

NUOPT/DFO 利用ガイド

株式会社 数理システム

Phone: 03-3358-1701

Fax: 03-3358-1727

Email: nuopt-support@msi.co.jp

2010/02/22 更新

目次

| | |
|-----------------------------------|----|
| 1. はじめに | 3 |
| 2. NUOPT/DFO をご利用になる前に | 4 |
| 2.1. NUOPT のバージョンとモジュールの確認 | 4 |
| 2.2. パスの設定 | 4 |
| 3. NUOPT/DFO の実行イメージ | 5 |
| 3.1. 全体のイメージ | 5 |
| 3.2. 目的関数値の取得方法のイメージ | 6 |
| 3.3. NUOPT/DFO の実行方法例 | 6 |
| 4. モデリング言語 SIMPLE の記述方法 | 10 |
| 4.1. 目的関数の宣言 | 10 |
| 4.2. 目的関数への変数の登録 | 10 |
| 4.3. パラメータ設定 | 11 |
| 4.3.1. 停止条件に関するパラメータ | 11 |
| 4.3.2. 初期設定に関するパラメータ | 12 |
| 4.3.3. ペナルティパラメータに関するパラメータ | 12 |
| 4.4. その他の注意点 | 13 |
| 5. 目的関数値計算プログラムに関して | 15 |
| 5.1. 変数値情報ファイルのフォーマット | 15 |
| 5.2. 目的関数値情報ファイルのフォーマット | 15 |
| 5.3. .bat ファイルの内容と記述例 | 16 |
| 6. NUOPT/DFO の実行方法と各機能の解説 | 18 |
| 6.1. NUOPT/DFO の一般的な最適化計算機能 | 18 |
| 6.2. 最適化計算再開機能 | 19 |
| 6.3. 解情報出力機能 | 20 |
| 7. 解情報の解説 | 21 |
| 8. エラーメッセージ一覧 | 24 |
| 索引 | 29 |

1. はじめに

本利用ガイドでは、最適化パッケージソフト NUOPT のアドオンである NUOPT/DFO に関して、その利用法や注意点等を記述しています¹。なお、本利用ガイドでは NUOPT/DFO に特化した部分を中心に記述してあります。本利用ガイドに掲載されていない情報に関しましては「NUOPT/SIMPLE マニュアル」や「NUOPT/SIMPLE チュートリアル」をご参照下さい。また、最適化計算・数理計画の分野にあまり詳しくないお客様は、NUOPT の web ページ内にございます「数理計画用語集」のページ²において数理計画の分野で用いられる主な用語を解説しておりますので是非ご活用下さい。

ここでは、アドオンの名前となっている “DFO” とは何かについて簡単にご紹介します。DFO は、derivative free optimization の略称です。また、derivative free optimization とは、目的関数の数式表現が困難な状況や目的関数の微分に関する情報を用いることが出来ない状況下で求解を行なう解法の総称となります。なお、NUOPT/DFO では信頼領域法のアイデアを応用した解法をアルゴリズムとして搭載しております。搭載のアルゴリズム内では、目的関数を近似した 2 次のモデル関数を作成し部分問題を逐次解いています。

NUOPT/DFO を用いることにより、お手持ちの数式で書けない関数を目的関数とし最適化計算を行なうことが可能となります。また、モデリング言語 SIMPLE を用いた制約条件の記述にも対応しております。

¹ 一部 Windows 版での利用を想定した記述となっております。

² <http://www.msi.co.jp/nuopt/glossary/index.html>

2. NUOPT/DFO をご利用になる前に

本章では、NUOPT/DFO をお使いになる前にご確認ください事項を列挙します。

2.1. NUOPT のバージョンとモジュールの確認

お使いになられる環境が、次に挙げる「NUOPT/DFO に対応している環境」に適合しているかをご確認下さい³。

- NUOPT/DFO に対応している環境
 - NUOPT V11 以降の NLP モジュールもしくはフルセットで NUOPT/DFO 対応のライセンスを適用済である環境
 - NUOPT/DFO 対応のランタイムライセンスを適用済みである環境（他の環境で作成したモデルの実行のみ可能です）
 - NUOPT V11 以降の試用版（ご利用期間に制限があります）

2.2. パスの設定

NUOPT/DFO の機能をご利用になる前に、mknuopt.bat（Windows 版のみ）および NUOPT/DFO の実行時に呼び出す DFO.exe があるフォルダにパスを通す必要がございます。また、Windows 版でモデルのコンパイル（.smp ファイルから .exe ファイルを作成すること）を行なう際にはこの 2 つのファイルは同じフォルダにある必要があります。

なお、Windows 環境では「NUOPT をコマンドラインで使うための設定」を行なうことによりパスを通すことが可能です。「NUOPT をコマンドラインで使うための設定」に関する情報は NUOPT のインストールガイド等でご確認下さい。

³ NUOPT 学生版では原則として NUOPT/DFO の機能はご利用いただけません。NUOPT 学生版で NUOPT/DFO の機能をご利用になりたい方は nuopt-support@msi.co.jp までお問い合わせ下さい。

