

# 超解像のための機械学習プログラムのハイパーパラメータ最適化に対する 自動チューニングの適用

楊暄<sup>†1,a)</sup> 藤家空太郎<sup>†1</sup> 矢島雄河<sup>†1</sup> 秋田和俊<sup>†2</sup> 藤井昭宏<sup>†1</sup>  
田中輝雄<sup>†1</sup> 浮田宗伯<sup>†2</sup> 大島聡史<sup>†3</sup>

**概要**：機械学習プログラムでは、より良い学習をするためにハイパーパラメータを最適化する必要がある。この最適化を行うために、我々が開発を進めているソフトウェア自動チューニングツールを適用する。対象とするのは、低解像データを学習し超解像画像を得る機械学習プログラムである。このプログラムでは1回あたりの学習時間が長い。さらに、自動チューニングでは、この学習を繰り返し行う必要がある。そのために、2段階学習の手法により、1回あたりの学習時間を1/8とした。さらに、複数回の学習を効率的に行うためにGPUクラスタ上で複数ジョブの多重実行を実施した。その結果、GPUクラスタの並列環境を用いて、学習時間を延べ3470時間から262時間に削減することができた。

## 1. はじめに

一般のユーザが、複雑化するアーキテクチャを理解し、使用する計算機環境に合わせた性能チューニングを行うことは困難である。そこで、自動的に、ユーザプログラムの性能チューニングを行う手法として、ソフトウェア自動チューニングが提案されている[1]。ソフトウェア自動チューニングでは、対象とするユーザプログラム中の実行時間、消費電力などに影響を与える処理をパラメータ化し、そのパラメータの最適化を行なう。このパラメータを性能パラメータと呼ぶ。複数の性能パラメータを同時に推定する手法として、我々は性能パラメータ空間における反復次元探索法を提案している[2][5]。この手法をソフトウェア自動チューニングツール“DSICE”として実装した[4]。

DSICEは、各性能パラメータの探索範囲内の値を繰り返し取得し、性能パラメータの取りうる値の組み合わせを使用してユーザプログラムを実行して性能評価値を取得することにより、性能パラメータの最適な組み合わせを自動的に推定する。

今回適用するのは機械学習を利用した超解像プログラムである。機械学習のプログラムには、ロスウェイトや学

習率などの学習効果に影響する要素、いわゆるハイパーパラメータが存在する。ハイパーパラメータの組み合わせが適切ならば機械学習により有効なトレーニングモデルを得ることは効果的である。しかしながら、ハイパーパラメータの組み合わせが膨大に存在するために、適切な組み合わせを推定するには時間がかかる[5]。そこで、ソフトウェア自動チューニングツールDSICEを用いて、機械学習を利用した超解像プログラムの最適なハイパーパラメータの組み合わせを推定し、より有効なトレーニングモデルを得る。

本研究の目的は、このソフトウェア自動チューニングを実用的な時間で実現するために、2段階の学習方式およびスーパーコンピュータによる並列GPU環境の利用方式を提案し実現することである。

