様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成２５年　５月２４日現在

機関番号：１７１０２
研究種目：基盤研究（C）
研究期間：2010～2012
課題番号：２２５５００１５
研究課題名（和文）超並列フラグメント分子軌道法プログラムライブラリの開発

研究課題名（英文）　Development of the library programs for massively parallel calculation of fragment molecular orbital method

研究代表者
稲富　雄一（INADOMI YUICHI）
九州大学・情報基盤研究開発センター・学術研究員
研究者番号：７０４３７７４７

研究成果の概要（和文）：
効率的な超並列実行により高速にフラグメント分子軌道法に基づいた大規模生体分子に対する FMO 計算を行うために，超並列 FMO プログラム OpenFMO の開発を行った．また，最適化において，計算機科学関連の研究者，技術者の協力が得やすいように，FMO プログラムをモジュール化して，各モジュールの接続部分を設計した．その結果，数万並列でも効率よく並列実行できるプログラムが完成した．また，作成したモジュールを利用することで，効率よく並列 FMO プログラムの開発が可能になることが示された．

研究成果の概要（英文）：
To carry out the effective massively parallel calculation of the fragment molecular orbital (FMO) method, we developed the parallel FMO program, OpenFMO. In addition, we probed the hierarchical structure of the FMO program and designed the interface of subprograms (modules) to make it easy to cooperate with the computer scientists and computer engineers to improve the performance of the parallel FMO program. In this study, we implemented the high performance FMO program and confirmed its high parallelization efficiency. And it was shown that the modules designed and implemented in this work was useful to construct the parallel FMO program in short time.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
２０１０年度	1,800,000	540,000	2,340,000
２０１１年度	1,100,000	330,000	1,430,000
２０１２年度	800,000	240,000	1,040,000
年度			
年度			
総　計	3,700,000	1,110,000	4,810,000

研究分野：化学
科研費の分科・細目：基礎化学・物理化学
キーワード：フラグメント分子軌道法，超並列化

１．研究開始当初の背景

(1)フラグメント分子軌道（FMO）法はたんぱく質などの巨大生体分子に対する量子力学に基づいた第一原理電子状態計算を高速に行うために開発された並列処理向きの計算手法である．最先端の高性能計算機（スーパーコンピュータ，スパコン）を用いることで，生体分子と薬物候補分子との相互作用計

算を FMO 法で高精度，かつ，高速に計算することが可能になれば，計算機支援による創薬を，これまでの古典的，経験的な評価関数を用いる場合よりも格段に効率よく行うことができると考えられる．

(2)FMO 法では，巨大分子に対する計算を多数の小規模な計算に分割して行うため，並列処理向きの計算手法であり，我々が開発していた並列 FMO プログラム OpenFMO だけではなく，いくつかの並列電子状態計算プログラムに実装され，その並列性能の高さが実証されてきた．しかしながら，当時開発中であった京コンピュータをはじめ，現在のスパコンは数万～数 10 万プロセッサを搭載した超並列計算機であり，既存コードでそのような超並列実行を効率的に行うことは並列化効率の観点から困難であり，並列化効率向上のための最適化が必須である．ただし，プログラムの超並列化に向けた最適化は，計算機のアーキテクチャを考慮しなければ困難であり，そのためには計算化学者だけでなく，計算機科学の研究者との連携が必須である．そのような連携を行いやすくするためには，作成したプログラム（ソースコード），ならびに，プログラムを構成するサブプログラム群（ライブラリ）の結合様式（API）を公開することが必要である．しかし，研究開発当初の時期は，ソースコードが公開されていても巨大で複雑であったり，有償で，かつ，複雑なライセンス契約があるため，ソースコードの改変が困難であったりと，他分野の研究者の協力を得にくい状況であった．

２．研究の目的

最先端スパコンを用いた数万～数10 万プロセッサでの効率的並列計算（超並列計算）を可能とし，たんぱく質などの巨大生体分子の高速電子状態計算を行うFMOプログラム開発を目覚ましく効率化するために，FMO法で用いる様々なサブプログラムのインターフェイスを設計しそれを公開すること，ならびに，大規模分子に対する超並列計算が可能な高性能 FMO 法プログラムの開発を促進するために，FMO計算用のライブラリを開発して，それを公開すること，を本研究での研究目標とした．

