Numerical Optimizer

DFO利用ガイド
V19

株式会社NTTデータ数理システム

2017年1月

MATHEMATICAL SYSTEMS INCARNATION

目次

1. はじめに

本利用ガイドでは，最適化パッケージソフト Numerical Optimizer のアドオンである Numerical Optimizer/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 Numerical Optimizer/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「Numerical Optimizer/SIMPLE マニュアル」や「Numerical Optimizer/SIMPLE チュートリアル」をご参照ください．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，Numerical Optimizer 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ください．

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，Numerical Optimizer/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．

Numerical Optimizer/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] 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. Numerical Optimizer/DFO をご利用になる前に

本章では, Numerical Optimizer/DFO をお使いになる前にご確認いただきたい事項を列挙します.

2.1. Numerical Optimizer 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「Numerical Optimizer/DFO に対応している環境」に適合しているかをご確認ください[3][4].

- Numerical Optimizer/DFO に対応している環境
  - Numerical Optimizer V11 以降のフルセットで Numerical Optimizer/DFO 対応のライセンスを適用済である環境
  - Numerical Optimizer/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）
  - Numerical Optimizer V11 以降の試用版（ご利用期間に制限があります）
  - Numerical Optimizer V11 以降 V17 までの NLP モジュールで Numerical Optimizer/DFO 対応のライセンスを適用済である環境

2.2. パスの設定

Numerical Optimizer/DFO の機能をご利用になる前に，mknuopt.bat（Windows 版のみ）および Numerical Optimizer/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「Numerical Optimizer 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.「Numerical Optimizer をコマンドラインで使うための設定」に関する情報は Numerical Optimizer のインストールガイド等でご確認ください.

---

[3] V15 までにつきましては「Numerical Optimizer」は「NUOPT」と読み替えてください.

[4] Numerical Optimizer 学生版では原則として Numerical Optimizer/DFO の機能はご利用いただけません．Numerical Optimizer 学生版で Numerical Optimizer/DFO の機能をご利用になりたい方は nuopt-support@msi.co.jp までお問い合わせください.

3. Numerical Optimizer/DFO の実行イメージ

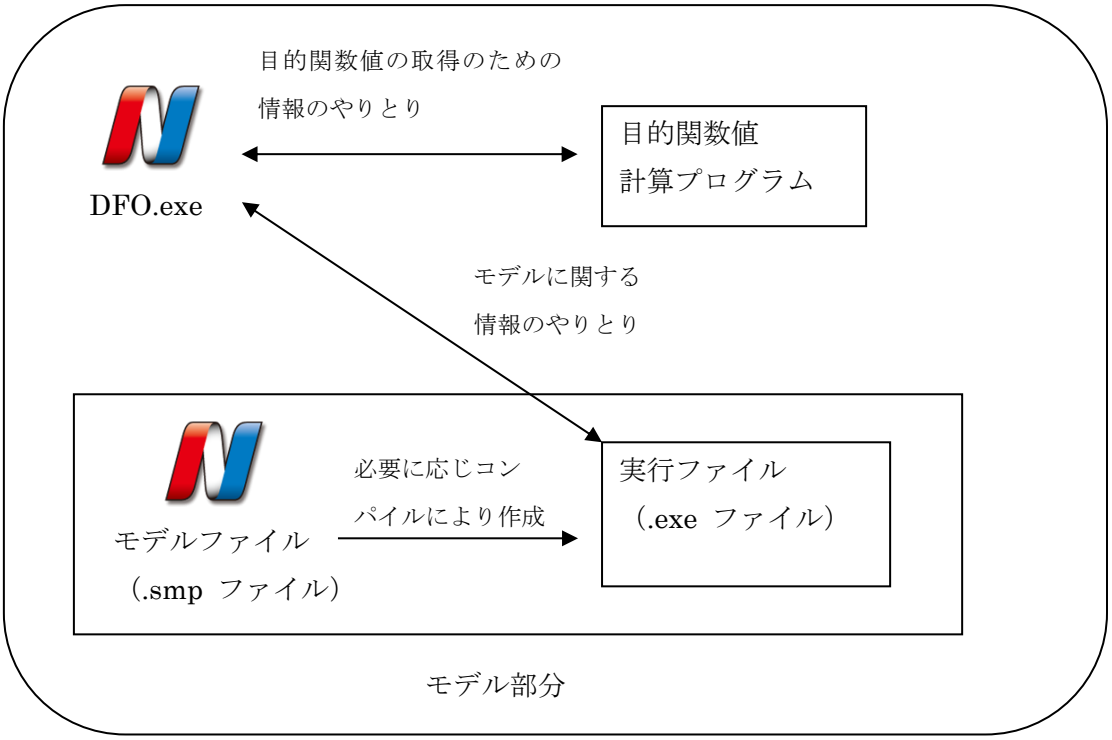
ここでは，まず Numerical Optimizer/DFO がどのような処理を行なうかについてのイメージをご紹介します．最後に，実行例をご紹介します．なお，本章の内容は Windows 環境で実行する場合を想定して記述してあります．