## 2. 自動チューニングにおけるパラメータ推定

### 2.1 自動チューニングの枠組み

ソフトウェア自動チューニング（以下、自動チューニング）の目的は、性能に影響を与えるパラメータ（以下、性能パラメータ）を調整することでプログラム性能を向上さ

†1 工学院大学  
a) em12027@ns.kogakuin.ac.jp

†2 豊田工業大学

†3 名古屋大学

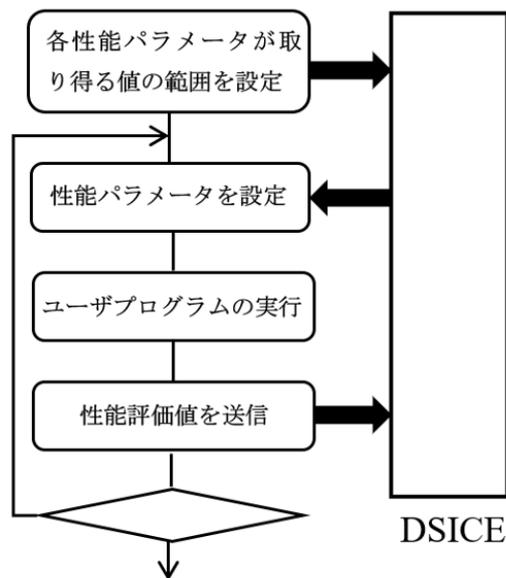


図 1 ユーザプログラムとソフトウェア自動チューニングツール DSICE の関係

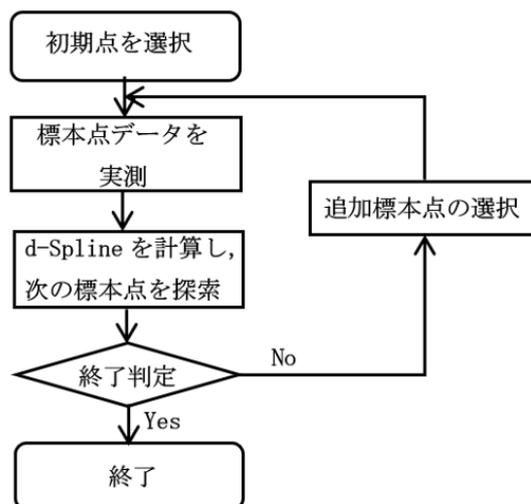


図 2 IPPE の最適値推定の手順

せることである。各性能パラメータは可能な値の範囲を設定し、自動チューニングツール DSICE は性能パラメータの適切な組み合わせを推定するように自動的に調整する。

各パラメータの組み合わせのユーザプログラムの性能を実際に測定することで最適な組み合わせを評価する。DSICE は、プログラムの実行中に性能パラメータを設定し、設定された性能パラメータの性能値を取得する。取得した性能評価値をもとに、適切な性能パラメータの組み合わせを推定する。

図 1 は、ユーザプログラムと自動チューニングツール DSICE との関係を示す。まず、ユーザプログラムで用いられる各性能パラメータの取りうる値を DSICE に設定する。次に、ユーザプログラムを繰り返し実行する。毎回のユーザプログラムの実行において、DSICE は新たな性能パラメータの組み合わせを設定する。ユーザプログラムの実行で得られた性能評価値は DSICE に送られる。DSICE では、それをそれまでに得られた性能評価値の情報に加味し、次の回に実行する性能パラメータの組み合わせを設定する。

## 2.2 反復次元探索

ソフトウェア自動チューニングツール DSICE の動作について説明する。DSICE は性能パラメータの取りうる値からなる空間上において、各空間上の点の値を性能評価値として、空間探索を行う。ここで、標本点逐次追加型性能パラメータ推定法 (Incremental Performance Parameter Estimation, IPPE) による次元探索と方向探索を繰り返す

手法を説明する[6]。ここで、標本点とは探索空間の中で DSICE によって設定された性能パラメータの取りうる値の点である。IPPE は、近似関数による最適値の推定に必要な標本点を、最低限の数から始め、新たな標本点を自動選択、追加しながら、近似関数を順次更新し、最適な性能パラメータを推定する手法である。図 2 に IPPE の最適値推定の手順を示す。

まず、初期点を決め、それを基準点とする。今回はパラメータ空間の中心点を最初の探索の基準点とする。方向探索と次元探索により、性能評価値がより良い点が新しい基準点に更新される。基準点が変わらなくなるまで方向探索と次元探索を繰り返す。

方向探索では、基準点に隣接する点を比較して探索方向を決定する。次に次元探索を行い、性能評価値が最適となる点を推定する。各探索において、一度探索した点は再度実測を行わない。次元探索では、近似関数としては、文献[6]で提案されているスプライン系関数 d-Spline を用いる。