３．研究の方法

(1)並列 FMO プログラムで用いる各モジュールの API 設計

我々が，これまでに作成してきた FMO 法プログラム OpenFMO の解析を行い，それを，いくつかの階層に分類する．この段階で，超並列化に向けて修正すべき点があったら，その階層構造やプログラムのモジュールのつなぎ目部分を変更する．この作業を行った後，FMO 法で用いる各種モジュールの API を設計する．

(2)分子積分プログラムのコード開発とハイブリッド並列化

FMO 計算では，従来の分子軌道計算の場合と同様に，計算時間のほとんどを，2 電子積分などの分子積分計算に費やしているため，この部分の並列処理による高速化が必須となる．そこで，並列化を考慮した分子積分計算のコード開発を行い，そのコードをもとに，超並列処理に向けたプロセス並列とスレッド並列を併用したハイブリッド並列化を行う．

(3)大規模分子の計算を可能にするためのデータの分散保存，および，アクセス方法の検討

FMO 法では，計算を行う際に，モノマー密度行列と呼ばれる中間データを保存して，そのデータに対する参照・更新を多数回行う必要がある．この中間データは入力として与えられる計算対象分子の大きさに比例して増加するが，将来，数万～10 万フラグメントの分子を扱う場合には，そのデータサイズが数10GB になると想定されることから，複数の計算ノードで分散保存する必要がある．そこで，大規模分子への対応，および，並列性能の向上の点から，並列 FMO 計算における中間データの分散保存の方法と，それに対するアクセス方法の検討を検討した．

４．研究成果

(1)ソースコードの公開

開発した並列 FMO プログラム OpenFMO のソースコードを，現在，文部科学省が行っている調査研究「アプリケーション分野からみた将来の HPCI システムのあり方の調査研究」において収集している，スパコンで動作させることを想定したベンチマークアプリケーションプログラム（フルアプリ）として，登録を行った．登録したソースコードは，近日中に，次世代のスパコン開発を行っている計算機科学者，あるいは，システム開発者に向けて公開が予定されている．したがって，FMO 計算プログラムの最適化などに対する計算機科学分野の研究者の協力を得やすくなると考えている．

(2)ハイブリッド並列分子積分プログラムの作成とその性能評価

分子積分計算は，FMO 計算の全計算時間の99%以上を占めている主要部分である．FMO 法に現れる分子積分には，通常の分子軌道計算に現れる 2 電子積分（電子反発積分）や 1 電子積分（重なり積分，運動エネルギー積分，核-引力積分）だけではなく，4 中心，3 中心，

および，2 中心のクーロン相互作用積分がある．超並列化に向けたハイブリッド並列化を考慮してすべての分子積分に対する逐次コードを作成して，それらに対してプロセス並列化には Message Passing Interface（MPI）を，スレッド並列化には OpenMP をそれぞれ用いた MPI/OpenMP ハイブリッド並列化を適用した. 今回用いた MPI や OpenMP は，並列化における事実上の業界標準であり，ほぼすべての並列計算機に導入されている．また，MPI/OpenMP ハイブリッド並列化は，京コンピュータ上のアプリケーションプログラム開発における推奨プログラミングモデルである．

ところで，並列処理を効率的に行うためには，並列処理を行っている各プロセスに割り当てられる計算量を均一にして，全プロセスの計算終了時間を揃える（負荷均等化を図る）必要がある．分子積分計算では計算精度を落とさずに計算量を削減するために，カットオフ処理を行っているが，そうすると，各プロセスへの均等な計算割り当てを事前に行うことが困難である．このように，事前を負荷割り当てが困難な場合に負荷均等化を図る手段としては，グローバルカウンタ（GC）を用いた動的負荷分散が有効であり，GAMESS などの既存の並列量子化学計算プログラムでの利用されている．GC は，すべてのプロセスから排他的に参照，および，更新が可能なカウンタである．各プロセスは，GC への排他的なアクセスによりカウンタ値の取得，および，更新処理を行い，得られたカウンタ値をもとに処理を行う．GAMESS や NWChem といった並列量子化学プログラムでは，GC の実装に ARMCI というライブラリを用いている．しかし，現在の京コンピュータでの ARMCI の実装では，京コンピュータを構成する各ノードに搭載された 8 個のプロセッサコアのうち，1 コアを ARMCI 専用に用いているため，各ノードあたり 7 コアしか計算に利用できない．本プロジェクトで改良している OpenFMO プログラムは汎用並列計算機向けに開発しているものではあるが，京コンピュータでの動作も想定している．したがって，京コンピュータ上では

