

Xpress-Optimizer Reference Manual

Release 13

© Copyright Dash Associates 1984–2002

本マニュアル中の商標のうち Dash Associates のものでないものについては承認を受けてあります。このユーザーガイド内の全ての会社、製品、名前、データは架空のもので、Xpress-MP の使い方を説明するためだけのものです。名前やデータが実在のものに似ていても偶然の結果です。

Dash へのお問い合わせ

Xpress-MP の使用についてのご質問、ご意見は、以下のテクニカルサポートにご連絡ください。

Xpress-MP ソフトウェアの販売や注文についてのお問い合わせは、あなたの地域の販売所か以下の Dash 販売所にご連絡下さい。

最新のニュース、Xpress-MP ソフトウェアやドキュメントのアップデートについては、Xpress-MP のウェブサイトをご覧ください。

<http://www.dashoptimization.com/>

USA, Canada and The Americas Elsewhere

Dash Optimization Inc. Dash Optimization Ltd.
560 Sylvan Avenue Quinton Lodge, Binswood Avenue
Englewood Cliffs Leamington Spa
NJ 07632 Warwickshire CV32 5RX
USA UK

Telephone: (201) 567 9445 Telephone: +44 1926 315862

Fax: (201) 567 9443 Fax: +44 1926 315854

email: support-usa@dashoptimization.com email: support@dashoptimization.com

USA, Canada and The Americas Elsewhere

Dash Optimization Inc. Dash Optimization Ltd.
560 Sylvan Avenue Blisworth House, Church Lane
Englewood Cliffs Blisworth
NJ 07632 Northants NN7 3BX
USA UK

Telephone: (201) 567 9445 Telephone: +44 1604 858993

Fax: (201) 567 9443 Fax: +44 1604 858147

email: sales@dashoptimization.com email: sales@dashoptimization.com

Xpress-Optimizer Reference Manual Contents i

目次

1 はじめに 1

1.1 概要	1
1.2 Xpress-Optimizer	2
1.3 整数計画法	2
1.4 Xpress-Optimizer の実行	4
1.5 本マニュアルの構成	6
1.6 表記法	6

2 Xpress-MP による問題の解決 9

2.1 概要	9
2.2 初期化と終了	9
2.3 問題を読む	10
2.4 問題を解く	11
2.5 結果を見る	11
2.6 最適化の例	12
2.7 クイックリファレンス	12
3 最適解，実行不能，非有界 15	
3.1 求解過程	15
3.2 実行不能	16
3.3 非有界	17
3.4 スケーリング	17
4 求解アルゴリズムの性能 19	
4.1 アルゴリズムの選定	19
4.2 シンプレックス法の性能	19TENSACU
4.3 パリア法の性能	21
4.4 整数計画法の性能	23
5 アルゴリズムの実行 29	
5.1 行列を見る，修正する	29
5.2 事前分析のはたらき	31
5.3 コールバックを使う	33
5.4 カットマネージャのはたらき	34
5.5 目標計画法	37
6 コンソールとライブラリ関数 41	
6.1 コンソールモード関数	41
Xpress-Optimizer Reference Manual Contents ii	
6.2 関数の記述要領	42
7 制御パラメータ 255	
7.1 制御パラメータの扱い方	255
8 問題特性 301	
8.1 問題特性の検索	301
9 エラーメッセージ&リターンコード 317	
9.1 Xpress-MP Optimizerのリターンコード	317
9.2 Optimizer のエラーコード	318
A ログとファイル形式 339	
A.1 ファイル形式	339
A.2 XMPS 行列ファイル	340
A.3 ASCII 解ファイル	348
A.4 ASCII レンジファイル	353
A.5 ディレクティブファイル (.dir)	357
A.6 行列変更ファイル (.alt)	358
A.7 グローバルログ	359
索引 361	
Xpress-Optimizer Reference Manual Overview 1	

1 はじめに

1.1 概要

Xpress-OptimizerはXpress-MPソフトウェアセットの一部で、様々な最適化問題に適用できる強力なツールです。コンソールやグラフィカルインタフェースにより、またライブラリ関数によって直接アクセスでき、Optimizerは最良の結果を得るために高速で柔軟性のある簡単な機能を備えています。最高技術を駆使したアクセスルーチンや最適化アルゴリズムは、解くのにかなりの時間がかかりそうな大規模な問題の攻略を可能にします。プログラム単体で、あるいはユーザ自身のプログラムにおいてタスクを処理する場合、Xpress-Optimizerは、無限の可能性の世界への扉を開きます。コンソールXpressユーザの場合、「コンソールモード」にはOptimizerの重要な核となる機能を使用するためのたくさんのコマンドがあります。MPSやLP形式ファイルからモデルは読み込まれ、Optimizerにサポートされたアルゴリズムで最適化され、様々な方法で解を見ることができます。更に、アルゴリズムに様々な影響を与える制御パラメータの設定により、最適化プロセスを制御できます。