3.1. 全体のイメージ

Numerical Optimizer/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.exe[5]：Numerical Optimizer/DFO のメインとなる部分であり，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．

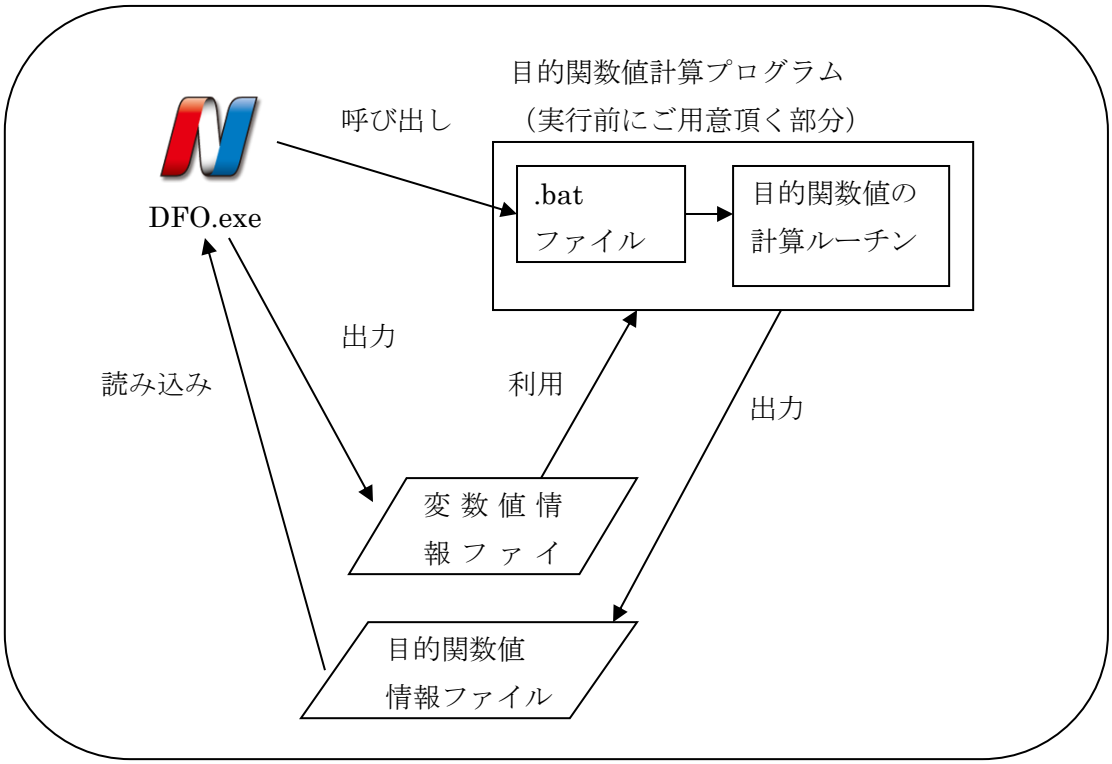
なお，各種データのやり取りは全てファイルを介して行なうことになります．



[5] DFO.exe は Python を用いて作成しております．

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. Numerical Optimizer/DFO の実行方法例

本節では Numerical Optimizer/DFO の実行の流れについて具体例を用いご説明します．まず，本節では以下の問題[6]を解くこととします．

6 「Willi Hock, Klaus Schittkowski：Test Examples for Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems, No 187, 1981. Springer.」内の PROBLEM No.1

変数：	x_1，x_2
目的関数（最小化）：	$100(x_2 - x_1^2)^2 + (1 - x_1)^2$ [→ 外部プログラムで計算]
制約条件：	$-1.5 \leq x_2$
変数の初期値：	$x_1 = -2$，$x_2 = 1$