図 3 に、反復次元探索を用いた方向探索の手順を示す。ここでは、性能パラメータの種類だけある次元空間の方向探索において、最初は、各性能パラメータのうち、ひとつだけ異なる方向 (つまり、各軸方向) のみを探索する。さらに、任意の 2 つの性能パラメータの値を同時に変化させる方向を追加し、この同時に変化させる性能パラメータの数を増やす。このように順次探索方向を増やしていくことで方向探索のコストを削減する。

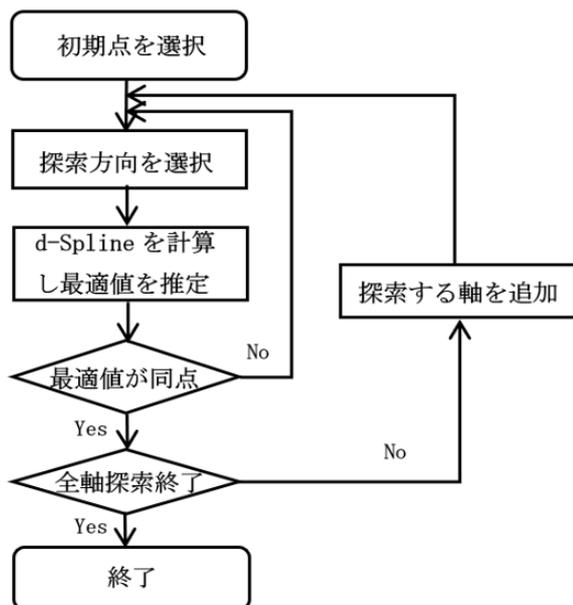


図3 反復次元探索を用いた方向探索の手順

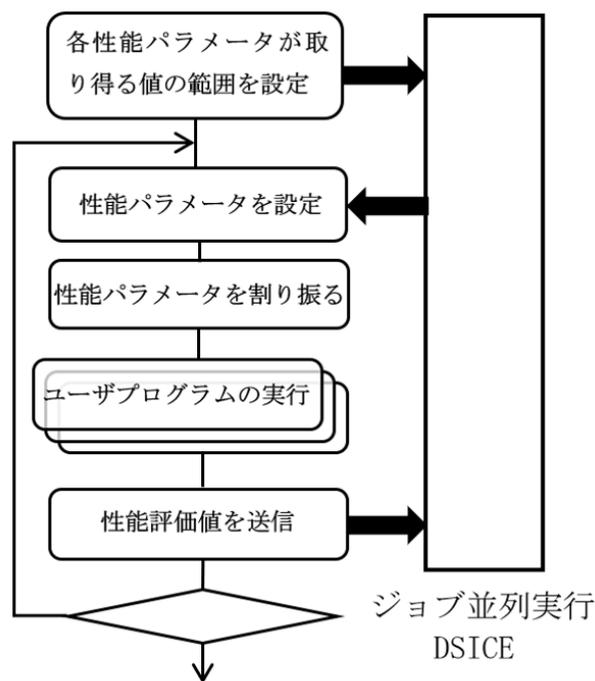


図4 ユーザプログラムとジョブ並列実行 DSICE の関係

### 3. 並列実行機能の DSICE への実装

スーパーコンピュータの高並列環境を用いて、ユーザプログラムの実行を並列化し、一度に複数の性能パラメータの組み合わせを探索することを実現する。

図4は、DSICEと連携して、複数のユーザプログラムを実行する関係を示す。ここでは、その時点の基準点から方向探索ならびに次元探索に必要な標本点の実測を並列に実行することとした。この並列性を実行するコンピュータで並列実行可能な数、またはそれ以上与えることにより、コンピュータの能力を十分に使いこなすことができる。

このような並列実行機構を DSICE に組み込んだ。

並列に実行されるユーザプログラムの結果としての性能評価値などの実測結果(ログ)はファイルを介して、DSICEに伝えられる。DSICEは並列に実行される複数の実行結果をまとめて最適な点を求め、次の基準点を計算する。