Xpress-MP Optimizer ライブラリのユーザは、更に「アドバンスモード」を利用することで、コンソールモードの全ての機能が使えます。このインタフェースの拡張により、Optimizerの内部データ構造へアクセスできたり、大規模な行列の管理ができます。カット面の追加ができるカットマネージャは、混合型整数計画問題を効率的に解くための Tailor-made な戦略の実現性を提供します。

しかしながら、ライブラリの一番の利点は、ユーザが自分のプログラムでOptimizerを呼びだし、自身のアプリケーションとXpress-Optimizerのパワーを連結し、上記のことを自動的にに行えることです。C, Fortran, Java, VisualBasicを含む汎用言語で、ライブラリ関数が使えます。表記法についてC言語の点から説明しますが、他の言語の場合の相違点や簡単な例はXpress-MP Essentialsマニュアルに載っています。

Xpress-Optimizer Reference Manual The Xpress-Optimizer 2

Introduction 1

1.2 Xpress-Optimizer

Xpress-Optimizer は、行列構造を最大限に利用するよう、綿密にプログラムされた最適化ツールです。整数計画法については、多様な離散要素、すなわち実行可能解で非線形性を満たさなくてはならない要素を扱えます。ここではそのような解を、学術用語としての厳密な正確さは別にして、整数解と呼びます。

Optimizerは、次の3つの方法の中の1つにより問題を読み込みます。

- MPS形式の行列ファイル
- バイナリ形式 (Xpress-Modelerにより作成したBIF形式) ファイル
- LP形式の行列ファイル

Optimizer は、主・双対シンプレックス法による線形最適化、あるいはライセンスを持っているならばニュートンバリア法で最適化を行います。整数解の探索法を指示することができます。レンジ (感度分析) も実施可能です。通常、Optimizer は解を解ファイル (.sol) に書き出します。表計算やデータベースソフトとのインタフェースとして ASCII 形式の解ファイルを作成できます。

Optimizer は、また、途中の解を基底ファイルの形式で保存できます。ある線形問題の最適解や変数の状態を基底ファイルとして保存しておき、よく似たモデルのシンプレックス LP 解のパスは、出発点としてその基底ファイルを使うと、短くなる場合があります。

MPS形式やLP行列ファイルは、エディターで調べたり修正できるASCIIファイルですが、解ファイル、レンジファイル、workファイルはバイナリ形式に圧縮されているため、エディターで見ることはできません。これらのファイルの説明と表記法については、参考資料A「ログとファイル形式」をご覧ください。

1.3 整数計画法

線形計画法で精度よくモデル化できる問題は多数ありますが、決定問題の中核に離散的な決定要素のある状況があります。このような非線形性の機能が必要なものには、主として以下の3つがあります。

- 離散的な集合の中から要素を選ぶ
Xpress-Optimizer Reference Manual Integer Programming Considerations 3

Introduction 1

- 論理的な関係を表す
- 関数の大域的最適解を求める

このような問題は Xpress-Mosel , Xpress-BCL , Xpress-Modeler のいずれかを使うことでモデル化できます。ライセンスがあれば、Optimizer で大域的最適解を求めることができます。普通、基本的構造は線形ですが、非線形性がいくつかの変数に関して分離型のときも、うまく使えます。

Xpress-MPには、大域的最適化のために以下の道具がそろっています。

2進変数(BV)	— 0か1のどちらかをとる変数。0-1型変数
整数変数(UI)	— 整数値のみを取る変数。上限を指定する必要があります
部分的整数変数(PI)	— 指定された限界までは整数値を取り、それ以上は任意の値を取る変数
準連続変数(SC)	— 0かある範囲の値をとる変数。変数が全て使われるか、最低限以上使う必要のある状況のモデルに使われます
準連続整数変数(SI)	— 0、またはある範囲の値を取る整数変数
タイプ1の特殊順序集合(SOS1)	— 変数の順序付き集合で、0以外の値を取る変数が高々1つのもの
タイプ2の特殊順序集合(SOS2)	— 変数の順序付き集合で、0以外の値を取る変数が高々2つで、2つあるときは変数が集合の中で隣り合っているもの

Optimizer は最適解を見つけ、最適であることを確認するために分枝限定法を使用します。探索法については、多くの問題でうまくいった方法をデフォルトとして設定してあります。しかしながら、大型の MIP 問題ではモデル化の知識が非常に重要です。大型の実用モデルを解く際に、小型化した例題で注意深く実験してみることは、常に有効であると思われます。