この問題に関して，ここでは目的関数の具体的な式をモデリング言語 SIMPLE で記述するのではなく，外部プログラムが計算した目的関数値の情報を利用するものとします．

なお，

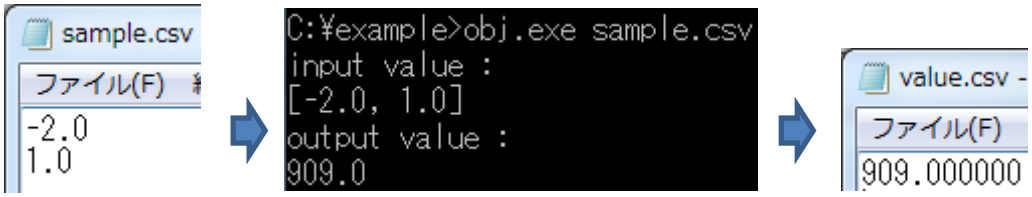
（Numerical Optimizer のインストール先）¥SAMPLES¥DFOsample

に本例題に関する目的関数値計算プログラム等のサンプルがございますので，適宜ご利用ください[7]．

- 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

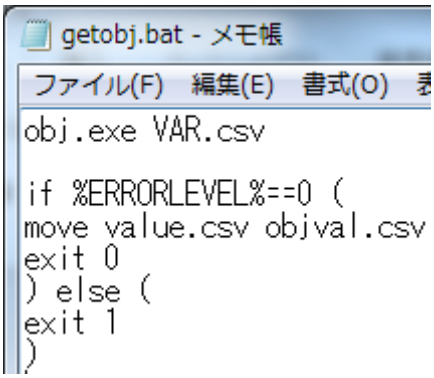
前節でご説明した目的関数値計算プログラムおよび制約条件の情報等を記述したモデルファイルを用意します．目的関数値計算プログラムの作成の際に必要となる情報は「5 目的関数値計算プログラムに関して」を，モデルファイルの記述に関する情報は「4 モデリング言語 SIMPLE の記述方法」をあわせてご参照ください．

目的関数値の計算に関して，ここでは以下のように x_1，x_2 に対応する値を記述した csv を引数として与えると，value.csv というファイルに目的関数値を出力する実行ファイル obj.exe を用意したものとします．



次に，この obj.exe を DFO.exe から実行する際に利用する .bat ファイルを用意します．ここではファイル名は getobj.bat とし，以下の内容を記述します．

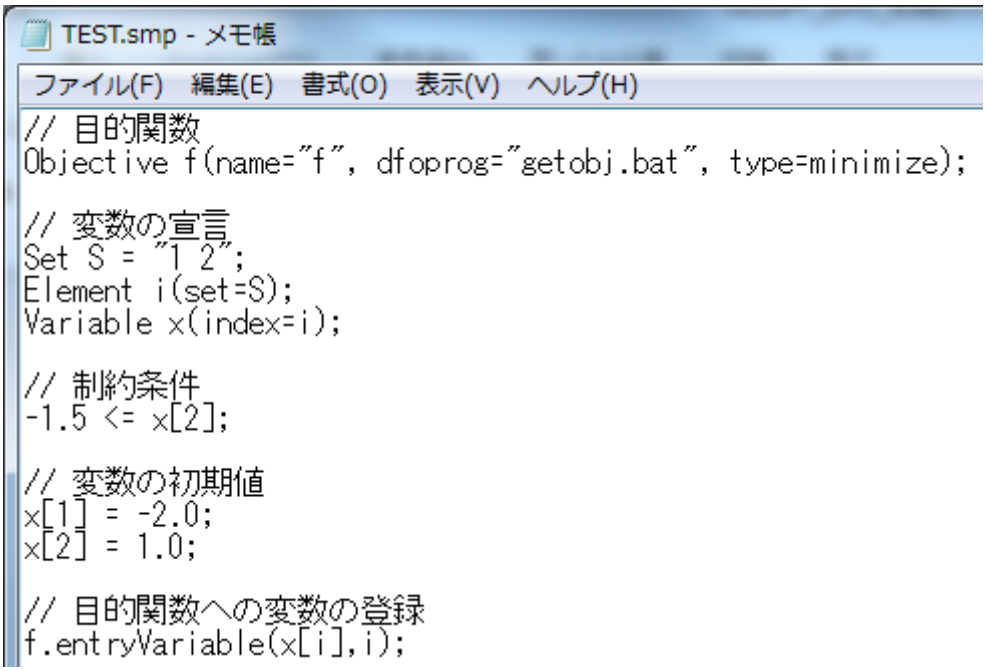
[7] Numerical Optimizer のデフォルトのインストール先は
「C:¥Program Files¥Mathematical Systems Inc¥NUOPT」(64bit 版 Windows では
「C:¥Program Files (x86)¥Mathematical Systems Inc¥NUOPT」)となります．