また、DSICEは実行したユーザプログラムのジョブ実行状況を把握している。何らかの理由で結果を出さずに途中で終了してしまったジョブについては、再度、そのジョブを投入することで、必要な情報(性能評価値)を得ることを可能とした。

なお、今回実現した並列実行管理は、コンピュータの並列実行可能数には依存せずに稼働させることができる。

### 4. 超解像の機械学習プログラムへ適用

#### 4.1 対象プログラム

超解像とは、低解像度画像を高解像度画像に変換する技術である[7]。本研究で用いた超解像プログラムは豊田工業大学の浮田宗伯教授のチームが開発した Dense Deep Back-Projection Networks(D-DBPN)である[8][9]。文献[8]では、低解像度画像の各辺を2倍、4倍、8倍の高解像度画像に拡大したが、本研究では4倍への拡大を対象とした。

DIV2K データセット[10]は、インターネットから収集された様々な2K解像度の画像を有する。現在普及しているデータセットよりもはるかに解像度が高い。

文献[8]の手法には、学習フェーズとテストフェーズの2段階がある。学習フェーズではエポック数を2000とし、DIV2Kの800枚の学習画像を学習し、良好なトレーニングモデルを得る。テストフェーズは学習フェーズで得たトレーニングモデルを採用し、DIV2Kの100枚のテスト画像を辺ごとに4倍に拡大するとともに、拡大後の画像の画質が優れていることを確保した。

画像の画質を Perceptual Score で評価する[11]。Perceptual Score は人が見て綺麗かどうかの指標である。Perceptual Score が小さいほど性能が良い。

本研究では、Perceptual Score の値から最適なハイパーパラメータの組み合わせを推定する。ハイパーパラメータと

表 1 事前学習, ファインチューニングで設定した  
 ハイパーパラメータの値

パラメータ	事前学習	ファイン チューニング
バッチサイズ	8	8
学習率	0.0001	0.0001
エポック数	1000	2000
w1	0.01	0.01
w2	0	0.1
w3	0	0.001
w4	0	0.001

して, MSE 損失ウェイト(w1), Perceptual 損失ウェイト(w2), Adversarial 損失ウェイト(w3), Style 損失ウェイト(w4), 学習率を対象とする.

## 4.2 事前学習とファインチューニング

文献[8]の D-DBPN のプログラムでは, 学習のエポック数は 2000 に設定されており, 1 回の実行には 92 時間かかる.

DSICE によるハイパーパラメータを探索する自動チューニングを行うためには, D-DBPN のプログラムを繰り返し実行する必要がある, 実行時間を大幅に減らす必要がある. そのため, 事前学習とファインチューニングからなる方法を用いた.

まず, 事前学習として, ハイパーパラメータの MSE 損失ウェイト(w1)のみを 0.01 とし, Perceptual 損失ウェイト(w2), Adversarial 損失ウェイト(w3), Style 損失ウェイト(w4)はすべて 0 とし, 学習を行う. バッチサイズは 8 とし, エポック数 1000 回の事前学習を行った. ここで得たトレーニングモデルをもとにして, エポック数を削減してファインチューニングを行う. このファインチューニングを, ハイパーパラメータのうち w1 を固定し, w2, w3, w4 および学習率の組み合わせを変えて何度も実行することで自動チューニングを行う.

ここで, 何度も実行するファインチューニングの実行回数を設定するために, w2 を 0.01, w3 を 0.001, w4 を 0.001 とし, エポック数を 2000 に設定してファインチューニングを 1 回実行した.

表 1 に, 今回設定した事前学習, 自動チューニング時の実行回数を決めるためのファインチューニングのハイパーパラメータの値を示す.