(事前分析を除いて) Optimizerにおける整数計画法の機能は、LP問題としての最適化の後に用いられます。入力データにあった非線形性や離散性は、ここで考慮されます。種々のインタフェイスにおけるOptimizerによる整数計画の使用例は、すでにリファレンスマニュアルの前に読んだとは思いますが、Xpress-MP Essentialsに示してあります。

Xpress-Optimizer Reference Manual Running the Xpress-Optimizer 4

Introduction 1

1.4 Xpress-Optimizer の実行

初期化

Optimizer を実行するには有効なパスワードファイル `xpress.pwd` が必要です。これは、特定の機械、イーサネットアドレス、ドングルによって作られます。これらのどれかが存在しない場合には、「No purchased authorization found」というメッセージが画面表示され、「student mode (学生モード)」に入ります。そして、この制限のある状態で動作を続けます。Optimizer はパスワードファイルを Optimizer のあるローカルディレクトリで探します。そこになれば、環境変数 `XPRESS` に指定されたディレクトリを探します。パスワードファイルは Xpress-MP DLLs, executables と同じディレクトリに、常時置いておく必要があります、`XPRESS` 変数を設定する必要なく見つけられることに注意してください。

続いて、Optimizer は問題に名前をつけます。問題名を指定しなければ、コンソールユーザは、指定するように促されますが、それを無視すると、デフォルトの問題名 `$$$$$$$$` となります。ユーザに指定された問題名については、ドライブ、パスとファイル名だけが受け付けられ、拡張子は無視されます。

問題名 `problem_name` は Optimizer が作成する各拡張子で区別された種々のファイル全ての基本として使われます。行列 (MPS) ファイルは `problem_name.mat`、バイナリのインターフェイスファイルは `problem_name.bif`、解ファイルは `problem_name.sol` です。

問題名が決まると、Optimizer は `optimizer.ini` という初期化ファイルを読みこみます。これは、Optimizer の実行形態にかかわらず、デフォルトで使用される記述やコマンドを含む ASCII 形式のファイルです。このコマンドは制御パラメータの設定や表示に限ることなく、行列入力、最適化等、他の一般的なコマンドを含むことができます。初期化ファイルが存在しない、もしくは見つからない場合、Optimizer は組み込みのデフォルトで初期化されます。

コンソール Xpress オプション

コンソール Xpress ユーザの場合、以下のようにして Optimizer を呼びだします。

```
C:\> optimizer [problem_name] [@filename]
```

Optimizer が起動される際、問題名はオプションでコマンドラインに指定できますが、省略すると、上記のように促されます。

Xpress-Optimizer Reference Manual Running the Xpress-Optimizer 5

Introduction 1

また、コマンドライン上で、@に続いて指定されたセカンドファイルからコマンドを読んでコンソール Xpress をバッチモードで実行することができます。@とセカンドファイル名の間にはスペースを入れられません。このようにすると、コマンドがセカンドファイルからキーボード入力のように読まれます。一見これはリダイレクト機能 (つまり、<) のようですが、@機能はバッチモード動作で他のプログラムから Optimizer が呼びだされるときに使えます。

バッチファイルがある場合、それは完結している必要があります。キーボードに戻ってコマンドの続きを入力することはできません。STOP や QUIT コマンドがファイル中には必要で、そうしないとシステムはハングアップします。

バッチファイルがコマンドラインに指定されない場合は、初期化処理の後、Optimizer のプロンプト `>` が表示され、標準入力からコマンドが読まれます。

コンソール Xpress での最適化実行の中止

Optimizer のコンソール Xpress ユーザの場合、CTRL-C を入力すると、Optimizer は実行中のコマンドを中断し、プロンプト `>` に戻ります。この方法で中断しようとする、安全な状態で計算を実行した後、終了し、プロンプトに戻ります。その時の解は解ファイルに保存され、

後で中断したコマンドを再入力すると、その時点から繰返しを再開します。
次のコマンドが先回りしてタイプしてあった場合、OSによってはCTRL-Cが使えないことがあります。

終了

Optimizerには次の3種類の終わり方があります。

- (a) エラーのない正常終了
- (b) 解の状態を示すエラーコードを伴う終了
- (c) 致命的なエラーによる即時終了

Xpress-Optimizer Reference Manual Structure of this Manual 6

Introduction 1

正常終了は、QUIT や STOP コマンドで正常に終了したことを示します。ただし、コマンドの実行結果や致命的でないエラーがあったかによるので、問題が解けて最適解が得られたかどうかは分かりません。

コンソール Xpress ユーザが STOP コマンドで Optimizer を終了する場合、解の状態を示す 0 でないリターンコードが返されます。リターンコードのリストについては9章「エラーメッセージ&リターンコード」をご覧ください。

致命的なエラーが発生すると、Optimizerは直ちに停止します。これは、必要なファイルが見つからない、ディスク容量に空きがない、メモリが足りないといった場合に起きます。