getobj.bat に関して，この例では DFO.exe が変数値を出力するファイル VAR.csv と obj.csv が処理できる入力ファイルのフォーマットが同じであることから 1 行目で obj.exe の引数として VAR.csv を与えています．また，obj.exe が出力するファイル value.csv であるのに対し DFO.exe が目的関数値の情報として受け取るファイルは objval.csv であることから，4 行目でファイルの移動により対応しています．

次に，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．

TEST.smp には以下のようにモデルを記述します．



目的関数の宣言で dfoprog=”getobj.bat” と指定している点と entryVariable を用いて目的関数への変数の登録を行う点が Numerical Optimizer/DFO による最適化計算を行なう際に特徴的な事項となります.

- 最適化計算の実行

Numerical Optimizer/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.

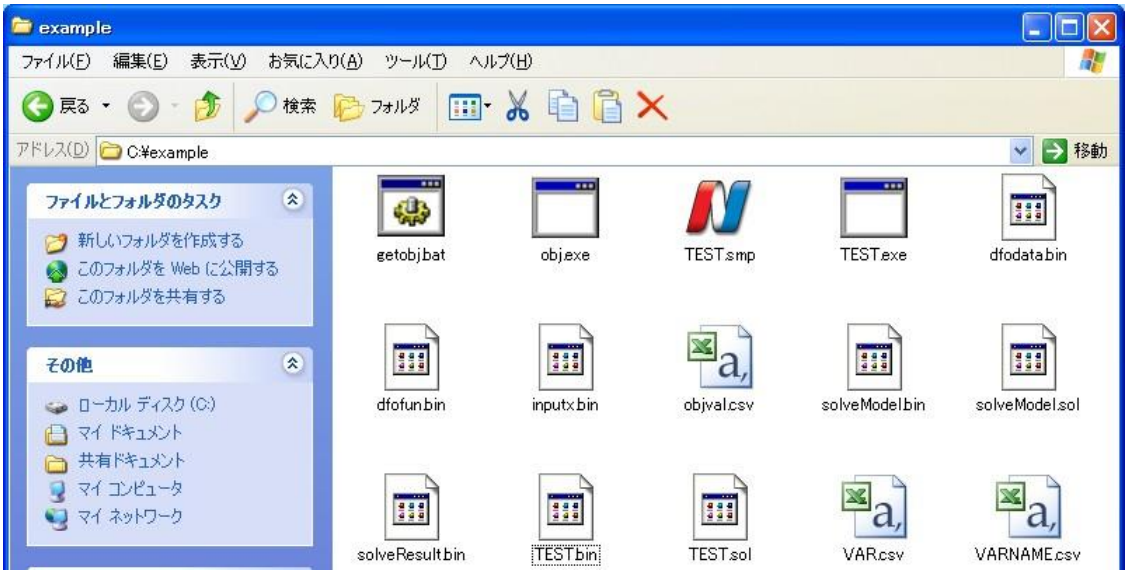
DFO.exe （モデル名） ＜データファイル名＞

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「6 Numerical Optimizer/DFO の実行方法と各機能の解説」を参照してください.