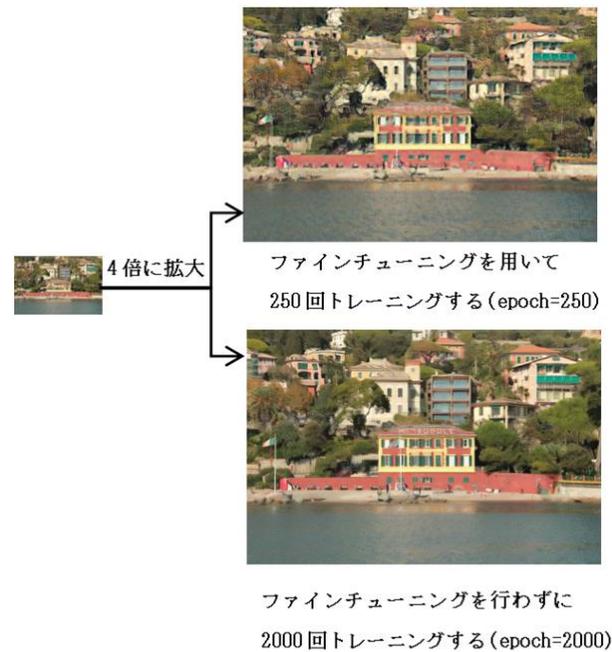


図 6 画像比較

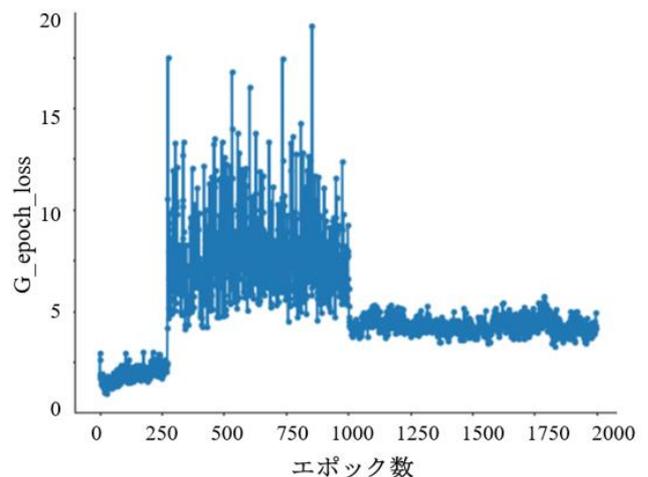


図 5 エポック回数に対応する G\_epoch\_loss の挙動

適切なファインチューニング回数を決定するために, エポック数に関する G\_epoch\_loss の挙動を調べる. D-DBPN が使用される敵対的生成ネットワーク (GAN) では, 偽のデータを生成する Generator (生成器) と, データが本物か偽物かを判別する Discriminator (識別器) を敵対させ, 競わせることで互いの性能が向上するように学習する. G\_epoch\_loss は, 毎回エポックの Generator Loss を示す.

G\_epoch\_loss の挙動が安定していることはトレーニング状況が良好であることを示している. 図 5 において, 横軸はトレーニング回数 (エポック数), 縦軸は対応する G\_epoch\_loss である. 250 回までは G\_epoch\_loss が安定し,

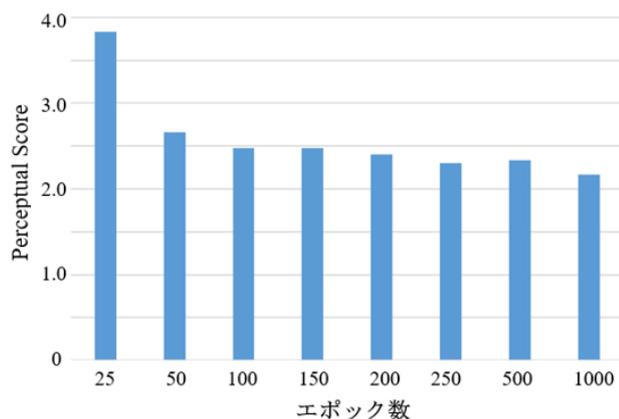


図7 ファインチューニングを行う場合のエポック番号とそれに対応する画像性能評価値

250 から 1000 の間で不安定となり、1000 回以降安定しているのは学習率を変えているためである。1000 回のトレーニングにかかる時間も長く、自動チューニングができなくなる。そこで、250 回トレーニングした後の出力画像の画質を検証してみる。