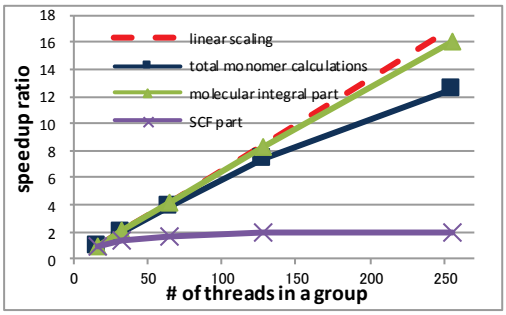

図 1 ：モノマーSCF 計算の並列性能

ARMCI を利用した場合の性能への影響が大きいために，今回はその利用を断念した．そこで本研究では，GC を MPI と OpenMP を用いて実装した．本実装では，並列処理を行っている全スレッドのうち 1 つが GC 専用で動作し，他のすべてのスレッドを計算に利用できるようになっている．このように実装した GC とハイブリッド並列化した分子積分コードを用いて，並列実行性能を評価した．

評価環境は，計算機として九州大学情報基盤研究開発センターの富士通 PRIMERGY CX400（8-core Xeon(E5-2680, 2.70GHz)x2/ノード，1476 ノード，インターコネクト InfiniBand(FDR)）を，コンパイラは富士通 C コンパイラ（バージョン 1.2.0），MPI は富士通 MPI（バージョン 1.2.0）,また，数値演算ライブラリとして SSL-II を利用した．入力データは，対象分子としてアデノウィルス KNOB ドメイン（PDB ID=1NOB），基底関数として，分子軌道計算で頻繁に用いられている 6-31G*とした．FMO 計算で行うモノマーSCF 計算と呼ばれる計算部分における分子積分の計算時間を，並列計算に用いるスレッド数を変化させて測定して，その並列性能を求めた．その結果を図 1 に示す．この図はモノマーSCF 計算部分の並列性能を示したもので，横軸は並列計算に利用したスレッド数（並列度），縦軸は，速度向上比をそれぞれ表しており，赤い破線が理想的な速度向上比である．また，この図には，分子積分部分だけでなく，モノマーSCF 計算で行う SCF 処理部分，および，モノマーSCF 計算全体の速度向上比も記されている．まず，分子積分部分は並列化効率が非常に高く，256 並列で 94%という高い並列化効率を保っていた．したがって，ハイブリッド並列化，および，GC を用いた負荷均等化によって，数 100～1000 並列でも効率的に並列処理ができることがわかった．一方で，SCF 計算部分の並列性能の悪さから，並列度が大きくなるとモノマー電子状態計算全体での効率が低下することが明らかとなった．これは，一般化固有値問題求解（対角化）のように，複数プロセスを用いた並列処理が困難な部分が SCF 計算部分に存在するためである．今後，この部分を対角化以外の方法に置き換える，あるいは，他の処理と並列に実行して，処理時間を隠蔽する，などの工夫を行って，更なる並列処理性能向上を行うことが必要である．

(3)データ分散保存，アクセス方法の性能評価

FMO 計算を行う際には，その中間データとしてモノマー密度行列データ（密度行列データ）を保存して，そのデータに対する参照・更新処理を行う．このデータは，計算を行っているすべてのプロセスからアクセス可能である必要がある．入力の大きさに比例して