1.5 本マニュアルの構成

マニュアルは主に2つの部分で構成されています。まず、2章「Xpress-MPによる問題の解決」には、Optimizerの基本的な使い方に関する概要が書かれており、よく使うルーチンやその設定方法が紹介されています。次に3章「最適解、実行不能、非有界」では、最適化の過程で発生する問題や問題から得られる解についての概要を示してあります。4章「求解アルゴリズムの性能」には、頻繁に使う制御の説明が、求解過程を高度化する方法のいくつかの考えと一緒に説明してあります。最後に、5章「アルゴリズムの実行」には、Optimizerの使用方法についての、より高度な話題が詳述されています。

以上の5章に続いて、リファレンスを示す3章があります。6章「コンソールとライブラリ関数」にはコンソール、アドバンスモードの全関数をアルファベット順に詳述してあります。7章「制御パラメータ」と8章「問題特性」には様々な制御と特徴についてのリファレンスがあり、9章「エラーメッセージ&リターンコード」には、Optimizerのエラーとリターンコードのリストがあります。そして、ファイルフォーマットについての説明を付録に示します。

1.6 表記法

マニュアルでは以下のような標準表記法を用います。

コンピュータコードフラグはfixed width fontで、数式や変数はイタリック体で表してあります。ライブラリ関数の拡張については、C言語を用いて説明され、配列等のような構造については、Cの言語表記が用いられます。

Xpress-Optimizer Reference Manual Conventions Used 7

Introduction 1

コンソールモードルーチンについては、コンソール、ライブラリとも、通常、コンソールコマンドが括弧に入れられた形で書かれます。コンソールユーザに関係ない章では、ライブラリフォームのみで示されます。コンソールとライブラリの間のコマンドや制御の文法的な違いは、

6章「コンソールとライブラリ関数」をご覧ください。以下の表記法については、適宜使用します。

- 角括弧 [...] はオプションを囲みます
- 波括弧 {...} は選択された1つのオプションを囲みます
- 式の中のイタリック体はメタ変数を表します。メタ変数に続く説明に使用方法を示します
- 「.」は、「リターンキーあるいはエンターキーを押す」という意味です。通常、大きなキーのマークです
- CTRLに1文字続くものは、「CTRLキーを押した状態で、その文字を入力する」の意味です。CTRL, コントロールキーは、通常、キーボードの左下隅にあります
- 縦棒の「|」は、多くのキーボードでは、画面上では、しばしば中央の隙間がない状態で表示され、混乱します。UNIXでは、これをパイプと言います。PC画面上で箱を描くのに時々使われる文字とは違うので注意してください。ASCII形式では、「|」はヘキサコード「7C」, 10進法「124」です。

Xpress-Optimizer Reference Manual Conventions Used 8

Introduction 1

Xpress-Optimizer Reference Manual Overview 9

Problem-Solving with

Xpress-MP 2

2 Xpress-MP による問題の解法

2.1 概要

Xpress-Optimizer では、コンソールモードコマンドによる中核的機能の操作、アドバンスドモードルーチンによる内部データ構造への広範囲なアクセスが提供され、広く様々なルーチンが使えます。Xpress-Optimizer のユーザは様々で、扱う問題も広範囲に渡りますが、ほとんどの最適化は、ルーチンを「自然にグループ化」し、標準テンプレートで実行されます。ここでは、このことについて示します。

一般的な最適化の過程を図1に示します。一般に、問題を解く場合には、5つの主要なプロセスを経ます。Optimizerは、問題を解くにあたり、まず初期化されます。次に、その問題の行列がメモリに読み込まれ、問題を解く最適化ルーチンの1つが呼びだされます。そして、その問題の解を確認し、最後にユーザはOptimizerを終了します。この構成は、複数の問題を同時に操作できるというOptimizerの能力が使われるとより複雑になるかもしれませんが、コマンドを構築する上では非常に有用です。

図1 問題の最適化

Initialization Read in a Optimize
 Problem Problem
 View the
 Solution
 Termination

2.2 初期化と終了

Optimizer

Optimizer を使う際、まず、初期化の過程を経ます。コンソール Xpress では、この処理は自動的に行なわれますが、ライブラリユーザの場合、他のライブラリルーチンの前に XPRSinit

を呼びだして、手動でシステムを初期化しなくてはなりません。このコマンドは必要なライブラリをチェックし、あなたの Xpress-MP のライセンスを確認します。一旦、このチェックが終われば、他のすべてのルーチンが動作可能となります。

セッションの終わりでOptimizerルーチンの全ての呼びだしが完了すると、終了処理に移ります。ここでは、Optimizerによって使われたメモリが他の処理のために解放され、読み書きするために開かれていたファイルが閉じられます。コンソールユーザの場合、QUITやSTOPコマンドが与えられると、この処理は自動的に実行されます。ライブラリユーザは、同様の処理を実行するためにXPRSfreeルーチンを呼びます。

