

データセンタの持続可能な運用のための多目的最適化手法の開発

代表研究者 鮫島 正樹 大阪大学 大学院情報科学研究科 マルチメディア工学専攻 助教

1 はじめに

クラウドコンピューティングの普及に伴い、科学計算のための High-Performance Computing (HPC) のクラウドへの移行が進んでいる [1, 2]. ユーザがクラウド上の HPC にジョブを送信すると、データセンタの HPC サーバによってジョブは処理される. クラウド上の HPC を利用する利点として、ユーザはサーバを購入することなく利用できるため、一般的にコストを低く抑えることができる. また、HPC サービスの提供者にとっては、サーバをデータセンタに集約できるため、空調や電力といったランニングコストを抑えることができるという利点がある. しかし、データセンタにおけるサーバの収容能力には限りがあるため、複数のデータセンタに分散し、それらを高速のネットワークで接続することが行われている. 本研究では、このような分散データセンタを研究対象とする.

分散データセンタは、分散に伴って追加の電力を必要とするため、その電力を安価に確保することが重要となる. 近年のデータセンタは、屋根に太陽光発電システムを搭載するなど、データセンタ自体に発電の機能が備えられている. 得られた電力をデータセンタの運用に利用することで、外部から調達する電力を低減することができる [3]. データセンタにおいては、ジョブを処理する際に HPC サーバが稼働し、より多くの電力を消費することから、発電量が多い時間帯 (一般的に昼間) にジョブを処理することが、調達電力量の低減に有効である. 一方、ユーザはジョブに対して、ジョブの処理が完了すべき時刻 (完了希望時刻) を与えており、なるべく完了希望時刻内に処理を終えることが望ましい. 調達電力量の低減のみを考慮して、すべてのジョブを昼間に処理するようなジョブスケジューリングでは、完了希望時刻以内にジョブを完了できず、完了希望時刻からの遅延時間を生じてしまう. 従って、調達電力量と遅延時間の双方を最小化する多目的ジョブスケジューリング問題を解くことが求められている [4].

調達電力量と遅延時間の双方を最小化するジョブスケジューリング問題について、様々なアルゴリズムが提案されている. ヒューリスティックな方法として、完了希望時刻までの猶予時間が短いジョブから順に、調達電力量を最小化するようにジョブスケジューリングを作成する GreenSlot [5] とよばれるアルゴリズムが提案されている. また、より良いジョブスケジューリングを決定するため、多目的整数計画問題に定式化して解く方法も提案されている. 多目的整数計画問題を解くためには膨大な計算時間が必要であり、近似解法として進化的アルゴリズム等のメタヒューリスティクス [6, 7, 8] が用いられている. これらのアルゴリズムでは、データセンタにおける発電量の予測誤差を考慮せず、スケジューリングを行っている.

本研究では、調達電力量と遅延時間の双方を最小化するジョブスケジューリング問題において、発電量の予測誤差を考慮した定式化を行い、多目的最適化のための効率的なアルゴリズムを提案する. 具体的には、発電量を確率変数で表現し、機会制約条件を定式化に導入する. さらに、多目的最適化問題の厳密解法である Epsilon-constraint 法 [9] にもとづいて、効率的な最適化手法を提案する.

2 関連研究

データセンタにおける調達電力を最小化するジョブスケジューリング手法に関して、これまで多くの研究が行われている. その一つのアプローチとして、利用しないサーバを停止することにより、消費電力を低減する方法がある [10, 11]. サーバを停止すると、その間の消費電力を大幅に低減できる代わりに、ジョブを処理することができなくなってしまう. 多数のジョブを処理するために、停止したサーバを再度起動すると、起動時に多くの消費電力を必要とする. 従って、再起動の回数をなるべく抑え、サーバの停止期間を長期化するようなスケジューリングを作成することが、消費電力の低減に有効である. もう一つのアプローチとして、データセンタにおける再生可能エネルギーを利用することで、外部に調達する電力を削減する手法があり、本研究ではこちらに着目する.

再生可能エネルギーの利用を考慮した初期のスケジューリング手法として、GreenSlot [5] とよばれるヒューリスティックな手法が提案されている. GreenSlot では以下のようにジョブのスケジューリングを行う.

1. 余裕時間が小さいジョブから順に高い優先度を与える。余裕時間は、完了希望時刻までの時間から処理時間を引いた時間である。
2. 優先度の高いジョブから順に選択し、そのジョブを処理するのに十分な CPU コアを有するデータセンタを列挙する。調達電力が最小となるような時間とデータセンタを選択して、ジョブを割り当てる。

また、同様のスケジューリング問題を混合整数計画問題に定式化し、最適解を求める研究もなされている[12]。これらの研究では、ジョブの処理時間は十分に短く、遅延時間が生じないということを仮定し、調達電力量のみを最小化することを目的としている。この仮定によって、スケジューリング問題を各時間におけるジョブの割り当て問題に分解することができ、短時間で最適解が求まることを示している。また、発電量の不確実性を考慮して、発電量を確率変数で表現し、定式化に組み込む研究も報告されている。発電量の期待値を最小化するように定式化し、Lyapunov 最適化を用いた解法が提案されている[13, 14]。

