されている。これらの工夫により、目的関数の数が増加したとしても（目的関数が4つ以上のmany-objectiveに対応可能）、パレートフロント上で広範囲かつ均等に最適解群を得ることができ、かつ計算時間は大幅に軽減されている。具体的なアルゴリズムの手順は以下のようになっている。

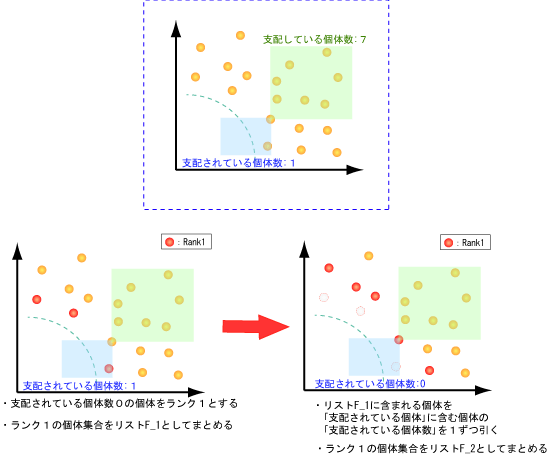


(図1 :NSGA-Ⅲの手順(参考文献2より))

4.1.1　高速非優越ソート

NSGAアルゴリズムで用いられていた非優越ソートでは、ある個体と他の個体の優越関係を一つ一つ確認することで、ランク付けを行っていた。この方法は確かにパレートフロントを算出することはできるが、計算コストが非常に高い。複雑性は各ランクに1個体しか存在しないという最悪の場合でO(mN3)である。

そこでNSGA-Ⅱから高速非優越ソートが導入された。高速非優越ソートでは各個体に対して、優越している個体と優越されている個体の数を同時に数える。優越されている個体がゼロの個体をランク1、ランク1の個体を取り除いて優越されている個体がゼロの個体をランク2とする（図１参照）。これを全個体がなくなるまで繰り返すことによって、高速なランク分けを実現している。高速非優越ソートの複雑性は最悪の場合でO(N2)である。



（図2 :高速非優越ソートの手順（参考文献7より））

4.1.2　エリート主義

NSGA-Ⅱアルゴリズムでは個体の選択の際に、エリート主義を導入している。エリート主義とは常に優良個体を保存する親母集団と探索用の子母集団を使い、探索で発見した優れた解の消失を防ぐ手法である。

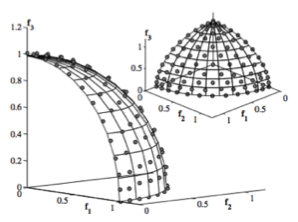
手順は次のようになっている。世代tにおいて、親母集団をPt、子母集団をQt、PtとQtを合わせた合成集団をRtとする。PtからQtを選択し、選択されたQtに選択、交叉、突然変異という遺伝的操作を加えてQtを更新する。そしてQ'tと親母集団Ptを組み合わせてRtを生成する。RtからPt+1を選択し、次の世代へと移行する。

4.1.3　参照点による選択

NSGAではシェアリングパラメータρを定めることにより、一定範囲内で複数の解を選択することを禁じていた。これにより集団の多様性を確保することができる。しかし一方で、このパラメータを事前に定める必要性があることや、計算負荷が大きい点が問題となっていた。

NSGA-Ⅱではシェアリングパラメータを必要としない混雑度という概念を創出し、個体間の距離を調べることで集団の多様性を確保していたが、これもNSGA同様に計算負荷が大きい点は改善されなかった。

そこでNSGA-Ⅲでは参照点による選択が採用されている。参照点とはパレートフロント上に規則的に配置され、最も近い個体が選択される"点"のことである。参照点の数𝐻はおおよそ可能解の数と同等とし、目的関数の数を𝑀、任意の数を𝑝としてH = M+p−1Cpで表される。参照点の数がH = M+p−1Cpで表されるのは、パスカルの三角形に従って均等に配置するためである。これによりパラメータの設定なしに、パレートフロント上で広範囲の解を選択することができ、しかも計算時間は大幅に軽減可能である。