Xpress-Optimizer Reference Manual Reading in a Problem 10

Problem-Solving with

Xpress-MP 2

問題環境

ライブラリユーザの場合、複数の問題環境を活用するためには、一度Optimizerを初期化し、問題の行列が読み込まれる前にXPRScreateprobルーチンを用いて問題環境が作られなくてはなりません。（ライセンスの範囲で）これらの環境はいくつでも作ることができ、最適化、保存、実行というプロセスは与えられた任意の回数だけ実行可能となります。これは、各関数を呼び出す際に最初の引数として問題ポインタを指定することによって、実施可能となります。このような環境が必要でなくなったときには、XPRSdestroyprobを用いて取り除くことができます。全ての環境を取り除くときには、XPRSfreeを呼び出す前に実行してください。

Optimizerの出力

コンソールXpressユーザの場合、出力は直接、画面とログファイルに送られます。一方、ライブラリユーザの場合、更にフレキシブルな出力が可能です。コマンドXPRSsetlogfileにより、特定の問題に関する出力にログファイルを特化できます。あるいは、XPRSsetcbmessageコマンドにより定義される出力コールバック関数を用いることで、出力を受け取って、アプリケーション内で直接処理することが可能です。詳しくは5.3節「コールバックを使う」を参照してください。

2.3 問題を読み込む

初期化が終わると、Optimizerは問題を読み込みます。問題は基本的に2つの方法でOptimizerデータ構造に読み込まれます。シンプルなのは、XPRSreadprob (READPROB)を用いたMPSまたはLPファイルからの行列の読み込みです。ライブラリユーザの場合は、XPRSloadlp, XPRSloadqp, XPRSloadglobal, XPRSloadqglobalを用いて問題をデータ構造からロードすることも可能です。そしてMPS名をXPRSaddnamesを用いてモデルに付加することができます。問題がロードされると、それに続いて呼び出す入力ルーチンにより、問題は上書きされます。以下のことは重要です。MIP問題において、原問題の行列は、求解アルゴリズムによって、もとの状態に戻せなくなるように書き換えられることがあります。つまり、最適化処理が始まると、原問題を復帰できると思うべきではありません。恐らく最も重要な事は、事前分析されたMIP行列は、XPRSglobal (GLOBAL)を呼び出した後には、決して事後分析されないことです。そして、概して、その問題を変更したり再度使うのであれば、原問題の行列を再ロードするか、複製して保存する必要があります。詳しくは、5.2節「事前分析のはたらき」を参照ください。

Xpress-Optimizer Reference Manual Solving the Problem 11

Problem-Solving with

Xpress-MP 2

2.4 問題を解く

問題のロード後、その問題を解く（解こうとする）ことができます。XPRsmaxim (MAXIM), XPRsminim (MINIM) コマンドにより LP, QP, MIP (混合型整数計画) 問題の最大化や最小化が可能です。ここで、MIP 問題については、LP 緩和問題を解いた後、コマンド XPRsglobal (GLOBAL) により解きます。また、XPRsrange (RANGE) を使うことでレンジング (感度分析) ができます。

他にも様々なことができます。シンプレックスLP最適化処理は、事前に類似の問題から得られた最適基底から開始することで、高速化できます。これらの基底はXPRswritebasis (WRITEBASIS), XPRsreadbasis (READBASIS), XPRsloadbasis を用いてユーザデータ構造からの入力を用いて保存、復元することができます。また、整数探索方向を指定でき、それはXPRsreaddir (READDIRS) によりファイルからロード可能です。ディレクティブファイルの形式については、付録A「ログとファイル形式」に説明があります。他にも高度な機能としてコールバック関数やカットマネージャのカスタマイズができます。これらについては、5章「アルゴリズムの実行」に説明があります。

2.5 結果を見る

最適化が終了すると、解を得られるか、問題が実行不能、非有界であることを確認できるかのいずれかです。後者の場合、ユーザはどのようにして問題がそうなったのかを知り、それを修正したいと思うでしょう。これについては、後程、3章「最適解、実行不能、非有界」で触れます。前者の場合、ユーザは解について詳しく知りたいと思うでしょう。