上記の問題は単一目的の最適化問題として定式化されているが、調達電力量や遅延時間等を目的関数として、多目的最適化問題として解く研究も報告されている。多目的最適化問題は、単一目的の最適化問題よりも、求解に多くの時間を要する。そのため、進化計算等のメタヒューリスティクスを利用した近似アルゴリズムが提案されている[6, 7, 8]。

上述したように、再生可能エネルギーを利用したジョブスケジューリング問題に関して、確率的最適化もしくは多目的最適化を行う研究はすでに行われている。本研究では、これらを同時に扱う確率的多目的最適化問題として定式化し、解くことを目的とする。加えて、既存の定式化では確率変数を含む式を期待値で評価していたが、実際の運用においては悲観的なスケジュールを採用することが多いことに着目する。例えば、スケジュールで利用予定の電力量が実際の発電量を超えると再スケジューリングが必要となることから、このような状況を平均的に回避するというより、むしろ高い確率で回避したい。このような制約条件は機会制約条件と呼ばれ、その評価に多くの計算時間を要する。本研究では、多目的であり機会制約条件を含むジョブスケジューリング問題に対して、効率的なアルゴリズムを提案する。

3 調達電力量と遅延時間に関する多目的ジョブスケジューリング問題

3-1 問題の定式化

本研究で対象とする分散データセンタのモデルについて図1に示す。各データセンタは、太陽光発電システムによる発電が可能であり、また、地理的に離れた場所に設置されている。データセンタには、複数のサーバが設置されており、ジョブの処理に利用される CPU コア数はサーバによって異なっている。図1の下部に示すように、ユーザが送信したジョブは各データセンタに割り振られ、CPU コアによって処理するスケジュールが作成される。ジョブを処理するためには、ジョブが要求する CPU コア数以上の CPU コアが、データセンタにおいて利用可能でなければならない。また、ユーザに指定された完了希望時刻以内にジョブを処理する必要がある。完了希望時刻を超えた遅延時間を最小にする必要がある。さらに、太陽光発電によって得られた電力を、ジョブの処理に活用することで、調達電力量を最小にすることも必要となる。

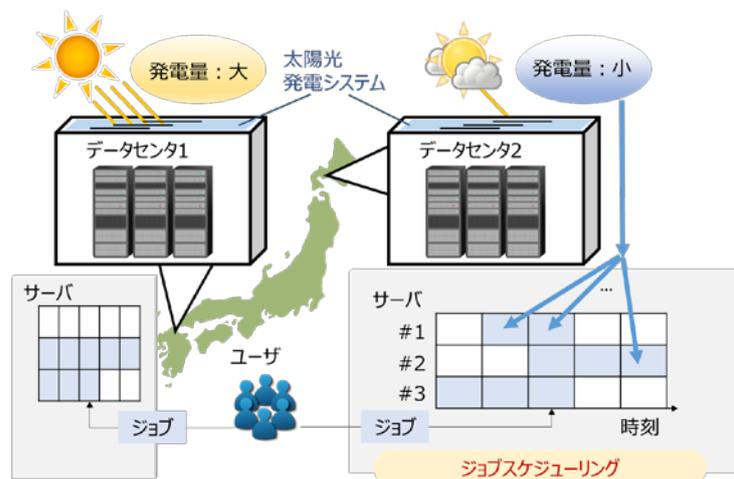


図1 分散データセンタモデル

そこで、遅延時間と調達電力量を最小にするためのジョブスケジューリング問題を、多目的最適化問題として定式化する。表 1 に示す決定変数ならびに表 2 に示すパラメータを用いた、本問題の定式化について以下に示す。

$$\min. f_1 = \sum_{j=1}^J v_j \quad (1)$$

$$\min. f_2 = \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^T (p_j - g_{jkt}) x_{jkt} \quad (2)$$

$$\text{s. t.} \quad \sum_{j=1}^J g_{jkt} x_{jkt} \leq G_{kt} \quad (3)$$

$$g_{jkt} x_{jkt} \leq P_j x_{jkt} \quad (4)$$

$$\sum_{j=1}^J R_{jkt} x_{jkt} \leq F_{kt} \quad (5)$$

$$\sum_{k=1}^K \sum_{t=1}^T x_{jkt} = L_j \quad (6)$$

$$\sum_{t=\tau}^{\tau+L_j} x_{jkt} \geq z_{jkt} L_j, \quad 1 \leq \tau \leq T - L_j \quad (7)$$

$$v_j \geq t z_{jkt} + L_j - 1 - d_j \quad (8)$$

$$v_j \geq 0, g_{jkt} \geq 0 \quad (9)$$

$$x_{jkt}, z_{jkt} \in \{0,1\} \quad (10)$$

表 1 決定変数

記号	説明
x_{jkt}	ジョブ j を時刻 t においてデータセンタ k で処理するとき $x_{jkt} = 1$
g_{jkt}	時刻 t 、データセンタ k におけるジョブ j の処理に割り当てる電力量
v_j	ジョブ j の完了希望時刻からの遅延時間
z_{jkt}	ジョブ j の処理が、時刻 t 、データセンタ k ではじまるとき $z_{jkt} = 1$