3. NUOPT/DFO の実行イメージ

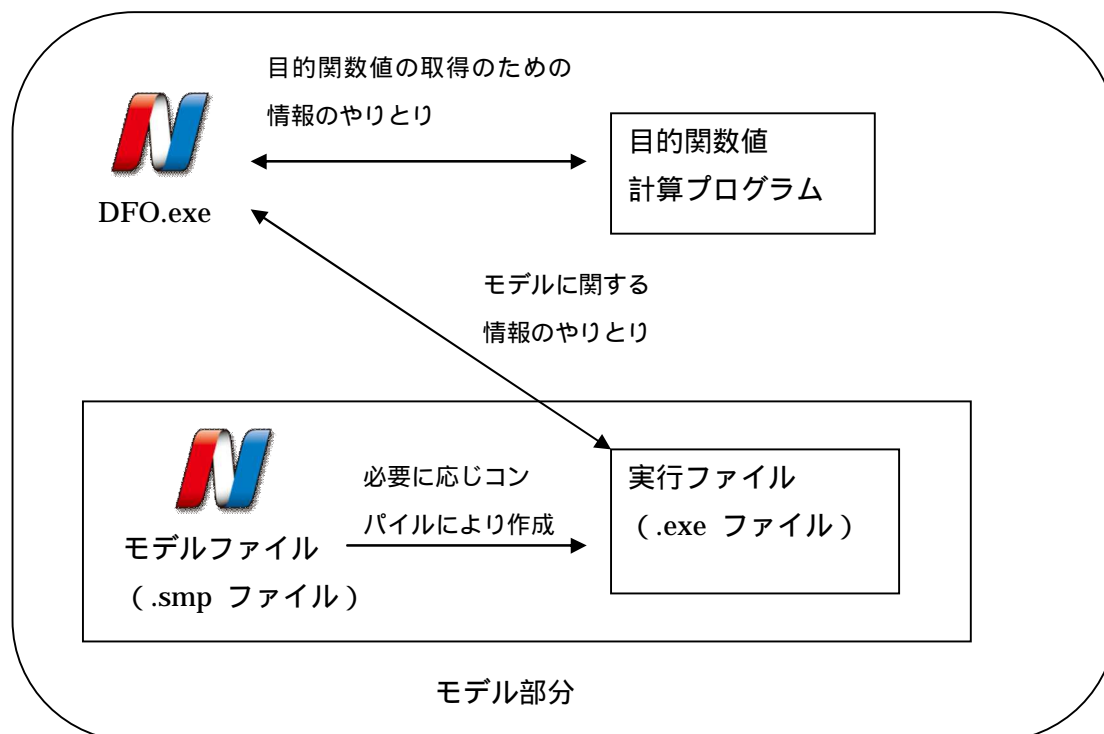
ここでは、まず NUOPT/DFO がどのような処理を行なうかについてのイメージをご紹介します。最後に、NUOPT/DFO の実行例をご紹介します。なお、本章の内容は Windows 環境で実行する場合を想定して記述してあります。

3.1. 全体のイメージ

NUOPT/DFO は次の 3 個の部分の間でデータのやり取りを行なうことにより最適化計算を行ないます。

- DFO.exe⁴ : NUOPT/DFO のメインとなる部分。NUOPT/DFO の機能を利用するにはユーザーはこの DFO.exe を実行することとなります。
- 目的関数値計算プログラム : DFO.exe から与えられたデータをもとにお手持ちの目的関数の値を計算し、出力する部分。なお、このプログラムは実行前にご用意頂く必要がございます。また、次節にイメージを記述してあります。
- モデル部分 : 制約条件の情報の記述・制御や部分問題の求解を行なう部分。なお、モデル部分は実行前にご用意頂く必要がございます。

なお、各種データのやり取りは全てファイルを介して行なうこととなります。

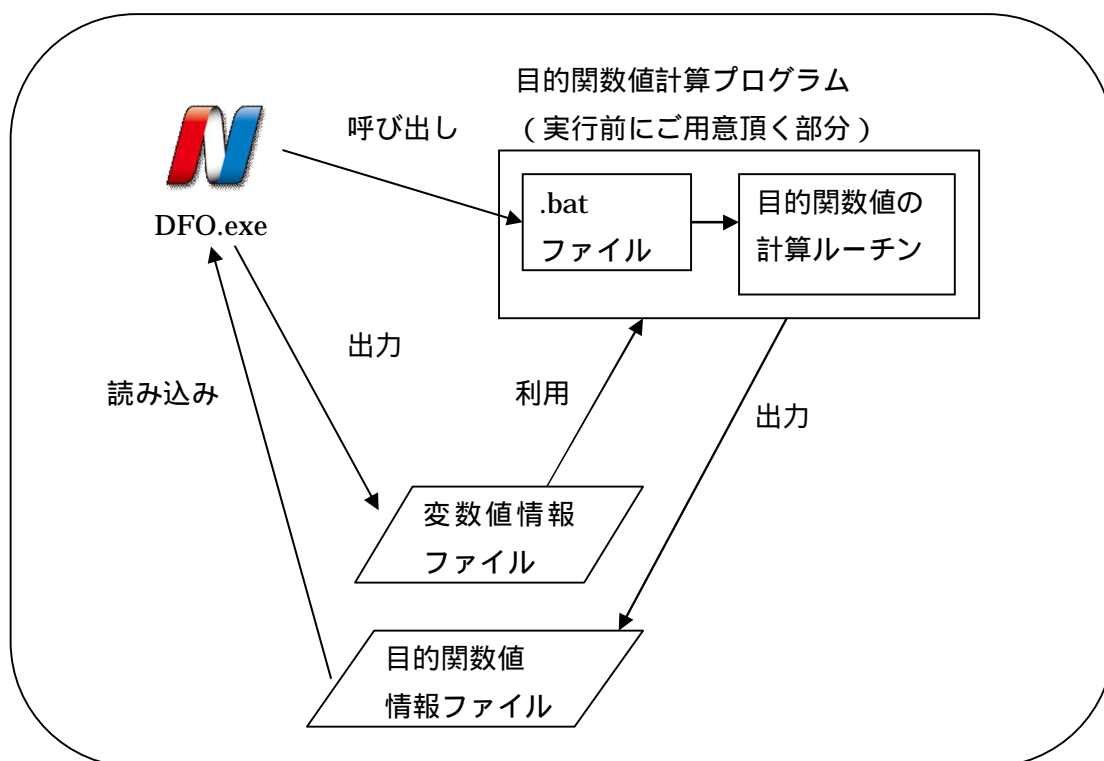


⁴ DFO.exe は Python を用いて作成しております。

Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved

3.2. 目的関数値の取得方法のイメージ

目的関数値の計算の際には、DFO.exe はまず変数値の情報をファイルに出力します。その後、モデルファイル内で指定した .bat ファイルを呼び出します。呼び出された .bat ファイル内では、変数値の情報から目的関数となるお手持ちの関数の値を計算し所定のファイルに値を出力するという処理を行なうことになります。



3.3. NUOPT/DFO の実行方法例

本節では NUOPT/DFO の実行の流れについて具体例を用いご説明します。

- 目的関数値計算プログラムおよびモデルファイルの用意

前節でご説明した目的関数値計算プログラムおよび制約条件の情報等を記述したモデルファイルを用意します。目的関数値計算プログラムの作成の際に必要な情報は「5 目的関数値計算プログラムに関して」を、モデルファイルの記述に関する情報は「4 モデリング言語 SIMPLE の記述方法」をご参照下さい。