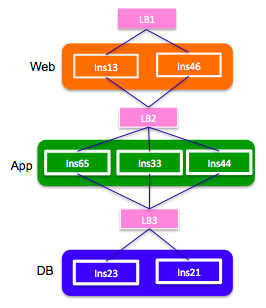


（図3 : 参照点による選択（参考文献1より））

4.2　遺伝子表現

本論文では遺伝子をWeb三層モデルの一つの構造として定義する。三層アーキテクチャはユーザーインターフェース、アプリケーションサーバー、データベースという、三つの独立したモジュールとして構成される。三層はそれぞれ独立しているため、別々に開発・改変することが可能であり、技術革新やユーザーの要求に対応しやすい。Web三層モデルは現在主流となっているアーキテクチャであり、ほとんどのWebサービスはこのWeb三層モデルで構成されている。

本論文では遺伝子表現として、可変長の二次元配列にWebサーバー用クラウド、アプリケーションサーバー用クラウド、データベースサーバー用クラウドの3種類を格納する。それぞれの層には独自に実装したデータベースアプリケーションから取得したインスタンスのタイプ（AWS, Google Cloud, Rackspaceから得たデータ）が格納される。population[x]={13,46|65,33,44|23,21}の場合、図のように配置されることを想定する。



(図4 :Web3層モデルでの遺伝子表現)

4.3　目的関数

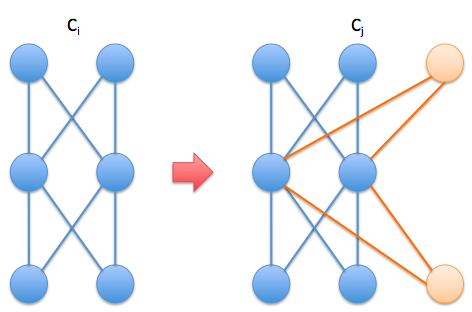
本論文ではNSGA-Ⅲアルゴリズムでの最適化に用いる４つの目的関数を提案する。なお、遺伝子が示す構成をciとする。

4.3.1　構成コスト

構成コストの目的関数fconfigration\_costでは、ある構成ciからある構成cjに、構成を変更した場合にかかるコスト（金額）を求める。もし、初めてシステムを組み立てる場合は、ciは何もない構成であり、cjは新しく考えられた構成と考える。

Web サーバーの構成の変更にかかるコストをCweb 、アプリケーションサーバーの構成の変更にかかるコストをCapp 、データベースサーバーの構成の変更にかかるコストをCdb 、ロードバランサーの構成の変更にかかるコストをClbとすると、

で表すことが出来る。



（図5：構成変更の例。オレンジが変更部分。）

4.3.2　オペレーションコスト

オペレーションコストの目的関数foperation\_cost で求めるのは、遺伝子が示すシステムを構成の場合、 一ヶ月あたりにかかる料金である。 Webサーバーのインスタンス料金の合計 をPweb 、アプリケーションサーバーのインスタンス料金の合計をPapp 、データベースサーバーのインスタンス料金の合計 をPdb 、ロードバランサーの料金の合計 をPlb 、VMの監視料金の合計をPwatchとすると以下のように記述される。

4.3.3　SLA

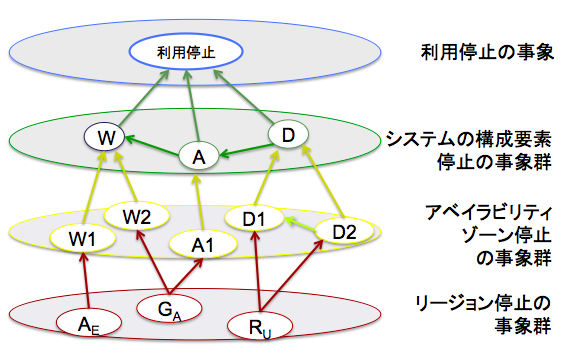
SLAとはサービスの品質を保証する制度のことであり、ユーザーがサービスプロバイダと契約する際にサービスの目安として利用する。SLAには回線の最低通信速度、ネットワーク内での平均遅延時間、利用不能時間の上限などのサービス保証項目や、それを実現できなかった場合の返金に関する規定が含まれる。クラウドサービスにおいては可用性・安定性の指標としても利用されている。この論文においては、特に稼働率を指すこととする。