表 2 パラメータ

記号	説明
p_j	ジョブ j が単位時間に消費する電力量
G_{kt}	時刻 t 、データセンタ k において、太陽光発電で得られた電力量
R_j	ジョブ j に必要な CPU コア数
F_{kt}	時刻 t 、データセンタ k において利用可能な CPU コア数
L_j	ジョブ j の処理時間
d_j	ジョブ j の完了希望時刻

式(1)はジョブ j の遅延時間 v_j の総和を最小化する一つの目的関数であり、式(2)は各ジョブに必要な調達電力量 $(p_j - g_{jkt})$ の総和を最小化する二つの目的関数である。遅延時間 v_j は、式(8)において、ジョブの開始を表す z_{jkt} 、処理時間 L_j 、完了希望時刻 d_j によって定まる。式(3)は、割り当てられる電力量 g_{jkt} の総量が発電量 G_{kt} 以下であることを表す。式(4)は、ジョブ j が必要とする以上の電力量を割り当てることのできないことを表す。式(5)は、ジョブが使用する CPU コア数が利用可能な CPU コア数以下であることを表す。式(6)、(7)は、CPU コアに割り当てられたジョブは中断できないことを表す。

3-2 パレート最適化

本問題は多目的最適化問題であるが、トレードオフの関係にある二つの目的関数を同時に最小化する解は存在しない可能性が高い。このため、二つの目的関数の値が他のいずれの解にも劣らないような解

(パレート解)を発見することを本研究の目的とする。最小化問題において、解 X_1 がパレート最適であるということは、任意の解 X_2 と以下に示す関係が成り立つ必要がある。

$$X_1 \preceq X_2 \Leftrightarrow \begin{cases} f_u(X_1) \leq f_u(X_2) & \forall u \in \{1, \dots, U\} \\ \exists v \in \{1, \dots, U\} & f_v(X_1) < f_v(X_2) \end{cases}$$

多目的最適化問題を直接的に解くことは困難であるため、これまでヒューリスティクスにもとづく近似解法や、単一目的最適化問題に分解して解く方法が提案されている。本研究では、単一目的最適化問題に分解することで厳密解を求める Epsilon-constraint 法を適用する。

Epsilon-constraint 法では、ある目的関数以外を制約条件に置き換えることで単一目的最適化問題に変換し、これを繰り返し解くことによってパレート最適解を得る。本問題においては、式(2)の目的関数を以下の制約条件に置き換える。

$$f_2 = \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^T (p_j - g_{jkt}) x_{jkt} < \epsilon \quad (2)'$$

パラメータ ϵ の初期値は非常に大きな値とすることで、最初の単一目的最適化問題の解として、遅延時間を最小化するような解を得ることができる。その解に対する調達電力量 f_2 を次の ϵ として、再度、単一目的最適化問題を解くと、初期解と比較して、調達電力量が小さく遅延時間の大きい解を得ることができる。この手順を実行可能解が得られなくなるまで繰り返し、実行可能解からパレート最適解を得ることができる。

3-3 研究課題

前節で述べた Epsilon-constraint 法によるパレート最適化を適用する際の課題について述べる。

(1) 発電量に不確実性が存在する。

本問題における発電量 G_{kt} は計画段階の予測値であり不確実性を伴う。そこで、 G_{kt} の予測値が正規分布に従う確率変数として扱う。 G_{kt} が確率変数であるとき、解を一意に定めることができないため、 G_{kt} を含む式(3)を以下のような機会制約条件に変換して解く。

$$\Pr \left[\sum_{j=1}^J g_{jkt} x_{jkt} \leq G_{kt} \right] \geq \alpha \quad (3)'$$

$\Pr[\cdot]$ は不等式を満たす確率を示す関数であり、 α はその確率の下限値である。しかし、上記の制約条件を評価するためには、 G_{kt} からのサンプリングを伴うため、計算時間を要する。

(2) 整数計画問題に整数条件以外の非凸項が含まれる。

整数計画問題自体は非凸な整数条件を含むため、整数条件を連続条件に緩和した凸計画問題から下界値を得て、Branch-and-Cut 法を適用することが主要な解法となっている。しかし、本問題では非凸項 $g_{jkt} x_{jkt}$ を含んでいるため、整数条件を緩和しても非凸な計画問題であり、解くことが困難である。

(3) パレート最適解の探索効率が低い。

Epsilon-constraint 法は多目的最適化問題の効率的なアルゴリズムとして知られているが、パレート最適解ではなく、弱パレート最適解を導く問題を解いてしまうことがある。弱パレート最適解とは、パレート最適解と比較して、ある目的関数において劣る解を指す。2 目的最適化問題に Epsilon-constraint 法を適用した場合の、解の評価値の分布を図 2 に示す。

図 2 に示すように、Epsilon-constraint 法では、パレート最適解を得るまでに、弱パレート最適解を導く問題を多数解いているため非効率である。そこで、パレート最適解のみを導く問題を解くようにすることでアルゴリズムを効率化する。