Xpress-MPには解かれた問題の詳細にアクセスするためのたくさんの機能があります。

XPRswritesol (WRITESOL) や XPRswriteprtsol (WRITEPRTSOL) を使うことで、完全な解をASCIIファイルで得られます。XPRswritesol (WRITESOL) は他のアプリケーションとの互換性に優れています。XPRswriteprtsol (WRITEPRTSOL) は主にプリンタに送るのに適しています。レンジ情報を得るための同様のコマンドは、XPRswriterange (WRITERANGE) と XPRswriteprtrange (WRITEPRTRANGE) です。ライブラリユーザはまた、XPRsgetsol を使うことで自分たちのプログラムから直接解にアクセスすることができます。この場合、最適な決定変数の値やスラック変数、双対値、リデューストコストが返されます。更に、最適化アルゴリズムは、*problem attributes* (問題特性) を用いて、解かれた問題についての更に詳しい情報を出力します。これらは、特定の情報について問い合わせられたり、ライブラリ関数 XPRsgetintattrib, XPRsgetdblattrib, XPRsgetstrattrib を使うタイプのユーザの変数に関して提供されるオブジェクトの集合です。特性の例としては、LPや(M)IP問題の目的関数の最適値を各々返すLPOBJVAL, MIPOBJVALなどが含まれます。特性のリストについては、8章「問題特性」を参照してください。

Xpress-Optimizer Reference Manual Optimization by Example 12

Problem-Solving with

Xpress-MP 2

2.6 最適化の例

これまで、Xpress-Optimizer の一般的な特徴と使い方について簡単に紹介しました。以下の節では、よくある問題や Optimizer の性能のような話題を Optimizer のより高度な特徴と共に考えます。また、種々の関数や使い方についてのリファレンスを本マニュアルの後半部分に示します。

Optimizerの使用例としては、多くのソースプログラムが、特にXpress-MP Essentialsにおいて利用可能です。Xpress-MP EssentialsはXpress-MPソフトウェアを着実に使いこなすための基本

となるものであり、リファレンスマニュアルを読む前に適切な章を読むことを強くお勧めします。より高度な例は、Xpress-MP CD-ROMにあります。

2.7 クイックリファレンス

初期化と終了

XPRSinit	Optimizerを初期化する
XPRScreateprob	問題環境を作成する
XPRSsetlogfile	Optimizerの全出力を直接ログファイルへ書き出す
XPRSsetcbmessage	出力コールバック関数を定義する
XPRSdestroyprob	問題環境を破棄する
XPRSfree	Optimizerにより使用されたメモリを解放し、全ての開いているファイルを閉じる

Xpress-Optimizer Reference Manual Quick Reference 13

Problem-Solving with

Xpress-MP 2

問題を読む

問題を解く

結果を見る

XPRSreadprob	MPS, LP, QPまたはバイナリ形式ファイルを読み込む
XPRSloadlp	LP問題をOptimizerにロードする
XPRSloadqp	QP問題をOptimizerにロードする
XPRSloadglobal	MIP問題をOptimizerにロードする
XPRSloadqglobal	2次のMIP問題をOptimizerにロードする
XPRSaddnames	行や列のレンジに名前を付ける
XPRSreadbasis	ファイルから基底を読み込む
XPRSloadbasis	ユーザデータ構造から基底をロードする
XPRSreaddirs	ディレクティブファイルを読み込む
XPRSmaxim	問題の目的関数を最大化する
XPRSminim	問題の目的関数を最小化する
XPRSglobal	整数解を探索する
XPRSrange	レンジ情報を計算し、ファイルに保存する
XPRSgetbasis	現在の基底をユーザ配列に戻す
XPRSwritebasis	基底をファイルに書き込む
XPRSwritesol	現在の解をASCIIファイルに書き込む
XPRSwriteprtsol	現在の解を印刷可能な形式でファイルに書き込む
XPRSwriterange	レンジ情報をASCIIファイルに書き込む
XPRSwriteprtrange	レンジ情報を印刷可能な形式でファイルに書き込む
XPRSgetsol	解の値をユーザ配列に戻す
XPRSgetintattrib	整数型問題の属性の値を回復する
XPRSgetdblattrib	倍精度型問題の属性の値を回復する
XPRSgetstrattrib	文字列問題の属性の値を回復する

Xpress-Optimizer Reference Manual Quick Reference 14

Problem-Solving with

Xpress-MP 2

& Unboundedness 3

3 最適解, 実行不能, 非有界

3.1 求解過程

Optimizer に問題の行列が作られると、一般にユーザはそれを解くために最適化ルーチンの 1 つを呼びだし、有用な最適解を得ます。しかし、最適化はいつもこのように単純なものではなく、求解の過程でいくつかの問題が起こり得ます。実際の場合では、しばしば問題規模が大きく、解くのに長時間を要することがあります。このことは、複雑な MIP 問題の場合に特にそうです。解くのに長時間を要しそうな問題において、最適化に要する時間は制御パラメータの注意深い設定や工夫により、しばしば改善されます。以下ではこのことについて示します。コンソールユーザは CTRL-C キーを押すことで、いつでも最適化計算を安全に終了できます。