ここでは、getobj.bat という .bat ファイルを呼ぶことで、obj.exe という実行ファイル

が目的関数値を計算し値を出力するという仕組みを用意したものとします。また、モデルファイル名は TEST.smp とします。なお、モデルファイル（もしくはモデルファイルのコンパイルにより作成された実行ファイル）と .bat ファイルは同じフォルダにある必要がございます。



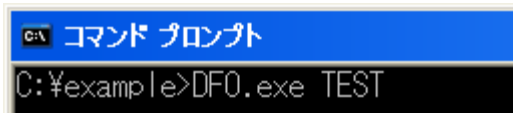
● 最適化計算の実行

NUOPT/DFO による最適化計算を行なう際には、コマンドプロンプト等に次のコマンドを入力することになります。

DF0.exe (モデル名) <データファイル名>

ここで、<データファイル名>は必要に応じて入力します。また、(モデル名)の部分について拡張子(.bat や .exe 等)を入力する必要はありません。なお、実行時にオプションを設定することが出来ますが、オプションの内容など実行方法の詳細は「6NUOPT/DFO の実行方法と各機能の解説」を参照してください。

今回の例では、オプションもデータファイルも指定しませんので、コマンドプロンプトに次のように入力します。



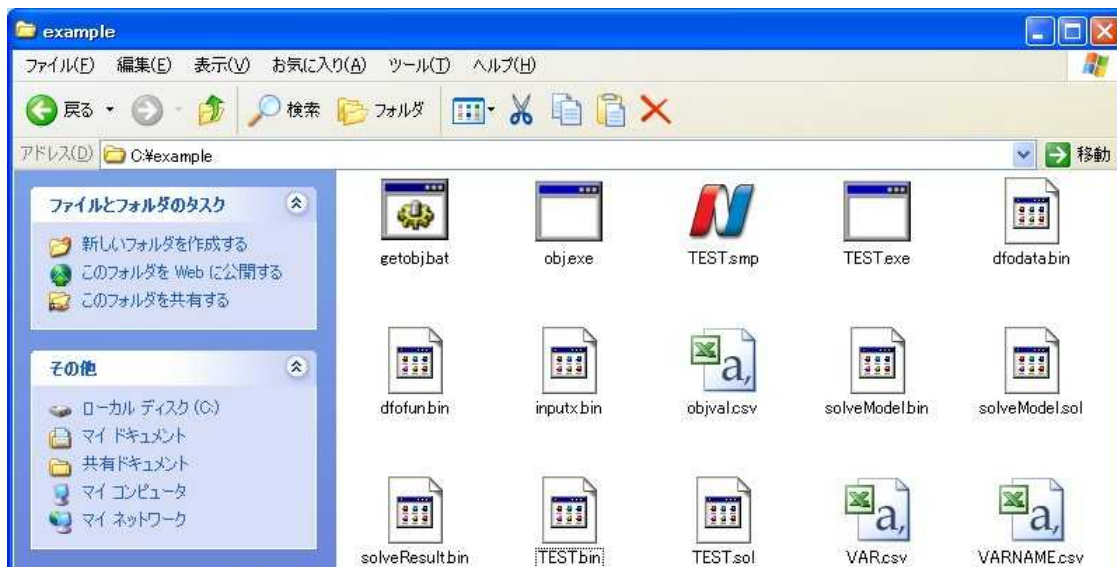
すると、DF0.exe はまずモデルファイルのコンパイルを行い TEST.smp から TEST.exe を作成します。その後、最適化計算を行ない、次の図のように結果を出力して終了します。

```

===== Status =====
NUMBER_OF_VARIABLES          2
NUMBER_OF_FUNCTIONS          1
PROBLEM_TYPE                  MINIMIZATION
METHOD                        TRUST_REGION
STATUS                        OPTIMAL(Grad)
VALUE_OF_OBJECTIVE            5.319845324e-007
ITERATION_COUNT              159
FUNC_EVAL_COUNT              198
FACTORIZATION_COUNT          0
RESIDUAL                      2.316283389e-005
ELAPSED_TIME(sec.)           38.80

```

この際、モデルファイル等があったフォルダを見るといくつかファイルが増えていることがわかります。



この内、TEST.sol というファイルには標準出力に出力されたものと同様の解情報が記述されています。

```

NUMBER_OF_VARIABLES          2
NUMBER_OF_FUNCTIONS          1
PROBLEM_TYPE                  MINIMIZATION
METHOD                        TRUST_REGION
STATUS                        OPTIMAL(Grad)
VALUE_OF_OBJECTIVE            5.319845324e-007
ITERATION_COUNT              159
FUNC_EVAL_COUNT              198
FACTORIZATION_COUNT          0
RESIDUAL                      2.316283389e-005
ELAPSED_TIME(sec.)           38.80

```


また,TEST.bin というファイルには様々な情報が格納されており,「6.2 最適化計算再開機能」や「6.3 解情報出力機能」で説明する各機能をご利用になられる場合に必要となります。このため,必要に応じて大切に保存をするようにしてください。

その他,拡張子が.csv のファイルは目的関数値(お手持ちの関数の値)を取得する際に用います。また,拡張子が.bin のファイルは DFO.exe とモデル部分との間のデータのやり取りの際に用いています。

4. モデリング言語 SIMPLE の記述方法

本章では、モデリング言語 SIMPLE の記述方法について説明します。ただし、本利用ガイドでは NUOPT/DFO に特化した記述を取り上げ説明しております。このため、本利用ガイドにて説明されていないモデリング言語 SIMPLE の一般的な記述方法に関しましては、「NUOPT/SIMPLE マニュアル」等をご参照下さい。

4.1. 目的関数の宣言

目的関数 Objective を宣言する際、目的関数値を計算する際に呼び出す .bat ファイルの名前を dfoprogram という引数で指定する必要があります。例えば、

```
Objective f(type=minimize,name="TEST",dfoprogram="getobj.bat");
```