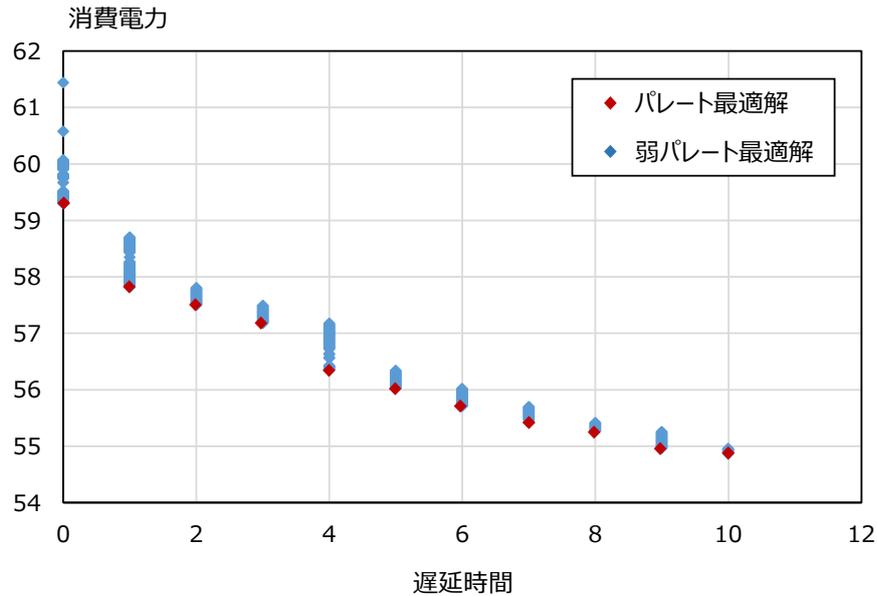


図2 Epsilon-constraint 法によって得られた解の評価値分布

4 機会制約条件つき多目的最適化アルゴリズム

4-1 アプローチ

前節で述べた課題に対して以下の方法を提案する.

(1) 機会制約条件の緩和

機会制約条件については, 決定性の制約条件に緩和する方法が知られており, 本研究では Sample Average Approximation 法(SAA) [15]を適用する. SAA によって機会制約条件(3)'は以下の制約条件に置き換えられる.

$$\sum_{j=1}^J g_{jkt} x_{jkt} - G_{kt} \leq w_s \pi \quad (\forall s \in \{1, \dots, S\})$$

$$\sum_{s=1}^S w_s \leq S(1 - \alpha)$$

$$w_s \in \{0, 1\}$$

π は非常に大きな値とする. w_s は元の制約条件(3)を満足するかどうかを0または1で表す決定変数である. $w_s = 0$ であれば元の制約条件を満たすが, $w_s = 1$ であれば満たさない. w_s の総和, すなわち制約条件を満たさない回数が $S(1 - \alpha)$ 以下になるような w_s を決定する. S は緩和した制約条件の評価回数であり, 大きいほど元の機会制約条件の良い近似となる.

決定変数 w_s の導入により問題の規模が拡大するものの, この制約条件は整数条件を除けば線形であり, 主要な整数計画法である Branch-and-Cut 法[16]の適用対象となる.

(2) 非凸項の置換

非凸項である $g_{jkt} x_{jkt}$ に対して $g_{jkt} x_{jkt} = y_{jkt}$ とおくと

$$g_{jkt} x_{jkt} = y_{jkt} = \begin{cases} g_{jkt} & (x_{jkt} = 1) \\ 0 & (x_{jkt} = 0) \end{cases}$$

となり y_{jkt} は離散変数となる. 離散条件を緩和して, y_{jkt} を $0 \leq y_{jkt} \leq g_{jkt}$ を満たす連続変数に変換すると, 線形緩和問題を得られるため, Branch-and-Cut 法によって元の整数計画問題を解くことができる. 得られた y_{jkt} にもとづいて x_{jkt} を以下のように決定する.

$$x_{jkt} = \begin{cases} 1 & (y_{jkt} > 0) \\ 0 & (y_{jkt} = 0) \end{cases}$$

(3) 弱パレート最適解の排除

制約条件(2)'から弱パレート最適解を導いてしまう原因として、微小な ϵ の更新を許してしまうことが挙げられる。そこで、 ϵ の更新をなるべく大きくするような Augmented Epsilon-constraint 法[17]を適用する。Augmented Epsilon-constraint 法では、原問題の f_1, f_2 に関する目的関数ならびに制約条件(2)'を以下のように変形する。

$$\begin{aligned} \min \quad & f_1 - \rho s \\ \text{s.t.} \quad & f_2 + s = \epsilon \end{aligned}$$

決定変数 $s(> 0)$ は f_2 の改善度を表すスラック変数であり、これを目的関数に含めることで、なるべく大きな s を決定することができる。 s が大きくなるにつれて、 ϵ の更新幅を大きくすることができる。また、 ρs が f_1 に対して十分に小さくなるように $\rho(> 0)$ を設定すれば、パレート最適解を求めることができる。

4-2 アルゴリズム

アルゴリズムの疑似コードを Algorithm 1 に示す。また疑似コード中の各関数について以下で述べる。

Algorithm 1 多目的最適化の疑似コード

Input: パラメータ集合 $\mathcal{P} \leftarrow \{p_j, G_{kt}, R_j F_{kt}, L_j, d_j\}, \forall j, k, t \in \mathbb{N}$