稼働率とは稼働が期待される一定の期間内で、システムやハードウェアが正常に稼働している時間の割合のことである。SLAの目的関数fSLAでは遺伝子が示すシステムの稼働率を評価値とする。今回はベイジアンネットワークを用いて、利用停止に至る確率事象を計算し、一年間のうち何時間停止するかを計算する。

一般にベイジアンネットワークをとする。ただし、は確率構造、は条件確率付きパラメータである。n個の離散変数集合の場合、同時分布は一般に、

である。ベイジアンネットワークではを所与としているので、親ノード変数集合として、以下のように表せる。

よって、fSLAは以下のように計算される。



（図6：ベイジアンネットワークによるシステムがダウンする事象の分析）

4.2.4　パフォーマンス

パフォーマンスの目的関数ではユーザーが求めるインスタンスのパフォーマンスに、遺伝子が合致する度合いを評価値とする。ユーザーリクエストは1秒あたりに処理されるリクエスト数をスループットとして、各層ごとに要求されることを想定した。それに対して各層のインスタンスのスループットは、コア数の合計を以下のように表したとき

Web サーバーのコア数の合計をCRweb 、アプリケーションサーバーのコア数の合計CRapp 、データベースサーバーのコア数の合計をCRdb 、VMのクロック数をCl、１リクエストにかかるクロック数をPrとすると、以下のように表せる。

そして各層で求めたスループットの値とユーザーリクエストで要求された値を用いて、以下の目的関数で評価する。

# 実験

5.1 適切なシステムの提案

実験１ではユーザーリクエストから適切な構造を提示できるかを実験する。

親母集団 P (t);100 個体

子母集団 Q(t);100 個体

停止条件;100世代

交叉の割合;0.1

突然変異の割合;0.1

参照点の数 H = M+p−1Cp = 3+19−1C19 = 210 個

とパラメータを設定した。その上で、ユースケースとして以下のユーザーリ クエストを入力した。

Webサーバーのスループット;200(request/sec)

Appサーバーのスループット;100(request/sec)

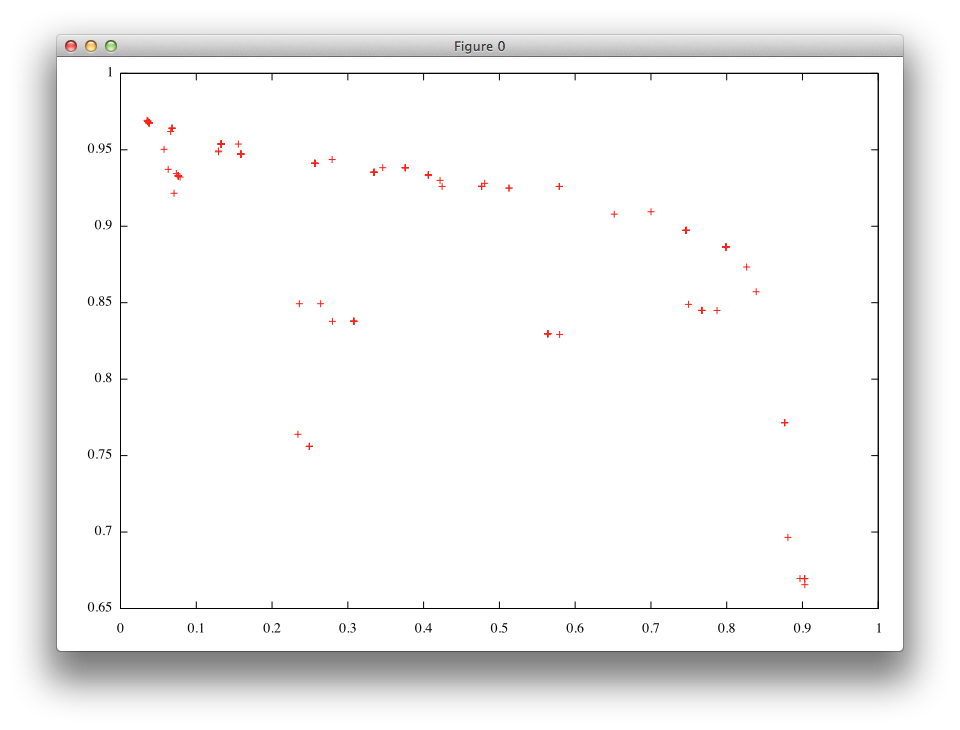
DBサーバーのスループット;100(request/sec)