Optimizer が終了する前に、そのときの最適化計算の現在位置が保存できます。そして、必要があればコマンドを再入力することで、計算を継続できます。ライブラリユーザはこの機能を使えませんが、シンプレックス反復計算の実行回数の上限を指定する `LPITERLIMIT`、あるいはニュートンバリア法において同様の指定をする `BARITERLIMIT` を設定すれば、最適化の計算量を制限できます。また、制御 `MAXTIME` により、実計算時間限度を設定できます。探索を早く終えたい場合には、これら 3 つの制御が適切な値に設定されているかを調査してみるべきです。また、最適化を中断させるものは 1 つとは限りません。

これらの問題が生じることなく、無事に最適化が終了する場合には、得られた解について有用な出力をたくさん得ることができます。理想的には、問題が適切で、その解が有用であると確認できると最適解が得られたことになるでしょう。しかしながら、これはこのような場合のみみ得られるものではありません。問題が実行不能で解が見つからない場合も同じように可能です。また、モデルによって無限解が得られる非有界である場合についても同様です。ここでは、これらの可能性について若干触れます。

& Unboundedness 3

3.2 実行不能

全ての制約条件を満足するような解が存在しない場合、その問題を実行不能と言います。

Xpress-Optimizerには、求解過程における実行不能な箇所を発見し、モデルの実行不能性について診断する様々な方法があります。

事前分析における実行不能性の診断

事前分析機能を使うことで (5.2節「事前分析のはたらき」参照)、実行不能性について様々なチェックができます。実行不能が検出された場合、その過程を遡ってトレースしたり原因を調べることができます。この診断は、最適化ルーチン (`XPRSminim(MINIM)`や`XPRSmaxim(MAXIM)`) が呼ばれる前に、制御パラメータ`TRACE`を 1 に設定しておけば、いつでも実施されます。この場合、実行不能の原因が最適化ルーチンからの出力の一部としてレポートされ、必要に応じて問題を修正できます。

既約実行不可能集合

実行不能性を分析するための一般的な方法に、それ自身が実行不能行列である部分小行列を発見することがあります。Optimizer は既約実行不可能集合 (IISs)を見つけることで、これを実行

します。IIS は実行不能な制約と変数の上下限値の最小集合で、その中の制約や上下限のどれかを取り除くと実行可能となります。

モデルによっては、複数の実行不能が存在することがあります。1つの IIS を修正するだけでは、モデルを実行可能にできないかもしれません。そのため、Optimizer は XPRS*iis* (IIS)によってモデル中の各実行不能についての IIS を見つけることができます。この探索はパラメータ MAXIIS により制御され、これは発見する IIS の最大数を制御します。実行不能が異なる IIS によって表されることがあり、Xpress-MP は実行不能性を簡単に診断できるように最小制約数の IIS を探します。ライブラリユーザの場合、IIS については XPRS*getiis* を用いることで探せます。

IISが見つかる、1つの制約や上下限を取り除くことで、IISによる実行不能を完全に解消できるかを知ることは有用です。よって、*sub-IIS isolation* (分離sub-IIS) と呼ばれる IIS の部分集合を見つけることを試みます。分離sub-IISのいずれかの要素を取り除くと、他のIISの実行不能性を増加させることなく、そのIISの全ての実行不能性を取り除きます。それゆえ、分離sub-IISはそれぞれが独立した実行不能の原因である可能性が高く、取り除くべき制約や上下限を示します。分離sub-IISを発見することは常に可能ではありませんが、見つければ、要素が星印でマークされます。分離sub-IISは全ての独立したIISが見つかるまで探されます。

Xpress-Optimizer Reference Manual Unboundedness 17

Optimality, Infeasibility

& Unboundedness **3**

整数実行不能

ある種のMIP問題の中には、LP (緩和) 問題では最適解が存在するのに、実行不能となる場合があります。この場合、LP緩和問題の実行可能領域は、困ったことに、整数条件を満足する解を含まず、これら条件の1つを除くか他の制約条件の1つを除くしか解を得る方法がありません。この問題は、恐らく最も難しい種類の実行不能問題であり、その原因を特定することは容易ではありません。

3.3 非有界

モデルの制約条件を満たしながら、目的関数が無限によくくなるような問題を *unbounded* (非有界) と呼びます。この場合、普通、無限に利益が得られるというより、解かれたモデルの定式化に問題があることを示しています。Optimizerが非有界を発見した場合、ユーザはそのモデルについて、不足しているもの、不十分なもの、不正確な制約条件、データについて確認する必要があります。

3.4 スケーリング

時として、制約条件や目的関数内で桁が大変大きく変化する問題を解こうとすることがあります。このようなモデルは悪いスケールのモデルです。簡単な例を示します。

最適化の過程において、Optimizerは制約条件や目的関数を用いて多数の引き算や割り算を実施します。これらの計算が、桁が大きく違う状態で実施されると、コンピュータの計算精度や Xpress-MP で用いられる固定許容範囲のために誤差が蓄積され、信頼可能な最適解が見つけれなくなります。

最大化:

制約条件:

$z = 10^6x + 7y + \dots$

$106x + 0.1y + 100$.
 $107x + 8y + 500$.
 $1012x + 106y + 50 + 106$.