図6は、文献[8]で求められている最適なハイパーパラメータの組み合わせによるエポック数 2000 回トレーニングした画像と、事前学習後のトレーニングモデルにファインチューニングによりエポック数 250 回トレーニングした画像の画質を肉眼で比較したところ、250 回トレーニングした画像の画質は悪くなかったことを示している。

図7に、ファインチューニング時のエポック数ごと Perceptual Score 値を示す。横軸はエポック回数、縦軸は画像の画質を評価するための Perceptual Score である。エポック数 1000 回で得られる画像の画質が最も良く、次にエポック数 250 回の画像で良い値を取る。しかし、エポック数 1000 回では 45 時間かかる。一方、エポック数 250 回では 9 時間となる。ファインチューニングのエポック数 250 回のトレーニングで、自動チューニングを行なうことにした。

### 4.3 DSICE の適用

自動チューニングツール DSICE を用いて、超解像プログラムのトレーニング中にハイパーパラメータを最適化する。各ハイパーパラメータについて、取りうる値を複数設定する。初期ハイパーパラメータの組み合わせとして、各ハイパーパラメータの取りうる範囲の中央値を選択する。

ここで最適化の指標として、トレーニング結果の画像から計算される性能評価値(Perceptual Score 値)を用いる。DSICE では、取得された性能評価値に基づいて、1 次元探

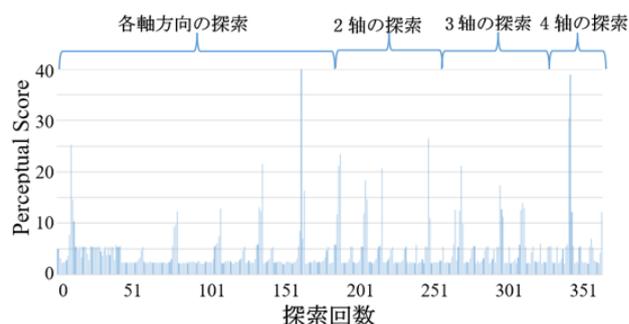


図8 Perceptual Score の推移

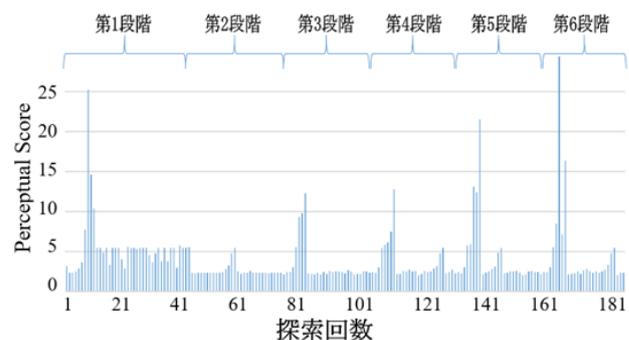


図9 各軸方向の探索の推移 (図8の一部拡大図)

索を繰り返すことにより、探索すべきハイパーパラメータの組み合わせを設定する。反復1次元探索が完了するまで、このプロセスが繰り返される。

本研究では、ハイパーパラメータとして、 $w_1$  を 0.01 に固定し、学習率、 $w_2$ ,  $w_3$ ,  $w_4$  の4種類を変化させる。ハイパーパラメータの組み合わせは合わせて 13,310 (=10x11x11x11) となる。実行環境には、名古屋大学のスーパーコンピュータ「不老」TypeII サブシステムを利用した。リソースグループは cx-share とする。cx-share は 1 ユーザあたり最大 50 のジョブを同時に実行できる。cx-share では、1 ジョブにつき 1 GPU を使うことができ、使用するノードはユーザ間で共有する。この手法を用いて「不老」TypeII サブシステムで複数のハイパーパラメータの組み合わせを並列実行する。