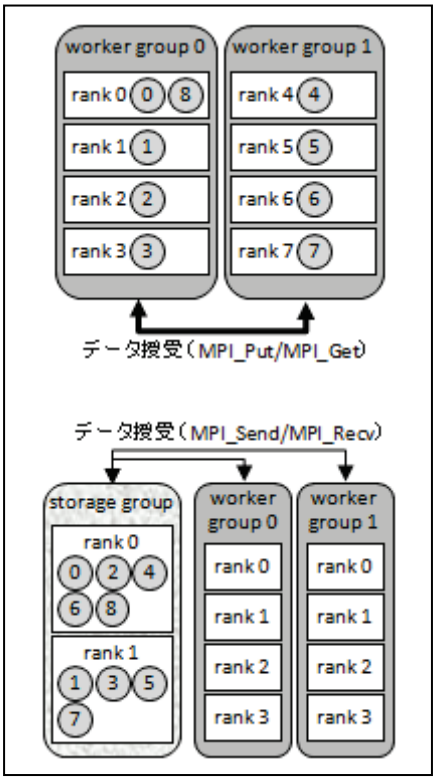

図２：2 つ密度行列データの保存・アクセス方法
（計算プロセス数＝８，モノマー数９の場合）
（上）片側通信を用いる手法（実装１）
（下）データ保存専用プロセスを用いる手法（実装２，データ保存プロセス数＝２）

密度行列データも増加するため，利用する計算機の規模に応じて大規模な入力に対応するためには，密度行列データを分散保存して，それに対して効率よくアクセスすることが必要となる．今回我々は，そのような分散保存・アクセス方法として，２つの実装を行い，その性能を評価した．１つは，データを全プロセスに分散保存して MPI-2 規格に実装されている片側通信機能を利用してアクセスする方法（実装 1），もう１つは，データ保存専用プロセスを導入して MPI-1 規格にもある１対１通信機能を用いる方法（実装 2）である（図２参照）．

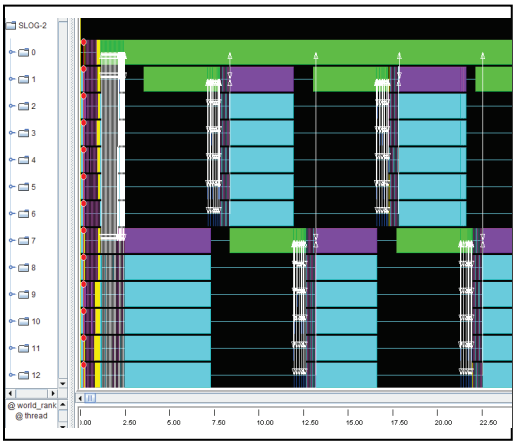

図３：実装１における MPI プロファイル

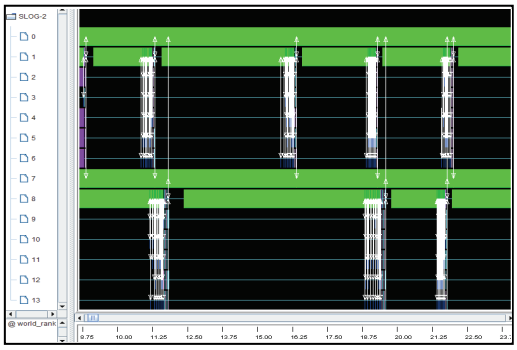

図４：実装２における MPI プロファイル

実装１，実装２を用いて FMO 計算を行った場合の MPI プロファイルを，それぞれ，図３，図４に示す．この図は， MPI Parallel Environment (MPE)環境を用いて MPI の実行プロファイルを取得して，MPE に付属するビューワ Jumpshot で表示した結果である．横軸が経過時間，縦軸がプロセスの rank 番号であり，黒い部分は，通信以外の処理（計算など）を行っている部分であり，色のついた帯で示した部分，あるいは，縦方向の線は，通信や同期待ちなど，計算以外の処理を行っている部分を表している．どちらの実装も，rank=0 のプロセスが並列 FMO 計算全体を制御するマスタプロセスとして動作していて計算を行っていないので，全時間帯で計算を行っているプロセスからの MPI_Recv のよるジョブ要求待ち（緑色の帯）であることが分かる．

まず，片側通信を利用した実装１のプロファイル（図３）を見ると，色のついた帯が多く見られていることから，計算を行わずに通信待ちをしている箇所が多く存在し，処理効率が大きく低下していることが分かる．これは，今回利用した MPI ライブラリにおける片側通信が，アクセス対象となるデータを保持しているプロセスが何らかの MPI 関数を呼び出すまで，他のプロセスからの片側通信要求に応答しない，という実装になっているからである．京コンピュータに実装されている MPI ライブラリも同様の実装が行われているので，効率的な超並列 FMO 計算の実装には，MPI-2 の片側通信機能を使うことを断念した．