Output: パレート最適解の集合 $\Lambda \ni \{x_{jkt}, g_{jkt}\}, \forall i \in \mathbb{N}$

```

1: Initialize:  $i = 1, \Lambda = \{\emptyset\}, \epsilon \leftarrow +\infty$ 
2: while true do
3:    $P_\epsilon \leftarrow \text{Generate\_P}(\mathcal{P}, \epsilon)$ 
4:   if  $P_\epsilon$  is feasible then
5:      $y_{jkt}, g_{jkt} \leftarrow \text{Branch\_and\_Cut}(P_\epsilon)$ 
6:     if  $y_{jkt} > 0$  then
7:        $x_{jkt} = 1$ 
8:     else
9:        $x_{jkt} = 0$ 
10:    end if
11:     $\Lambda \leftarrow \Lambda \cup \{x_{jkt}, g_{jkt}\}$ 
12:     $\epsilon \leftarrow f_2(x_{jkt}, g_{jkt})$ 
13:  else
14:    break
15:  end if
16: end while
17: return  $\Lambda$ 

```

- **Generate_P(\mathcal{P}, ϵ)**
パラメータ \mathcal{P} と ϵ から Augmented Epsilon-constraint 法と SAA 法を適用した問題 P_ϵ を生成する。
- **Branch_and_Cut(P_ϵ)**
問題 P_ϵ に対して Branch-and-Cut 法を適用し、解 y_{jkt}, g_{jkt} を出力する。

Algorithm 1 に示すように、本アルゴリズムでは ϵ を更新しながら問題 P_ϵ を生成し、 P_ϵ が実行不可能となるまで P_ϵ を解いて、パレート最適解を集合 Λ に保存する。 y_{jkt} から x_{jkt} を求めたのち f_2 の値を求め、 ϵ を新しい f_2 で更新する。

5 評価実験

5-1 実験の概要

日本の5地点(札幌, 東京, 長野, 大阪, 福岡)にデータセンタが分散していると想定し, 実際にHPCサーバへ送信されたジョブを対象として実験を行った. 各地点の2016年8月4日の0時から8時までの日照量 I_{kt} [0.01MJ/m²]を用いて[18], 各データセンタの発電量 G_{kt} [kWh]の予測値 \hat{G}_{kt} を以下の式で計算した. 予測値 \hat{G}_{kt} を図3に示す.

$$\hat{G}_{kt} = 0.7 \times 0.002778 \times 5 \times I_{kt}$$

上式において, 変換ロスを考慮した係数を0.7, MJからkWhへの変換係数を0.002778, 太陽光パネルの出力が5kW(=250Wを20枚)と想定している. 図3に示すように, 4時ごろより一部の地域で発電がみられ, 曇天であった大阪や福岡の発電量は晴天の他地域に比べて少なくなっている. スケジュール作成時には, 予測値 \hat{G}_{kt} のみがわかっているとし, 実測値 G_{kt} は以下の正規分布に従うと仮定した. ただし, 正規分布からのサンプリング値が負になるときは0として扱っている.

$$G_{kt} \sim N(\hat{G}_{kt}, 0.1)$$

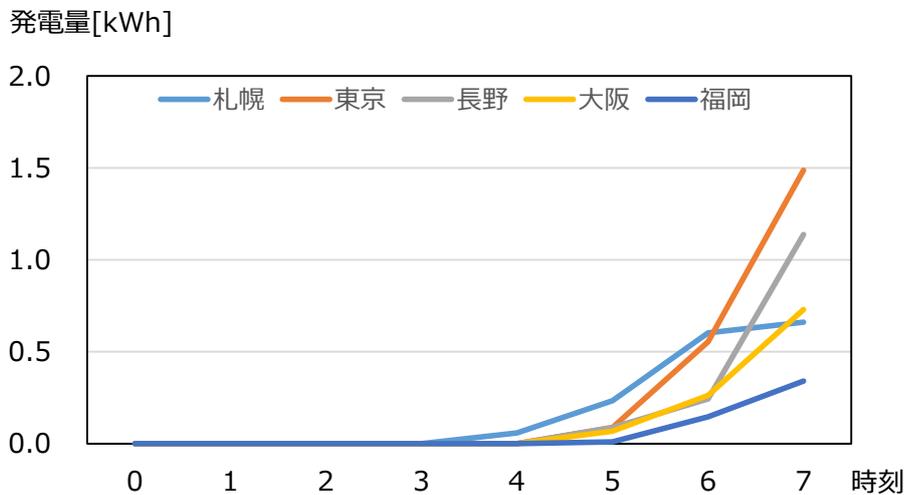


図3 各地点の発電量の予測値

また, 実際のデータセンタにおいて, 8時間に要求されるジョブ数の分布を調査し, 平均数15と90パーセンタイル値20をジョブ数として, スケジューリングを行った. 表3に実験に用いたジョブの必要コア数, 処理時間を示す. なお, 各ジョブの完了希望時刻は処理時間と同じ値に設定した. データセンタには, これらのジョブを処理するのに十分なCPUコアが備えられているものの, CPUコアにはすでにジョブの処理が割り当てられており, 全てのCPUコアを利用できるわけではない. 本実験では, 各データセンタにおいて利用可能なCPUコア数を表4に示すとおりとした. また, CPUコアがジョブを処理する際の電力は20[W]とした.