と記述した場合、getobj.bat というファイルを目的関数値の計算時に呼び出します。

なお、NUOPT/DFO をご利用になる場合には Objective を複数回宣言することは出来ません。

4.2. 目的関数への変数の登録

NUOPT/DFO で最適化を行なう変数について、Objective で宣言した目的関数に登録を行なう必要があります。登録作業は entryVariable() を用いて行ないます。なお、この登録は目的関数値計算用の変数値情報ファイル VAR.csv に出力する値の順番を決定するために行ないます。このため、登録の順番は慎重に決定してください。また、必要な変数の登録漏れが無いようご注意ください。

entryVariable() の書式は次の通りとなります。

```
<目的関数名>.entryVariable(<変数名>)
```

ただし、以下の変数に関しましては entryVariable() することが出来ません。

- IntegerVariable で宣言された整数変数
- DiscreteVariable で宣言された離散変数

以下では、entryVariable() の記述例をご紹介します。まず、目的関数 f に変数 x を登録する場合、次のように記述します。

```
Objective f(type=minimize,name="TEST",dfoprogram="getobj.bat");
Variable x; // 変数 x の宣言
f.entryVariable(x); // 変数 x の目的関数 f への登録
```

また，添え字付きの変数 y を登録する場合には次のように記述します．

```
Set S="1 .. 7";
Element i(set=S);
Variable y(index=i);
Objective f(type=minimize,name="TEST",dfoprogram="DFO.bat");
f.entryVariable(y[i],i); // y[1],y[2],...,y[7] の順に登録される
```

最後に，添え字付きの変数の内，一部のみを登録したい場合は次のように記述します．

```
Set S="1 .. 7";
Element i(set=S);
Variable y(index=i);
Objective f(type=minimize,name="TEST",dfoprogram="DFO.bat");
f.entryVariable(y[i],(i,i>4)); // y[5],y[6],y[7] のみが順に登録される
```

4.3. パラメータ設定

ここでは NUOPT/DFO に特化したパラメータについて解説します．必要に応じて値を設定するようにしてください．なお，その他のパラメータの詳細につきましては「NUOPT/SIMPLE マニュアル」をご参照下さい⁵．

4.3.1. 停止条件に関するパラメータ

- 反復回数による停止条件

NUOPT/DFO に搭載されているアルゴリズムにおいて反復回数が指定した回数に達した場合に終了します⁶．

```
options.dfomaxitn = 2000;
```

- 信頼領域⁷の大きさによる停止条件

信頼領域の大きさが指定した値より小さくなった場合に終了します．

```
options.dfoeps_tr = 1.0e-4;
```

⁵ 本利用ガイドに掲載されていないパラメータに関しては，通常「部分問題に対するパラメータ」として認識されます．

⁶ options.maxitn により指定した値は“部分問題の”反復回数上限と解釈しますのでご注意ください．

⁷ モデル関数が妥当であると考えられる領域のこと．

- 部分問題の結果による停止条件 (1)

暫定解と部分問題の解との差の無限大ノルムの値が指定した値より小さくなった場合に「部分問題の結果による停止条件 (2)」による判定を行ないます。

```
options.dfoeps_diff = 1.0e-6;
```

- 部分問題の結果による停止条件 (2)

「部分問題の結果による停止条件 (1)」にある条件を満たした上で現状の制約違反量が指定した値より小さい場合に終了します。

```
options.dfoeps_diff_g = 1.0e-4;
```

- 制約違反量とラグランジュ関数の勾配の推定値による停止条件

一定の条件化で制約違反量とラグランジュ関数の勾配のノルムの推定値の和が指定した値より小さい場合に終了します。

```
options.dfoeps_gnL = 1.0e-4;
```

4.3.2. 初期設定に関するパラメータ

- 信頼領域の大きさの初期値

信頼領域の大きさの初期値を設定します。

```
options.dfoinitr = 0.2;
```

- 初期の点の集合を構築する際のステップ幅

初期の点の集合を構築する際に用いるステップ幅を設定します。「信頼領域の大きさの初期値」の値とのバランスが悪いと、搭載のアルゴリズムにおいて妥当なモデル関数が構築できず期待した結果が得られない可能性がありますのでご注意ください。

```
options.dfoinistep = 0.18;
```

4.3.3. ペナルティパラメータに関するパラメータ

ここでは、制約違反量の評価の際に用いるペナルティパラメータに関係するいくつかの

パラメータを解説します．

- ペナルティパラメータの初期値

ペナルティパラメータの初期値を設定します．

```
options.dfoinipenal = 10.0;
```

- ペナルティパラメータの更新頻度

ペナルティパラメータは，一定回数反復を行なうごとに内部で更新処理を行ないます．何回反復を行なうごとに更新を行なうのかを設定します．

```
options.dfochpenal = 5;
```

- ペナルティパラメータの下限值

ペナルティパラメータの下限值を設定します．更新処理を行なう際に有効なパラメータとなります．

```
options.dfominpenal = 100.0;
```

- ペナルティパラメータの大きさ

ペナルティパラメータを，部分問題から得られた双対変数値と比較してどの程度大きいものにするかを設定します．1 以上の値を設定してください．

```
options.dforatepenal = 1.1;
```

4.4. その他の注意点

以下に挙げる場合⁸に関しては，NUOPT/DFO で取り扱うことが出来ませんのでご了承下さい．

- 整数変数 IntegerVariable を用いた場合
- 離散変数 DiscreteVariable を用いた場合
- 可変定数 VariableParameter を用いた場合
- 半正定値制約を用いた制約条件を含む場合
- wccsp および rcpsp に特化した記述をした場合
- 目的関数の具体的な式をモデルファイル内で記述した場合

⁸ 主なものを挙げています．

- モデルファイル内で `solve()` を複数回記述した場合
- 上下限の値が一致する等の理由により値が固定されてしまう変数がある場合

また、以下の場合につきましてはパフォーマンスが低下する可能性がありますのでご注意ください。


- `entryVariable()` されていない変数を含む場合

5. 目的関数値計算プログラムに関して


本章では，事前にご用意頂く目的関数値計算プログラムの作成の際に重要となる情報を解説します．具体的には，目的関数値の取得の際に用いるデータファイルのフォーマットの解説およびご用意頂く .bat ファイルに関する説明を行ないます．

5.1. 変数値情報ファイルのフォーマット

目的関数値計算用の変数値の情報は，VAR.csv というファイルに次のフォーマットで出力されます．

| | |
|--------------------|---|
| (1 番目に登録された変数の値) |  |
| (2 番目に登録された変数の値) | |
| (3 番目に登録された変数の値) | |
| | |