Web サーバーのストレージ容量;100GB以上

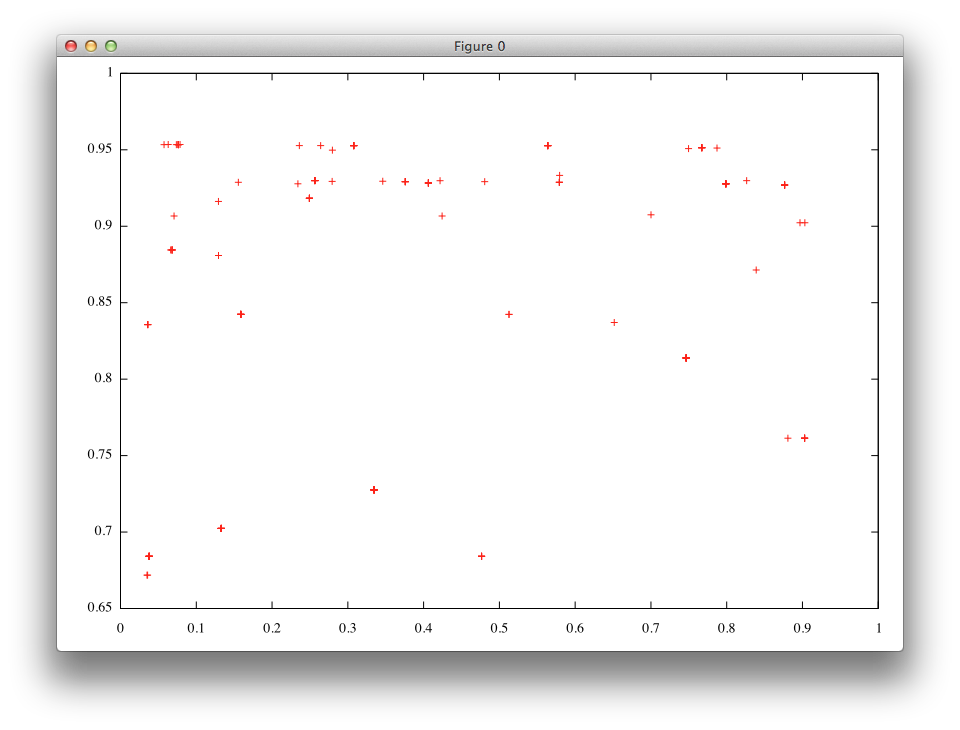
Appサーバーのストレージ容量;100GB以上

DB サーバーのストレージ容量;500GB 以上

図は最終世代において制約で個体をはじいていない状態の個体の分布を示している。x軸はコストによる値、y軸はパフォーマンスによる値、z軸は耐障害性による値である。二つのグラフからパレートフロントが観測でき、参照点による選択のときに、これらの個体群からパレート最適解が選択されていると考えられる。