### 4.4 結果分析

DSICE で推定したハイパーパラメータにおける Perceptual Score の推移を図8に示す。横軸は探索回数、縦軸は探索したハイパーパラメータの組み合わせを用いて生成した画像の性能評価値(Perceptual Score 値)である。

図8によれば、DSICE は探索を完了するまで探索 374 回を行い、183 回までハイパーパラメータの各軸方向の探索を行う。つまり、ハイパーパラメータのうち、1 つの値のみを変化させる。探索 183 回目以降は、複数のハイパーパ

表 2 最適な評価値，一括学習で設定した  
ハイパーパラメータの値

パラメータ	最適な評価値	一括学習
バッチサイズ	8	1
学習率	0.0001	0.0001
エポック数	250	2000
w1	0.01	0.01
w2	50	0.1
w3	0.01	10
w4	0.01	10

ラメータを同時変化させた組み合わせの斜め方向にも拡大した。同時に変化させるハイパーパラメータは 2 組，3 組，4 組と順次増やしていく。最適なハイパーパラメータは 150 回目の探索の Perceptual Score 値が 2.00053 となった。文献[8]では最適な Perceptual Score 値は 2.04 であったので提案手法が 0.039 小さい値となった。DSICE は 13,310 種類のハイパーパラメータの組み合わせ問題について，363 回検索した。これは全体の 2.7%である。

図 9 に，図 8 の 1 回目から 183 回目までの各軸方向の探索部分を示す。この探索部分は第 1 段階から 6 段階までである。それぞれの段階で，推定点をもとに探索を行う。評価値 (Perceptual Score) が一番小さい値の組み合わせを次の段階の推定点とする。次の段階ではその推定点を中心に探索を続ける。第 6 段階では，評価値がさらに良い組合せが見つからなかったため，推定点は変化しなかった。そこで各軸方向の探索を終了し，2 軸の探索に移行する。

各段階は，その段階のすべての点を並列に実行する。たとえば，第 1 段階では 39 回，第 2 段階では 31 回が同時に並列実行することができる。

図 10 に自動チューニング時のジョブの実行状況を示す。横軸はジョブの番号，縦軸は，最初のジョブの開始時刻を 0 としたときの，それぞれのジョブの開始時刻と終了時刻の間を示す。探索が完了するまで 262 時間かかる。最初のジョブ 1 からジョブ 26 まだが同時に実行され，次にジョブ 27 からジョブ 38 まだが同時に実行された。これが図 9 の第 1 段階に対応する。

この 38 回のジョブは同時に実行されるはずである。しかし，「不老」Type II の運用で最大 26 に抑えられているため(マニュアル上では 50 並列まで可能となっていたが，混雑していたために同時実行は 26 に抑えられたとのことである)，2 回分の実行時間を要した。他のジョブ実行も同

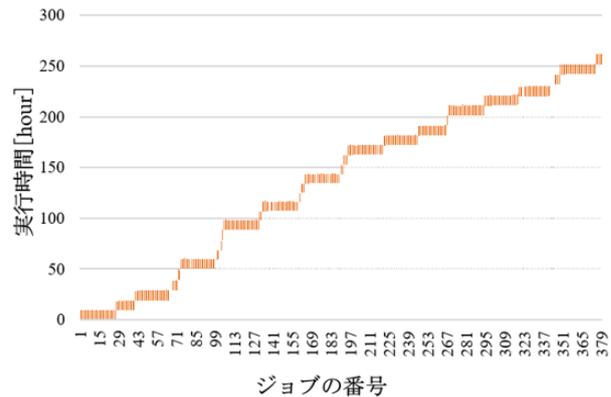


図 10 ジョブの実行状況

時実行数の最大は 26 となっており，残りは 26 回未満となっている。

また，数か所のジョブが並列に実行されていない。これは，ジョブが正常に終わらなかったときに再実行しているためである。並列実行を行い，実行時間を延べ 3470 時間から 262 時間へと約 13 倍高速化した。つまり，平均的な並列度は 13 となった。