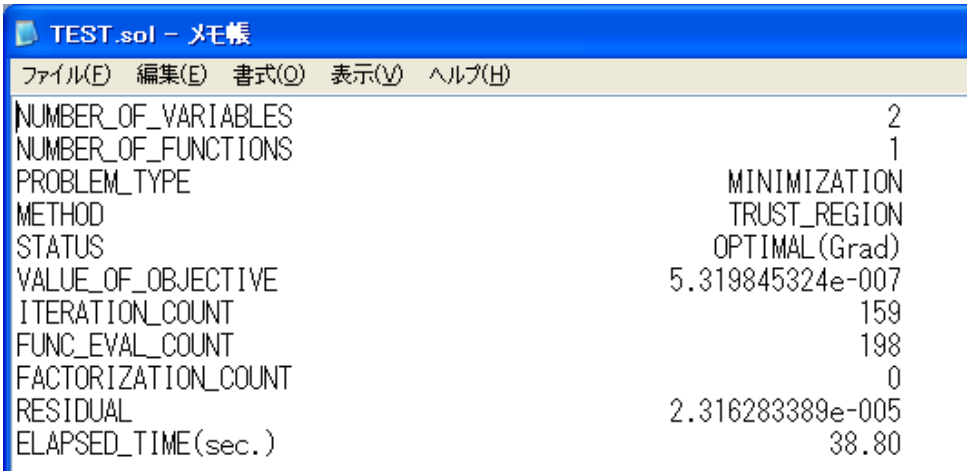
今回の例では，オプションもデータファイルも指定しませんので，コマンドプロンプトに次のように入力します.

すると，DFO.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.

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.



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，本利用ガイドでは主に Numerical Optimizer/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，「Numerical Optimizer/SIMPLE マニュアル」等をご参照ください．

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 dfoprog 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",dfoprog="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，Numerical Optimizer/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

Numerical Optimizer/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",dfoprog="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",dfoprog="DFO.bat");
f.entryVariable(y[i],i); // y[1],y[2],…,y[7] の順に登録される
```

最後に，添え字つきの変数の内，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",dfoprog="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 Numerical Optimizer/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「Numerical Optimizer/SIMPLE マニュアル」をご参照ください[8].

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

Numerical Optimizer/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[9].

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

- 信頼領域[10]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;
```

[8] 本利用ガイドに掲載されていないパラメータに関しましては，通常「部分問題に対するパラメータ」として認識されます.
[9] 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.
[10] 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;
```

- 初期値と上限（下限）制約との関係

上限（下限）が制約条件として課されている変数について，初期値と上限（下限）との近さを制御します．具体的には，初期値と上限（下限）との距離が options.dfoini_pcri で指

定した値より小さい場合，Numerical Optimizer/DFO では自動的に初期値を上限（下限）から少し離す処理を行ないます．なお，変数の上限（下限）以外の一般の制約条件については，このパラメータによる処理は行なわれません．

```
options.dfoini_pcri = 0.2;
```

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値

ペナルティパラメータの下限値を設定します．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. 変数の初期値の設定

Numerical Optimizer では変数の初期値を設定することが可能です.
例えば，変数 x の初期値を 1 に設定する場合には次のように記述します.

```
Variable x; // 変数 x の宣言
x = 1; // 変数 x の初期値を 1 に設定
```

また，添え字を持つ変数については初期値を一度に設定することが可能です．次の記述では y[1], y[2], y[3] の初期値を 3 に設定しています.

```
Set S="1 .. 3";
Element i(set=S);
Variable y(index=i);
y[i] = 3; // y[1],y[2],y[3] の初期値を一度に設定
```

ただし，変数の初期値を設定する際には以下の点にご注意ください.

- solve(); の前に記述する

モデルファイル内に solve(); を記述している場合，変数の初期値の設定は solve(); の前で行なう必要がございます．例えば，次のような記述をした場合，x の初期値は 1 となりますが，y の初期値は 2 とはなりません[11].