ここでの「登録された順番」は「entryVariable() された順番」のことです．なお，entryVariable() されていない変数の値は出力されませんのでご注意ください．また，「登録された順番」の確認用に VARNAME.csv というファイルに変数の名前に関する情報を次のフォーマットで出力いたします⁹ので，正しい目的関数値が得られない場合等に適宜ご確認下さい．

| | |
|---------------------|--|
| (1 番目に登録された変数の名前) |  |
| (2 番目に登録された変数の名前) | |
| (3 番目に登録された変数の名前) | |
| | |



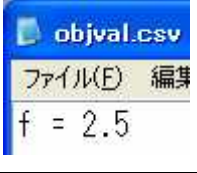


5.2. 目的関数値情報ファイルのフォーマット

お手持ちの環境で計算された目的関数値の情報は objval.csv というファイルに出力してください．なお，DFO.exe では，objval.csv 内の「数値（指数形式を含む）のみの行」を目的関数値として認識します¹⁰．また，「数値（指数形式を含む）のみの行」が複数ある場

⁹ ある変数について変数名が長すぎる場合については，その変数の名前は途中まで表示されます．

¹⁰ 区切り文字が含まれていると正しく認識しない可能性があります．

合には後にある方の値を優先します．以下にいくつかの例を示します．

| | objval.csv の内容 | 認識される 目的関数値 | 解説 |
|-----|---|----------------|---|
| 例 1 |  | 25.645 | 数値のみなので正しく認識します． |
| 例 2 |  | 0.1 | 指数形式は正しく認識します． |
| 例 3 |  | (エラー) | 「数値のみ」ではないので目的関数値として認識しません．このため objval.csv 内に目的関数値の情報が無いと判断し <<DFO 21>> のエラーを出力します． |
| 例 4 |  | 56.3 | 1 行目は「数値のみ」ではないので目的関数値として認識しません． |
| 例 5 |  | -6.35 | 「数値（指数形式を含む）のみの行」が複数ある場合は後の値が優先されます． |

5.3. .bat ファイルの内容と記述例

目的関数値の計算時，モデルファイル内で指定した .bat ファイルを呼び出すことになります．なお，具体的には NUOPT/DFO のメインモジュールから次のような引数を入れた形で呼び出します．

(.bat ファイル名) (変数値情報ファイル名) <データファイル名>

<データファイル名> は，NUOPT/DFO 実行時にユーザーが必要に応じ指定したファイル名の事を指します．このため，お手持ちの関数の値を計算する際にパラメータの情報が必要な際には引数にあるデータファイル名の情報を利用してください．

.bat ファイル内では，必要なプログラムを実行する等の処理を記述することになります．

その際、正常終了した場合は「exit 0」と記述し 0 が .bat ファイルからの戻り値となるように、また異常終了の場合は「exit 1」のように 0 以外の値が .bat ファイルからの戻り値となるようにして下さい¹¹。

例えば、変数値データが格納されたファイルの名前 VAR.csv を引数として与えると objval.csv に関数値を出力するプログラム obj.exe を呼び出すようにしたい場合には次のように .bat ファイルを記述します。ただし、obj.exe は正常終了時には 0 を、異常終了時には正の値を戻り値として返すプログラムであるものとします。

```
@echo off
obj.exe VAR.csv
if errorlevel 1 goto error
exit 0
:error
exit 1
```

¹¹ /b オプションは使用しないで下さい。

6. NUOPT/DFO の実行方法と各機能の解説

NUOPT/DFO の機能としては主に以下のものが挙げられます．

- モデルファイルからコンパイルにより実行ファイルを作成する機能
- 最適化計算を行なう機能
- 暫定解を初期点として最適化計算を再開する機能
- 解の情報を出力する機能

本章では，上記の各機能の解説および実行方法に関する説明を行ないます．

6.1. NUOPT/DFO の一般的な最適化計算機能

本節では NUOPT/DFO の利用方法をご説明します．Windows 版では，NUOPT/DFO のメインプログラムである DFO.exe はコマンドプロンプト等に次のコマンドを入力することで起動します．なお，NUOPT GUI からの起動には対応しておりませんのでご了承ください．

DFO.exe <オプション> (モデル名) <.bin ファイル名> <データファイル名>

上記のコマンドの内，(モデル名)に関しては拡張子の入力は不要です．また，<オプション>および<データファイル名>は必要に応じて記述することになり，<.bin ファイル名>は一部のオプションを指定した場合に必要となります．

オプションには次のものがございます．なお，「モデルのコンパイル」とは「モデルファイルからコンパイルにより実行ファイルを作成すること」を意味します．

| オプション | 意味 |
|--------------|---|
| -compile | モデルのコンパイルを“必ず”行った後に最適化計算を開始します．モデルファイルが存在しない場合やコンパイルが失敗した場合は最適化計算を行わず終了となります． |
| -nocompile | モデルのコンパイルを行わずに最適化計算を開始します． |
| -onlycompile | モデルのコンパイル“のみ”行い，最適化計算は実行しません． |
| -restart | 暫定解を初期点として最適化計算を再開します．詳細は「6.2 最適化計算再開機能」を参照してください．なお，モデルのコンパイルは行ないません． |
| -outdata | (暫定)解の情報の出力のみを行います．詳細は「6.3 解情報出力機能」を参照してください．なお，モデルのコンパイルは行ないません． |

オプションを指定しないで実行した場合，Windows 版では以下の場合にモデルのコンパ

イルを自動的に行ないます。

- (モデル名).smp は存在するが、(モデル名).exe は存在しない場合。
- (モデル名).smp と (モデル名).exe が共に存在し、最終更新時刻は(モデル名).smp の方が新しい場合。

自動的にモデルのコンパイルを行ないたくない場合には、`-nocompile` オプションを付けて実行してください。

<データファイル名>で指定したデータファイルは、NUOPT(SIMPLE) が解釈可能なフォーマットで記述されている必要がございます。データのフォーマットに関する詳細は「NUOPT/SIMPLE マニュアル」をご参照下さい。

(注意) NUOPT/DFO 用に作られた実行ファイルでは通常の NUOPT による最適化計算を行なうことが出来ません。また、通常の NUOPT による最適化計算を行なうために作成された実行ファイルでは NUOPT/DFO の機能を利用することが出来ません。なお、NUOPT/DFO 用の特別な処理を行っていない状態で `mknuopt.bat` を実行することにより作成された実行ファイルでは NUOPT/DFO の機能を利用することが出来ませんのでご注意ください¹²。