提案手法ならびに比較手法としてのEpsilon-constraint法を, 最適化ソフトウェアのGurobiを用いてpythonで実装した. 実験に利用した計算機の仕様は, OSがCentOS 7.2, CPUが2×Intel Xeon CPU E5-2650 V4@2.20GHz, メモリが8×8GBである. Epsilon制約である $f_2 < \epsilon$ を計算機で評価することは困難なので, 近似的に $f_2 \leq \epsilon - \sigma$ と置き換える. パラメータ σ に設定する値が小さいほど, 精度よくパレート最適解を求めることができるが, ϵ -制約法による f_2 の改善が小さくなり, より多くの計算時間を必要とする. 本実験では $\sigma = 0.01$ を用い, その他のパラメータについては, $\alpha = 0.9, S = 50, \rho = 0.01$ を用いた.

5-2 実験結果

各手法を適用した際の遅延時間と調達電力量の分布について, ジョブ数が15の場合の結果を図4に, ジョブ数が20の場合の結果を図5に示す. また, 各手法による計算時間の比較を表5に示す.

表3 スケジューリング対象のジョブ

(a) ジョブ数 15			(b) ジョブ数 20		
	CPU コア数	処理時間		CPU コア数	処理時間
#1	8	4	#1	8	4
#2	128	4	#2	128	4
#3	8	8	#3	8	8
#4	16	4	#4	16	4
#5	16	6	#5	16	6
#6	16	6	#6	16	6
#7	16	4	#7	16	4
#8	16	4	#8	16	4
#9	16	4	#9	16	4
#10	16	4	#10	16	4
#11	16	6	#11	16	6
#12	64	4	#12	64	4
#13	8	4	#13	8	4
#14	256	6	#14	256	6
#15	8	4	#15	8	4
			#16	8	4
			#17	8	4
			#18	8	4
			#19	8	4
			#20	8	6

表4 各データセンタの利用可能なCPUコア数

	札幌	東京	長野	大阪	福岡
利用可能なCPUコア数	128	256	128	64	64

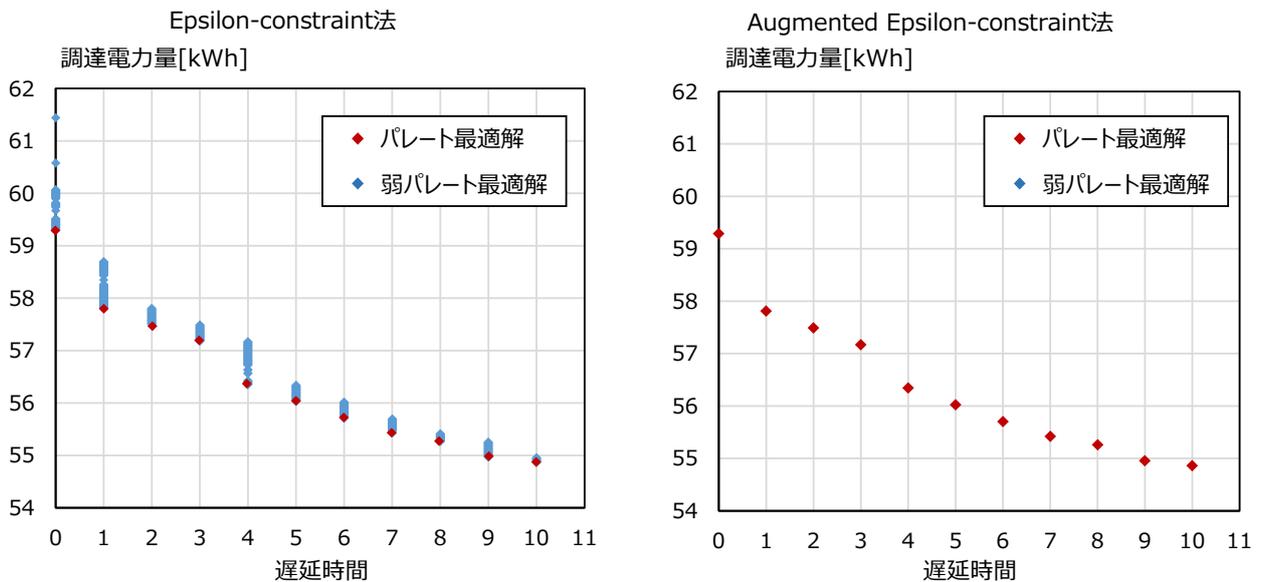


図4 解の評価値分布(ジョブ数15)