```
Variable x;
Variable y;
      （目的関数の宣言等は省略）
x = 1; // x の初期値は 1 に設定される
solve();
y = 2; // y の初期値は 2 に設定“されない”
```

- 上限（下限）に近い値や違反する値は初期値と出来ない

上限（下限）を制約条件として課した変数について，上限（下限）に近い値[12]や上限（下限）に関する制約条件に違反する値を初期値として設定することは出来ません．このような値を設定した場合は Numerical Optimizer/DFO 内部で調整を行ない上限（下限）から少しはなれた値を初期値とします.

---

[11] 問題の制約条件等によっては y の初期値が 2 となることがありますが，y = 2; の記述により設定されたわけではありません．また，x = 1 が x の上限（下限）に近い場合等では，1 以外の値が初期値となります.

[12] 上限（下限）に近いと判断する基準については「4.3.2 初期設定に関するパラメータ」でとりあげた options.dfoini_pcri で設定することが出来ます.

4.5. その他の注意点

以下に挙げる場合[13]に関しては，Numerical Optimizer/DFO で取り扱うことが出来ませんのでご了承ください.

- 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
- wcsp 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
- 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ござい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

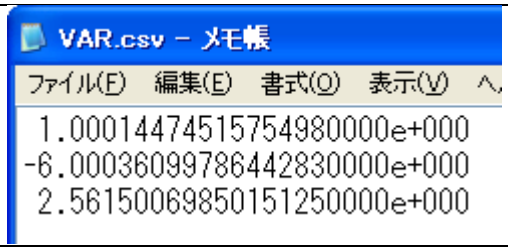
[13] 主なものを挙げています.

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[14]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ください．

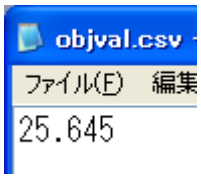
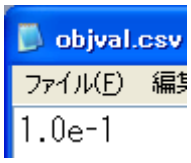
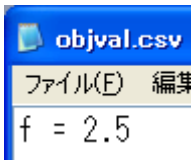
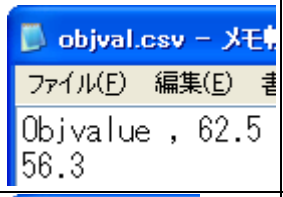
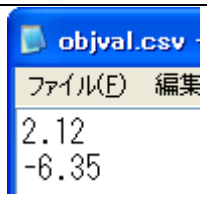
(1 番目に登録された変数の名前) (2 番目に登録された変数の名前) (3 番目に登録された変数の名前) ・・・・・	

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[15]．また，「数値（指数形式を含む）のみの行」が複数ある場

[14] 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．
[15] 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 Numerical Optimizer/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，Numerical Optimizer/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ください[16].

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
```

[16] /b オプションは使用しないでください.

6. Numerical Optimizer/DFO の実行方法と各機能の解説

Numerical Optimizer/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. Numerical Optimizer/DFO の一般的な最適化計算機能

本節では Numerical Optimizer/DFO の利用方法をご説明します．Windows 版では，Numerical Optimizer/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， Numerical Optimizer GUI (Visual Analytics Platform・Nuorium）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なわず終了となります.
-nocompile	モデルのコンパイルを行な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，Numerical Optimizer(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「Numerical Optimizer/SIMPLE マニュアル」をご参照ください.

（注意）Numerical Optimizer/DFO 用に作られた実行ファイルでは通常の Numerical Optimizer 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 Numerical Optimizer 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 Numerical Optimizer/DFO の機能を利用することが出来ません．なお， Numerical Optimizer/DFO 用の特別な処理を行なっていない状態で 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 Numerical Optimizer/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[17].

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

Numerical Optimizer/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＞
```

[17] 「4 モデリング言語 SIMPLE の記述方法」で説明した Numerical Optimizer/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.

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．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）[18]
- 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

Numerical Optimizer/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 というファイルを出力します．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.

[18]「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

Numerical Optimizer/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関しましては,（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

Numerical Optimizer/DFO に搭載のアルゴリズムの反復回数を表示します[19]．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 [19].

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

Numerical Optimizer/DFO による最適化計算の実行時間を表しています [19].

```
%%
%% 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.

[19] 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 Numerical Optimizer/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 Numerical Optimizer 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「Numerical Optimizer/SIMPLE マニュアル」をご参照ください．また，エラーメッセージに関して <<DFO **>> ではなく（DFO **）と表示されることがあります．

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>)	Numerical Optimizer/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ください．
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ください．
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 にご連絡ください．
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ご確認ください．
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ご確認ください．また，このエラーは同一フォルダ内で Numerical Optimizer/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)	変数の上下限値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.	Numerical Optimizer/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ください.
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ください．
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．Numerical Optimizer/DFO は IntegerVariable には対応しておりませんのでご了承ください．
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．Numerical Optimizer/DFO は DiscreteVariable には対応しておりませんのでご了承ください．
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. Numerical Optimizer/DFO は半正定値制約には対応しておりませんのでご了承ください．
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("dfoprog" 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．Numerical Optimizer/DFO では Objective の宣言は 1 回のみ行なうことが出来ます．
45	<<DFO 45>> Inappropriate method "<解法名>" applied to	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

	subproblem.	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な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． Numerical Optimizer/DFO は VariableParameter には対応しておりませんのでご了承ください.
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．Numerical Optimizer/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 Numerical Optimizer/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