一方，実装２を用いた場合の MPI プロファイル（図４）を見ると，通信による待ち時間がほとんどないことが分かる．この図の rank=7 のプロセスがデータを保存する専用プロセスとして動作しており，計算を行っているプロセスからのデータアクセス待ちを行っている．今回の結果は，データ保存専用プロセスを用いることで，密度行列データの参照・更新処理が効率的に行うことができることを示している．これら，２つの実装の性能評価結果から，並列 FMO 計算プログラム

OpenFMO における密度行列データの保存，アクセス方法として，データ保存専用プロセスを導入した実装２を用いることにした．

(4)大規模並列処理時の並列性能評価

ハイブリッド並列化した分子積分プログラム，ならびに，密度行列データ保存方法を組み合わせて，FMO 計算全体を行うプログラムを作成して，大規模な並列性能評価を行った．その結果を図５に示す．用いた計算機は，前出の富士通 PRIMERGY CX400 で，計算に利用するコア数を 1024～20480 に変化させて，並列性能を評価した．入力としては，前出のアデノウィルス KNOB ドメインで，基底関数として 6-31G*を用い，FMO 計算を通して行い，FMO 計算全体と，それを，モノマー電子状態計算部分（図中では SCC），ダイマーSCF 計算部分（図中では dimer SCF），近似ダイマー計算（図中の ES dimer）の 3 つに分割して，FMO 計算全体，および，それぞれの部分の計算時間の変化を測定した．この図の横軸は計算に用いるスレッド数（計算に用いるコア数と等価），縦軸は 1024 並列時の性能を 1 とした場合の速度向上比で，赤色の破線は理想的な速度向上比を表している．この結果を見ると，今回作成した並列 FMO プログラムの並列性能が非常に高いことが分かる．10240 並列までは，ほぼ理想的な速度向上比を示しており，20480 並列時でも 90%以上の並列化効率を示している．詳細に見ると，ダイマーSCF 計算と近似ダイマー計算の並列化効率は非常に高く，20480 並列でも，全く並列性能が低下していない．一方で，モノマーSCF 計算部分では 10240 並列までは，ほとんど理想的な速度向上を示しているが，20480 並列になると，大きく並列性能が低下している．これは，用いた計算機の規模に対して，入力データが小さかったために，負荷均等化が困難になった影響であると考えられる．この性能低下は入力規模が大きくなると解消されることから，数万フラグメントの大規模入力の計算に対しては，現状のプログラムでも効率的な超並列 FMO 計算が可能である．また，FMO 計算全体の時間を見ると，20480 並列時の計算時間は 30 分未満と，非常に短いことが分かった．今回用いた入力は 16000 原子を越える中程度の規模のたんぱく質であり，この規模の分子であれば，ごく短時間で FMO 計算が行えることが示された．さらに並列性能を向上させることで，この程度の大きさを持つ生体分子に対する FMO 計算が 10 分未満で計算できる可能性が見えた．このことは，近い将来に，多数の候補化合物と生体高分子との相互作用計算を，短時間に，かつ，大量に行うことにより，創薬分野への応用が可能になることを示している．

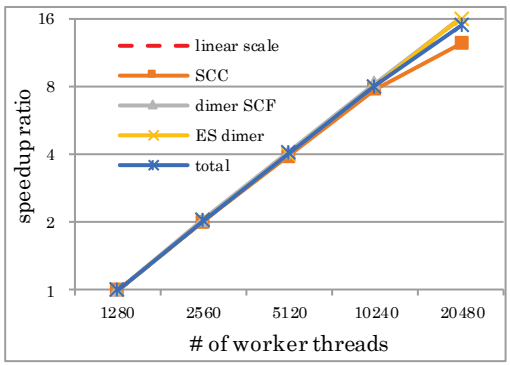

図５：大規模並列実行時の性能

(5)API 設計によるコード作成の効率化の確認

本研究では，FMO 計算プログラムを階層化，ならびに，モジュール化して API を決定し，その階層性，API を持つモジュールを作成した．そこで，今回作成したモジュールを用いて，OpenFMO とは構造が異なるマスタ・ワーカ型の FMO プログラムを，再構成した．コードは，全体を制御するマスタプログラムと，計算を行うワーカプログラム，および，データ保存を専門に行うメモリサーバプログラムが，全く別のプログラムとして動作するようになっており，MPI-2 の動的プロセス生成，および，プログラム間通信機能を用いて，マスタ・ワーカ型のプログラムになっている．