## 5. まとめ

本研究では，超解像プログラム D-DBPN に対して並列化した自動チューニング機構 DSICE を適用し，ハイパーパラメータ推定を行なった。

この自動チューニングを実用的な時間で行うために，2 段階の学習方式およびスーパーコンピュータの並列 GPU 環境の利用方式を適用した。

まず，事前学習とファインチューニングで超解像 D-DBPN プログラムのトレーニング回数を減らす。事前学習を行い，事前学習モデルを得た。事前学習モデルを用いたファインチューニングは，結果画像の画質を保証した場合にトレーニング回数を大幅に減少させることができる。超解像プログラム D-DBPN では，プレトレーニングモデルを用いてファインチューニングを行うことで，トレーニングエポック数を 2000 から 250 に減らし，トレーニング時間も 1/8 に短縮した。

次に，並列化により方向探索および次元探索を並列化することで，探索を約 13 倍高速化した。組み合わせ 13,310 に対して約 2.7%の 363 回の探索で完了した。

自動チューニングの結果，最適なハイパーパラメータは 150 回目の探索の Perceptual Score 値が 2.00053 となった。文献[8]では最適な Perceptual Score 値は 2.04 であったので提案手法が 0.039 小さい値が得られた。

## 参考文献

- [1] 片桐孝洋, ソフトウェア自動チューニング-数値計算ソフトウェアへの適用とその可能性-, 慧文社 ISBN4-905849-18-7 (2004).
- [2] 多部田敏樹, 田中輝雄, 藤井昭宏, 関直人, 范谷瑛, “ソフトウェア自動チューニングにおける複数パラメータを同時推定する手法の効率化”, 情報処理学会第 81 回全国大会講演論文集, Vol.2019, No.1, pp.109-110 (2019).
- [3] 望月大義, 藤井昭宏, 田中輝雄, “ソフトウェア自動チューニングにおける複数同時性能パラメータ探索手法の提案と評価”, 情報処理学会論文誌, ACS-11, No2, pp.1-16 (2018).
- [4] N. Seki, T. Tabeta, A. Fujii, T. Tanaka, “Approach and Application for Performance Parameter Search on AT”, Poster Session on SC19 (2019).
- [5] 藤家空太郎, 多部田敏樹, 藤井昭宏, 田中輝雄, 加藤由花, 大島聡史, 片桐孝洋, “GPU クラスタを用いて並列化した自動チューニングの機械学習プログラムへの適用と安定性の検証”, 情報処理学会, HPC 研究会論文集, Vol.2020-HPC-178, pp.1-8 (2021).
- [6] T. Tanaka, R. Otsuka, A. Fujii, T. Katagiri, T. Imamura, “Implementation of d-Spline-based Incremental Performance Parameter Estimation Method with ppOpen-AT”, Scientific Programming, Vol. 22, pp. 299-307 (2014).
- [7] M. Irani and S. Peleg: Improving resolution by image registration, Graphical models and image processing (1991).
- [8] M. Haris, G. Shakhnarovich and N. Ukita, "Deep Back-Projection Networks for Super-Resolution," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018).
- [9] M. Haris, G. Shakhnarovich and N. Ukita, "Deep Back-ProjectiNetworks for Single Image Super-Resolution," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 12, pp. 4323-4337 (2021).
- [10] E. Agustsson and R. Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017).
- [11] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, “The 2018 PIRM challenge on perceptual image super-resolution,” Proceedings of the European Conference on Computer Vision (ECCV) (2018).