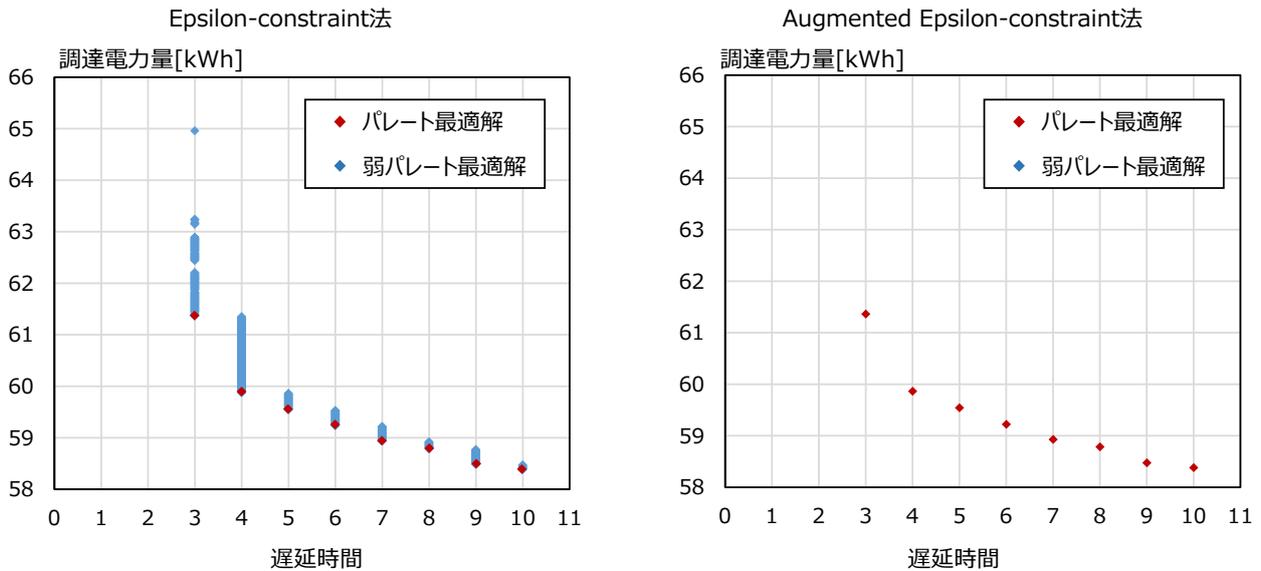


図5 解の評価値分布(ジョブ数 20)

表5 各手法の計算時間

	15 ジョブ	20 ジョブ
Epsilon-constraint 法	3 時間 56 分 48 秒	8 日 1 時間 7 分 7 秒
Augmented Epsilon-constraint 法	2 時間 36 分 55 秒	15 時間 36 分 8 秒

図4, 図5に示す結果から, Augmented Epsilon-constraint 法を用いることによって, 弱パレート最適解を求めることなく, パレート最適解を求めることができています. 結果として, 表5に示すように計算時間が短縮されている. 特に 20 ジョブを対象としたスケジューリング問題の場合, Epsilon-constraint 法では8日以上必要であるが, Augmented Epsilon-constraint 法では15時間まで短縮できている.

ジョブ数 15 と 20 の場合をそれぞれ比較すると, ジョブ数が 20 の方がより多くの CPU コアを利用することから, 調達電力量ならびに遅延時間が大きくなっている. またジョブ数が 15 の場合, 遅延時間が 10 のスケジュールの調達電力量は, 遅延時間が 0 のスケジュールと比較して約 4kWh 小さい. 一方, ジョブ数が 20 の場合, 遅延時間が 10 のスケジュールの調達電力量は, 遅延時間が 3 のスケジュールと比較して約 3kWh 小さい. 従って, ジョブ数が多い場合, 仮に遅延時間の増加を許したとしても, 調達電力量を十分に低減できない可能性がある. また, ジョブ数 15 と 20 の問題の双方において, 遅延時間を 10 より大きくしても, 調達電力量の小さいスケジュールを得ることはできなかった. これらの原因として, データセンタにおける発電量に限りがあるため, 発電量をほとんど使い切ってしまうことが挙げられる.

提案方式によって, 遅延時間や調達電力量のいずれかを最小化するスケジュールだけではなく, 双方を考慮したスケジュールを求めることができた. データセンタの管理者は, 提案方式を用いて得られたパレート最適解を閲覧し, 現状に適したパレート最適解を選択できる. いずれのジョブ数の問題においてもパレート最適解の数は 10 であり, データセンタの管理者が個々の解を評価可能な数であることを確認した.

6 結論

本研究では, 再生可能エネルギーを活用して, 調達電力量と遅延時間を最小化するジョブスケジューリング問題を対象とした. 不確実な発電量を確率変数で表現し, 機会制約条件を含む多目的最適化問題として, ジョブスケジューリング問題を定式化した. 大規模な問題であるため, 既存の多目的最適化手

法である Epsilon-constraint 法をそのまま適用した場合、スケジューリングに多大な計算時間を要してしまう課題があった。そこで、機会制約条件に対する SAA 法による近似、非凸項の変数変換、Augmented Epsilon-constraint 法による効率化を図り、スケジューリングに要する時間を削減する方法を提案した。データセンタにおける実際の発電量を想定して、ジョブ数 15 ならびに 20 のスケジューリング問題を作成し、既存の Epsilon-constraint 法と Augmented Epsilon-constraint 法を適用した。実験の結果、Augmented Epsilon-constraint 法によってパレート最適解を正しく求めつつ、計算時間を削減できることを確認した。特に、ジョブ数が多い場合において、計算時間を大きく削減できていた。パレート最適解を求めることで、遅延時間や調達電力量のいずれかを最小化するスケジュールだけではなく、双方を考慮したスケジュールを、データセンタの管理者に選択肢として提示することができた。