プログラムの構造が従来の並列 FMO プログラムと大幅に異なっているため，最初からコードの記述を行った．ただし，FMO 計算で使用する各サブプログラム部分には，これまで作成したモジュールを組み込むことで，コード作成の効率化を図った．

その結果，マスタ・ワーカ型の FMO プログラムの開発が，約 2 ヶ月という短期間で完了した．このことにより，サブプログラムをモジュール化して，それを再利用することで，プログラム作成が効率化されることを示された．

５．主な発表論文等

（研究代表者，研究分担者及び連携研究者には下線）

〔雑誌論文〕（計　1 件）

1. Yuichi INADOMI, Jun MAKI, Hiroaki HONDA, Toshiya TAKAMI, Taizo KOBAYASHI, Mutsumi AOYAGI and Kazuo MINAMI, "Performance Tuning of Parallel Fragment Molecular Orbital Program (OpenFMO) for Effective Execution on K-computer", Journal of Computer Chemstry, Japan, 査読あり, accepted.

〔学会発表〕（計 13 件）

1. 稲富雄一，眞木淳，本田宏明，西田晃，高見利也，小林泰三，青柳睦，南一生「FMO 法の超並列化への取り組み」第 4 回分子科学総合討論会 2010 大阪，ポスター発表 2P095，2010.9.15，大阪大学豊中キャンパス
2. 稲富雄一，眞木淳，本田宏明，高見利也，小林泰三，西田晃，青柳睦，南一生，「並列 FMO プログラム OpenFMO の超並列化への取り組み」，第 5 回分子科学討論会 2011 札幌，ポスター発表 1P108，2011.9.20～23，札幌コンベンションセンター
3. 稲富雄一，眞木淳，高見利也，小林泰三，青柳睦，南一生，「並列 FMO プログラム OpenFMO の性能最適化」HPCS2012，ポスター発表 P1-3，2012.1.25，名古屋大学
4. Yuichi Inadomi, "Optimization of FMO Code (OpenFMO) for Effective Massively Parallel Execution", poster P02, The 4th French-Japanese Workshop on Computational Methods in Chemistry,2012.3.5 ～ 6, Fukuoka, Japan
5. 稲富雄一，眞木淳，高見利也，本田宏明，小林泰三，南里豪志，青柳睦，南一生，「並列 FMO プログラム OpenFMO の性能最適化」，情報処理学会研究報告 2012-HPC-133，口頭発表，2012.3.26～27，有馬温泉
6. Yuichi Inadomi, Jun Maki, Hiroaki Honda, Toshiya Takami, Takeshi Nanri, Mutsumi Aoyagi and Kazuo Minami, "Performance Tuning of FMO Code (OpenFMO) for Effective Massively Parallel Execution", poster, Theory and Applications of Computational Chemistry(TACC-2012), 2-7, Sep. 2012, Pavia, Italy.
7. Yuichi Inadomi, "Performance improvement of FMO program for effective massively parallel execution on K-computer", Invited Oral, International Workshop on Massively Parallel Programming Now in Molecular Science, 28, Jan. 2013, Waseda University, Japan.
8. 村井均，南一生，横川三津夫，梅田宏明，佐藤三久，辻美和子，稲富雄一，青柳睦，中島真，「スーパーコンピュータ「京」におけるマスタ・ワーカ型プログラミングモデルの検討」，情報処理学会研究報告，Vol.2013-HPC-138(No.26) 福井

〔その他〕
ホームページ等
http://www.openfmo.org/OpenFMO/index.html

６．研究組織
(1)研究代表者
稲富　雄一（INADOMI YUICHI）
九州大学・情報基盤研究開発センター・学術研究員
研究者番号：70437747

(2)研究分担者
青柳　睦（AOYAGI MUTSUMI）
九州大学・情報基盤研究開発センター・教授
研究者番号：00260026

高見　利也（TAKAMI TOSHIYA）
九州大学・情報基盤研究開発センター・准教授
研究者番号：10270472

眞木　淳（MAKI JUN）
（財）九州先端科学技術研究所・研究員
研究者番号：10404047

小林　泰三（KOBAYASHI TAIZO）
九州大学・情報基盤研究開発センター・学術研究員
研究者番号：20467880