なお、標準出力には 1 反復終了毎に暫定解の値と目的関数値の情報を次のように出力いたします。この際、暫定解の値は `entryVariable()` により登録された順で出力されます。

```
current solution -> [0.68611054231422397, 0.46982330599912997]
Objective -> 9.8612037687e-002
```

6.2. 最適化計算再開機能

NUOPT/DFO では 1 反復終了ごとに (モデル名).bin というファイルを出力します。この .bin ファイル内には暫定解の情報が格納されており、対応するモデルの実行ファイルがある場合には暫定解の値を初期値として探索を再開することが出来ます。この機能は、求解途中でやむを得ず処理を中断させた等の場合に役立つものとなります。

最適化計算再開機能を実行する際には、Windows 環境ではコマンドプロンプト等から次のコマンドを入力します。この際、<データファイル名> は必要に応じて入力します。

```
DFO.exe -restart (モデル名) (.bin ファイル名) <データファイル名>
```

なお、.exe ファイルと .bin ファイルが対応していない場合の挙動に関しましては保証

¹² 「4 モデリング言語 SIMPLE の記述方法」で説明した NUOPT/DFO に特化した記述がされている場合でも正しく動作いたしません。

いたしかねますのでご注意ください。また、本機能では内部で用いる情報全てを再現した上で再開するわけではないため、中断等をせずに求解を続けた場合の挙動と最適化計算再開機能を用いた場合の挙動が一致するとは限りません。

最後に、「7 解情報の解説」で解説する解情報のうち次の 3 つに関しては再開前から通算した値が表示されます。再開後のみの値ではありませんのでご注意ください。

- ITERATION_COUNT （最適化計算の反復回数）¹³
- FUNC_EVAL_COUNT （目的関数値の評価回数）
- ELAPSED_TIME(sec.) （最適化計算の実行時間）

6.3. 解情報出力機能

NUOPT/DFO では 1 反復終了ごとおよび最適化計算終了時に（モデル名）.bin というファイル出力します。この .bin ファイル内にある情報をもとに（暫定）解に関する情報を（モデル名）.sol というファイルに出力することが出来ます。

解情報出力機能を利用する場合、Windows 環境ではコマンドプロンプト等から次のコマンドを入力します。

DFO.exe -outdata （モデル名）（.bin ファイル名）<データファイル名>

なお、.exe ファイルと .bin ファイルが対応していない場合の挙動に関しましては保証いたしかねますのでご注意ください。

¹³ 「4.3.1 停止条件に関するパラメータ」にある「反復回数による停止条件」は通算の回数が基準となります。

7. 解情報の解説

NUOPT/DFO による最適化計算の終了時には、標準出力に

```
===== Status =====
```

と表示された後に解情報もしくはエラー時の情報（暫定解に関する情報がある場合のみ）が出力されます。ここでは各行の内容をご紹介します。なお、本章でご紹介する情報にしましては（モデル名）.sol というファイルにも出力されております。また（モデル名）.exe と（モデル名）.bin の 2 つのファイルがある場合、（モデル名）.sol の内容を再度出力することが出来ます。再度出力する方法の詳細は「6.3 解情報出力機能」を参照してください。

| | |
|---------------------|--------------|
| NUMBER_OF_VARIABLES | 2 |
| NUMBER_OF_FUNCTIONS | 2 |
| PROBLEM_TYPE | MINIMIZATION |
| METHOD | TRUST_REGION |

上から順に、「変数の数」、「関数の数」、「問題の種類（最大化問題か最小化問題か）」、「部分問題に対して使用した解法」を表しています。

| STATUS | OPTIMAL(Model) |
|--------|----------------|
|--------|----------------|

どの停止条件を満たして終了したかあるいはどのようなエラーによって終了したかの情報を表します。停止内容の詳細は次の表の通りとなります。

| STATUS の表示内容 | 詳細 |
|------------------------------|---|
| OPTIMAL(Trust-region radius) | 信頼領域の大きさがある値より小さくなったことにより終了したことを表します。 |
| OPTIMAL(Model) | 制約違反量がある値より小さい領域内で、暫定解の値と部分問題の解が十分近いため終了したことを表します。 |
| OPTIMAL(Grad) | 一定の条件下で、ラグランジュ関数の勾配のノルムの推定値がある値より小さくなったため終了したことを表します。 |
| NON_OPTIMAL | エラーにより終了したことを表します。この場合、エラー内容は「ERROR_TYPE」の行に表示されます。 |
| INTERMEDIATE | 最適化計算の途中であることを表します。 |

| | |
|--------------------|------------------|
| VALUE_OF_OBJECTIVE | 1.027290347e-006 |
|--------------------|------------------|

得られた解での目的関数値が表示されます。なお、エラー終了時あるいは最適化計算の途中経過の出力時には暫定解での目的関数値が出力されます。

| | |
|-----------------|----|
| ITERATION_COUNT | 18 |
|-----------------|----|

NUOPT/DFO に搭載のアルゴリズムの反復回数を表示します¹⁴。部分問題の反復回数ではありませんのでご注意ください。

| | |
|-----------------|----|
| FUNC_EVAL_COUNT | 24 |
|-----------------|----|

目的関数値（お手持ちの関数の値）を何回評価したのかを表しています¹⁴。

| | |
|--------------------------|------------------|
| RESIDUAL | 3.935709213e-007 |
| CONSTRAINT_INFEASIBILITY | 4.067977954e-009 |

残差の情報および得られた解がどの程度制約条件に違反しているのかを表します。

| | |
|--------------------|------|
| ELAPSED_TIME(sec.) | 4.42 |
|--------------------|------|

NUOPT/DFO による最適化計算の実行時間を表しています¹⁴。

| | | | | | |
|--------------|----------|--------|---------------------------|--------------|---|
| %% | | | | | |
| %% VARIABLES | | | | | |
| %% | | | | | |
| NAME | VALUE | STATUS | SLACK [| BOUND TYPE |] |
| V# 1 X1 | 0.998986 | FREE | [-Inf <= | X1 <= +Inf] | |
| V# 2 X2 | 1.00036 | FREE | 2.50036100e+000 [-1.5 <= | X2 |] |

変数に関する情報を表示します。「VALUE」の列に書かれている値が得られた（暫定）解での値ということになります。

¹⁴ 最適化計算再開機能を用いた場合には再開前から通算した値が表示されます。

```
%%
%% FUNCTIONS
%%
      NAME      VALUE      STATUS      SLACK      [ BOUND TYPE      ]
F# 1 HS1    1.02729e-006  FREE                      [ OBJECTIVE (MINIMIZE)      ]
F# 2 HS6.smp:14    -4.06798e-009  INFS  -4.06797795e-009 [ HS6.smp:14 == 0 ]
```

関数に関する情報を表示します .