（図7 :コスト対パフォーマンス）



（図8 :コスト対SLA）

5.2　処理時間

実験2では最適化にかかる処理時間を計測する。実験1と同様にパラメータを設定した。その上で、実験1と同様のユーザーリクエストを入力したものとして、100 回の平均値と分散、標準偏差をとる。結果として、100回の平均は537.72(ms)、分散は28664、標準偏差は169.30となった。リクエストを送って返ってくるまでに0.5秒程度であり、実用する上で十分な結果であろう。

# おわりに [[4]](#endnote-3)\*【\*の文字書式「隠し文字」】

本論文では、クラウドサービスを選択する問題の解決策となるクラウドブローカーサービスに適した数理モデルを提案し、NSGA-Ⅲアルゴリズムを用いてこの問題を解いた。適切な構造を提示することに関しても処理時間に関しても問題がないことが分かった。今後の課題として、まず挙げられるのが、Web 三層モデル以外のアーキテクチャの最適化である。本論文では Web三層モデルの最適化を行なったが、Web 三層モデル以外のアーキテクチャの最適化では、遺伝子の設計、目的関数の設定などが困難を極めるだろう。しかし、クラウドブローカーサービスの実用を考えた場合、必ず乗り越えなければならない課題である。

**参考文献**

1) Himanshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sort- ing approach, part ii: Handling constraints and extending to an adap- tive approach. Evolutionary Computation, IEEE Transactions on, 18(4):602–622, 2014.

2）Jain, Himanshu, and Kalyanmoy Deb. "An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization." Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2013.

3）Smitha Sundareswaran, Anna Squicciarini, and Dan Lin. A brokerage- based approach for cloud service selection. In Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, pages 558–565. IEEE, 2012.

4）Przemyslaw Pawluk, Bradley Simmons, Michael Smit, Marin Litoiu, and Serge Mankovski. Introducing stratos: A cloud broker service. In IEEE CLOUD, pages 891–898, 2012.

5）Saurabh Kumar Garg, Steve Versteeg, and Rajkumar Buyya. A frame- work for ranking of cloud computing services. Future Generation Com- puter Systems, 29(4):1012–1023, 2013.

6）川勝崇史, 棟朝雅晴. 分散クラウド環境における sla を考慮した web システムの多目的資源割当最適化. 情報処理学会研究報告. MPS, 数理 モデル化と問題解決研究報告, 2013(9):1–6, 2013.

7）同志社大学 知的システムデザイン研究室 ホームページ 「NSGA-２（非優越ソートGA2）」http://mikilab.doshisha.ac.jp/dia/research/mop\_ga/moga/3/3-5-5.html

|  |
| --- |
| 齋藤 篤志（非会員）  2015.3 北海道大学工学部情報エレクトロニクス学科 卒業  2015.4 北海道大学大学院情報科学研究科 入学 |
| **棟朝 雅晴**（正会員）  1996.4 北海道大学大学院工学研究科 助手 （情報解析学分野）  1998.6 ～ 1999.3 Visiting Scholar, Illinois Genetic Algorithms Laboratory, Department of General Engineering, University of Illinois at Urbana-Champaign, Illinois, USA.  1999.10 北海道大学情報メディア教育研究総合センター 助教授 （情報メディアシステム分野）  2003.4 北海道大学情報基盤センター 助教授 （大規模計算システム研究部門）  2007.4 北海道大学情報基盤センター 准教授 （大規模計算システム研究部門）  2012.8 北海道大学情報基盤センター 教授（デジタルコンテンツ研究部門） |

1. \* †1 北海道大学大学院情報科学研究科

   Hokkaido University Graduate School of Infomation Science and Technology

   †2 北海道大学大学院情報科学研究科 教授

   Hokkaido University Graduate School of Infomation Science and Technology professor

   [↑](#footnote-ref-1)
2. \* [↑](#endnote-ref-1)
3. \* [↑](#endnote-ref-2)
4. \* [↑](#endnote-ref-3)