今後の課題として、多目的最適化手法のさらなる高速化を図る必要がある。本研究では、機会制約条件を SAA 法によって近似したが、正規分布のパーセンタイル値を用いることで、よりコンパクトな近似が可能となり計算時間をさらに削減できると考えている。また、発電量が昼間にピークを迎えるといった時系列的な傾向をアルゴリズムに反映させることで、高速化できると考えられる。例えば、ピークを過ぎると発電量が減少するだけでなく、遅い時間にジョブを割り当てることで遅延時間も増加してしまう。このため、時間帯によっては遅い時間にジョブを処理する利点が失われ、そのような割り当てを行うスケジュールを解の候補から除外できる可能性がある。このような性質を踏まえたヒューリスティックな解法を作成し、その解を暫定解とすることで、Branch-and-Cut 法の性能を向上できると考えられる。

【参考文献】

- [1] J. J. Rehr, F. D. Vila, J. P. Gardner, L. Svec, M. Prange. Scientific computing in the cloud. *Computing in Science Engineering*, Vol.12, No.3, pp.34-43, 2010.
- [2] S. Srirama, O. Batrashev, E. Vainikko. SciCloud: Scientific computing on the cloud. In *Proceedings of the IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp.579-580, 2010.
- [3] K. K. Nguyen, M. Cheriet, M. Lemay, M. Savoie, B. Ho. Powering a Data Center Network via Renewable Energy: A Green Testbed. *IEEE Internet Computing*, Vol.17, No.1, pp.40-49, 2013.
- [4] X. Tang, C. Chen, B. He. Green-aware Workload Scheduling in Geographically Distributed Data Centers. In *Proceedings of IEEE 4th International Conference on Cloud Computing Technology and Science*, pp.82-89, 2012.
- [5] I. Goiri, R. Beauchea, K. Le, T. D. Nguyen, M. E. Haque, J. Guitart, J. Torres, R. Bianchini. Greenslot: Scheduling energy consumption in green datacenters. In *Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis*, pp.1-11, 2011.
- [6] S. Iturriaga, S. Nasmachnow. Scheduling energy efficient data centers using renewable energy. *Electronics*, Vol.5, No.4. Article No.71, 2016.
- [7] H. Lei, R. Wang, T. Zhang, Y. Liu, Y. Zha. A multi-objective co-evolutionary algorithm for energy-efficient scheduling on a green data center. *Computers & Operations Research*, Vol.75, pp.103-117, 2016.
- [8] A.-A. Tantar, G. Danoy, P. Bouvry, S. U. Khan. Energy-Efficient Computing Using Agent-Based Multi-objective Dynamic Optimization. *Green IT: Technologies and Applications*, pp.267-287, 2011.
- [9] D. T. Luc, S. Schaible. Efficiency and Generalized Concavity. *Journal of Optimization Theory and Applications*, Vol.94, No.1, pp.147-153, 1997.
- [10] O. Mämmelä, M. Majanen, R. Basmadjian, H. De Meer, A. Giesler, W. Homberg. Energy-aware job scheduler for high-performance computing, *Computer Science - Research and Development*, Vol.27, No.4, pp.265-275, 2012.
- [11] K. Sato, M. Samejima, N. Komoda. Dynamic optimization of virtual machine placement by resource usage prediction. In *Proceedings of IEEE International Conference on Industrial Informatics*, pp. 86-91, 2013.

- [12] C. Gu, Z. Li, C. Liu, H. Huang. Planning for green cloud data centers using sustainable energy. In *Proceedings of IEEE Symposium on Computers and Communication*, pp. 804–809, 2016.
- [13] L. Yu, T. Jiang, Y. Cao, Q. Qi. Joint workload and battery scheduling with heterogeneous service delay guarantees for data center energy cost minimization. *IEEE Transactions on Parallel and Distributed Systems*. Vol.26, No.7, pp.1937–1947, 2015.
- [14] Y. Guo, Y. Gong, Y. Fang, P. P. Khargonekar, X. Geng. Energy and network aware workload management for sustainable data centers with thermal storage. *IEEE Transactions on Parallel and Distributed Systems*. Vol.25, No.8, pp.2030–2042, 2014.
- [15] B.K. Pagnoncelli, S Ahmed, A Shapiro. Sample average approximation method for chance constrained programming: theory and applications. *Journal of optimization theory and applications*, Vol.142, No.2, pp.399-416, 2009.
- [16] B. Korte, J. Vygen. *Combinatorial Optimization Theory and Algorithms*. Springer, 2012.
- [17] G. Mavrotas. Effective implementation of the ϵ –constraint method in multi-objective mathematical programming problems, *Applied Mathematics and Computation*, Vol.213, No.2, pp.455–465, 2009.
- [18] NEDO 日照量データベース, <http://app0.infoc.nedo.go.jp/metpv/metpv.html> (Accessed on 2017/6/30).

〈発 表 資 料〉

題 名	掲載誌・学会名等	発表年月
Epsilon Constraint 法を用いた分散データセンタにおけるジョブスケジューリング	電気学会 情報システム研究会, IS-16-008, pp. 41-46	2016年5月17日
Modeling Green-aware Job Scheduling Problem in Decentralized Data Centers	IIAI International Congress on Advanced Applied Informatics, pp. 1093-1096	2016年7月11日