8. エラーメッセージ一覧

次表は NUOPT/DFO の機能に関するエラーメッセージの一覧です。なお、下記のエラーメッセージ以外に SIMPLE や NUOPT に関するエラーメッセージが同時に出力される場合がございます。次表にないエラーメッセージにつきましては「NUOPT/SIMPLE マニュアル」をご参照下さい。

| エラー番号 | エラーメッセージ | 原因・対処法など |
|-------|--|--|
| 1 | <<DFO 1>> Please specify model name. Usage : DFO.exe (modelname) (datafilename<option>) | NUOPT/DFO を実行する際に、最適化するモデルの名前を指定していない場合に発生します。モデルの名前を指定して再度実行してください。 |
| 2 | <<DFO 2>> Can't find model file in current directory. | 指定したモデルに対応する実行形式とモデルファイルが共に作業ディレクトリに存在しない場合発生します。ファイルの存在の有無あるいは指定したモデル名が正しいかをご確認下さい。 |
| 3 | <<DFO 3>> Failed to create .exe file. | 実行形式の作成に失敗した場合に発生します。SIMPLE の記述に誤りがないかをご確認下さい。 |
| 4 | <<DFO 4>> Internal Error occurred. (<詳細情報>) | 内部エラーが発生したことを表します。 nuopt-support@msi.co.jp にご連絡下さい。 |
| 5 | <<DFO 5>> Can't find bat file in current directory.(<bat ファイル名>) | 目的関数値の計算の際に必要な .bat ファイルが作業ディレクトリ内に存在しない場合に発生します。.bat ファイルが存在するかをご確認下さい。 |
| 6 | <<DFO 6>> Invalid option. | パラメータとして有効な範囲外の値を設定しようとすると発生します。有効な値を指定してください。 |
| 7 | <<DFO 7>> Illegal value in parameter file. | パラメータとして(型が異なるなど)不正な値を設定しようとすると発生します。有効な値を指定してください。 |
| 8 | <<DFO 8>> I/O error occurred. | ファイル入出力の際にエラーが発生したことを表します。ファイル入出力に関する適切な権限を持っているか等をご確認下さい。また、このエラーは同一フォルダ内で |

| | | |
|----|---|--|
| | | NUOPT/DFO を用いて複数の問題を同時に求解しようとした時に発生する場合があります。 |
| 9 | <<DFO 9>> getValKind() failed. | 変数に関する情報の取得に失敗したことを表します。 |
| 10 | <<DFO 10>> getFuncKind() failed. | 関数に関する情報の取得に失敗したことを表します。 |
| 11 | <<DFO 11>> Failed to set objective function's name. | 目的関数名の情報を正しく取得できなかった場合に発生します。 |
| 12 | <<DFO 12>> getFunl() failed. | 関数の線形項に関する情報の取得に失敗したことを表します。 |
| 13 | <<DFO 13>> getInitialPointVal() failed. | 初期値の情報の取得に失敗した場合に発生します。 |
| 14 | <<DFO 14>> getGradnl() failed. | 関数の勾配に関する情報の取得に失敗したことを表します。 |
| 15 | <<DFO 15>> Infeasible.(variable bounds) | 変数の上下限值に問題があるときに表示されます。変数の上下限の値を正しく設定しているかご確認ください。 |
| 16 | <<DFO 16>> Invalid value.(variable kind) | 不正な変数があるときに表示されます。 |
| 17 | <<DFO 17>> Overflow occurred. | 計算処理の際にオーバーフローが発生したことを表します。 |
| 18 | <<DFO 18>> Division by zero occurred. | 計算処理の際に 0 割りが発生したことを表します。 |
| 19 | <<DFO 19>> getFunnl() failed. | 関数の非線形項に関する情報の取得に失敗したことを表します。 |
| 20 | <<DFO 20>> Illegal value from .bat file. (returncode = <エラー番号>) | 目的関数値取得用の .bat ファイルがエラーコード(0 以外の値)を返してきたことを表します。目的関数値の計算工程の中に誤りがないかご確認ください。 |
| 21 | <<DFO 21>> Can't find objective function's value in data file. | 指定されたファイルに目的関数値の情報が記述されていないと判断した場合に発生します。目的関数値の記述先が正しいかあるいは正しいフォーマットで記述されているかをご確認ください。 |
| 22 | <<DFO 22>> Can't find objective function's data file.(<ファイル名>) | 目的関数値の情報が記述されているファイルが存在しない場合に発生します。目的関数値の記述先が正しいかご確認ください。 |
| 23 | <<DFO 23>> Illegal trust-region radius. | 信頼領域の大きさが 0 以下になったことを表します。 |
| 24 | <<DFO 24>> Length of two vectors is different. | 長さが異なることが原因で 2 つのベクトルに関する演算に失敗したことを表しま |

| | | |
|----|--|---|
| | | す . |
| 25 | <<DFO 25>> Max value is too small. | 内部で最大値を取得する処理を行なった結果、その値が小さすぎるため後の処理に影響があると判断したことを表します . |
| 26 | <<DFO 26>> Failed to adjust Lagrange functions appropriately. | 内部で用いている補間多項式の調整処理に失敗した場合に表示されます . |
| 27 | <<DFO 27>> Keyboard interrupt. | キーボード割り込みにより処理を中止したときに表示されます . なお、モデルの実行形式や目的関数値取得のための .bat ファイル等の処理を待っている際にキーボード割り込みが発生した場合には「(waiting)」という文字列も一緒に表示されます . |
| 28 | <<DFO 28>> outModel() failed. | 結果ファイルの出力に失敗したことを表します . |
| 29 | <<DFO 29>> solveModel() failed. | 部分問題の求解の際に異常が発生したことを表します . |
| 30 | <<DFO 30>> Iteration limit exceeded. | NUOPT/DFO に搭載のアルゴリズムに関して、反復回数の上限に達した場合に表示されます . |
| 31 | <<DFO 31>> Failed to find new point.(min) | 部分問題を解くことによる新たな候補点の取得に失敗したときに表示されます . |
| 32 | <<DFO 32>> Failed to find new point.(max) | 部分問題を解くことによる新たな候補点の取得に失敗したときに表示されます . |
| 33 | <<DFO 33>> Please specify file name. Usage : DFO.exe -outdata (modelname) (logfilename) (datafilename<option>) | 解情報出力機能の実行時、必要な引数が記述されていなかった場合に出力されます . 引数を適切に入力して再度実行してください . |
| 34 | <<DFO 34>> Can't find exe file in current directory. | 解情報出力機能の利用時もしくは -nocompile オプション指定時に、引数で指定した実行ファイルが発見できなかった場合に表示されます . 引数の記述が正しいかもしくは実行ファイルが存在するかをご確認下さい . |
| 35 | <<DFO 35>> Can't find datafile in current directory. | 解情報出力機能の利用時に、引数で指定した解情報データファイルが発見できなかった場合に表示されます . 引数の記述が正しいかもしくは解情報データファイルが存在するかをご確認下さい . |

| | | |
|----|--|--|
| 36 | <<DFO 36>> Can't find mknuopt.bat. | -compile オプションもしくは -onlycompile オプションを指定したにもかかわらず mknuopt.bat が発見できなかった場合に出力されます .mknuopt.bat があるディレクトリにパスが通っているかご確認ください . |
| 37 | <<DFO 37>> Please specify model name and .bin file name. Usage : DFO.exe -restart (modelname) (.bin filename) (datafilename<option>) | -restart オプションで再スタートをする際の引数の指定に誤りがある場合に表示されます . 正しく引数を入力して再度実行してください . |
| 38 | <<DFO 38>> Can't find .bin file. | 再スタート時に指定した .bin ファイルを発見できなかったときに表示されます . 指定した .bin ファイルが正しく存在するかご確認ください . |
| 39 | <<DFO 39>> IntegerVariable "<変数名>" misapplied to DFO. | IntegerVariable を使用している場合に表示されます . NUOPT/DFO は IntegerVariable には対応しておりませんのでご了承下さい . |
| 40 | <<DFO 40>> DiscreteVariable "<変数名>" misapplied to DFO. | DiscreteVariable を使用している場合に表示されます . NUOPT/DFO は DiscreteVariable には対応しておりませんのでご了承下さい . |
| 41 | <<DFO 41>> SDP constraint misapplied to DFO. | 半正定値制約を使用している場合に表示されます . NUOPT/DFO は半正定値制約には対応しておりませんのでご了承下さい . |
| 42 | <<DFO 42>> Objective cannot be assigned. | 目的関数 (Objective) に具体的な式を代入しようとしている場合に表示されます . 目的関数値に関しては .bat ファイルを介してのやり取りとなります . |
| 43 | <<DFO 43>> Objective must be defined with arguments("dfoprogram" missing) . | 目的関数 (Objective) の宣言の際に必要な引数が記述されていない場合に表示されます . .bat ファイル名の情報が引数として記述されているかご確認ください . |
| 44 | <<DFO 44>> Objective can only be defined for once. | 目的関数 (Objective) の宣言を複数回行なっている場合に表示されます . NUOPT/DFO では Objective の宣言は 1 回のみ行なうことができます . |
| 45 | <<DFO 45>> Illegal method "<解法名>" applied to subproblem. | wcsp や rcpsp のような部分問題の解法として利用できないものを指定した場合に出力されます . 適切な解法を選択し , 再度実行してください . |
| 46 | <<DFO 46>> entryVariable() is not used in model file. | entryVariable() による変数の登録を行っていない場合に表示されます . |

| | | |
|----|---|--|
| | | entryVariable() による変数の登録を行なった後に再度実行してください。 |
| 47 | <<DFO 47>> VariableParameter “<変数名>” misapplied to DFO. | VariableParameter が用いられている場合に表示されます。NUOPT/DFO は VariableParameter には対応しておりませんのでご了承下さい。 |
| 48 | <<DFO 48>> Variable “<変数名>” is fixed. | ある変数の値が固定されてしまう場合に表示されます。該当するものを変数ではなく Parameter 等を用い定数とした上で実行してください。 |
| 49 | <<DFO 49>> solve() is called more than once. | モデルファイル内で solve() が複数回使用されている場合に表示されます。NUOPT/DFO ではモデルファイル内で solve() を複数回用いることは出来ません。 |
| 50 | <<DFO 50>> outParam() failed. | パラメータに関する情報の取得に失敗したときに表示されます。 なお、このエラーはモデルファイルをコンパイルして出来た実行ファイルが NUOPT/DFO 用に作成されたものではない場合にも表示されます。 |
| 51 | <<DFO 51>> Failed to get address. | DFO.exe があるフォルダにパスが通っていない場合に表示されます。パスを通した後に実行してください。 |

索引

.

.bat ファイル 6, 10, 15, 16

.bin ファイル 19, 20

D

derivative free optimization 3

DFO.exe 4, 5, 6, 7, 9, 15, 18

dfoprog 10

E

entryVariable 10, 14, 15, 19

M

mknuopt.bat 4, 19

O

Objective 10

objval.csv 15, 17

options.dfochpenal 13

options.dfoeps_diff 12

options.dfoeps_diff_g 12

options.dfoeps_gnL 12

options.dfoeps_tr 11

options.dfoinipenal 13

options.dfoinistep 12

options.dfoinitr 12

options.dfomaxitn 11

options.dfominpenal 13

options.dforatepenal 13

S

SIMPLE 3, 10, 19

V

VAR.csv 10, 15, 17

VARNAME.csv 15

か

解情報 8, 20, 21

さ

最適化計算再開機能 19, 22

暫定解 18, 19, 20, 21, 22

し

信頼領域 11, 12, 21

信頼領域法 3

て

停止条件 11, 12, 20

は

パラメータ 11, 12

へ

変数値情報ファイル 10, 15

も

目的関数値計算プログラム 5, 6, 15

目的関数値情報ファイル 15

モデルファイル 6, 7, 13, 16, 18