Xpress-Optimizer Reference Manual Scaling 18

Optimality, Infeasibility

& Unboundedness 3

この悪影響を最小化するために、問題は Optimizer に入れられる前に、できる限りスケールリングされるべきです。上記の例では、係数と最後の制約条件がスケールリングできます。前者による問題は、列スケールリングや効果的な座標変換を新変数による置換と共に実施することで解決できるかもしれません。後者による問題については、行スケールリングにより解決できるかもしれません。

モデルの定式化の際に、よくスケールリングされた行列の作成に多くの時間をかけることは、大変有効です。これはいつも簡単にできることではないため、Xpress-MP には、これを簡単に実施するためのスケールリングのオプションが、いくつか組み込まれています。用いるスケールリングの形式は、制御パラメータ SCALING に設定されるビットによります。スケールリングの特別な形を用いたいときは、要求するスケールリングに相当する値の合計を SCALING に設定します。例えば、行、列のスケールリングを行い、また再度、行をスケールリングする場合、SCALING をそのように設定します。XPRSreadprob (READPROB) の間に問題はスケールリングされ、その後、XPRSscale (SCALE) により再スケールリングされます。一般に、[XPRSalter (ALTER) を使った] 変更が続いて、あるいは [XPRSrecurse (RECURSE) を使った] 逐次線形計画の間に行列を再スケールリングしたいときがあります。XPRSscale (SCALE) を呼び出す前に SCALING を設定すれば、通常とは違うスケールリング戦略を実施できます。

SCALING のデフォルト値は 35 で、最大要素法により行、列をスケールリングします。スケールリングの必要がない場合には、XPRSreadprob (READPROB) を呼び出す前に SCALING を 0 にしてください。

スケールリングは行列要素によってのみ決定されます。目的係数、右辺値、上下限値はスケールリングに影響しません。非零 SCALING は LP 変数をスケールリングしますが、整数要素のスケールリングは XPRSglobal (GLOBAL) ではサポートされていません。つまり、整数要素のスケールリングは MIP 問題の定式化の際に注意深く考えられなくてはなりません。

ビット	値	スケールリング方法
0	1	行のスケールリング
1	2	列のスケールリング
2	4	再度の行のスケールリング
3	8	maximin 法
4	16	Curtis-Reid 法
5	32	0 =>幾何平均でのスケールリング 1 =>最大要素でのスケールリング (maximin法とCurtis-Reid法では使えません)

x
 $106x$
 $124++7=$

Xpress-Optimizer Reference Manual Choice of Algorithm 19

Performance Issues 4

4 求解アルゴリズムの性能

4.1 アルゴリズムの選定

Xpress-Optimizer は、基本的に 3 つのソルバーが 1 つになったものであり、LP や QP の問題を解くために、主シンプレックスアルゴリズム、双対シンプレックスアルゴリズム、ニュートンバリア内点法からユーザは選びます。与えられる状況における最も適切な最適化アルゴリズムは問題に依存します。一般に、モデルが実行不能でない、または実行不能に近いのであれば、通常、双対シンプレックス法の方が主シンプレックス法より、かなり高速です。問題が実行不能なようであれば、主シンプレックス法が恐らく一番よく、実行不能の原因を簡単に求められます。ニュートンバリア法のような内点法は、問題の行列が A で、行列 ATA が密な場合、バリアが遅くなりますが、あるクラスの問題では、よいパフォーマンスを示します。

デフォルトでは、Xpress-Optimizer は LP 問題を解くのに双対シンプレックス法を用い、QP 問題を解くのにバリア法を用います。ほとんどのユーザには、これで十分であり、このことを変更する必要はありません。しかしながら、問題を解くのに余りに長時間を要しそうであれば、別のアルゴリズムを試すのもよいと思われます。このことは、最適化ルーチン `XPRSmaxim` (MAXIM)、`XPRSminim` (MINIM) にフラグを立てて指定できます。デフォルトで使われるアルゴリズムは制御パラメータ `DEFAULTALG` の値で決められます。

以下では、これらの方法に関する性能と整数解の探索について詳しく示します。

4.2 シンプレックス法の性能

シンプレックス法

線形制約の集合で定義される領域は実行可能領域として知られる多面体です。同じ線形目的関数値を有する等高線集合である点は超平面を作ります。目的関数の最適値は、実行可能領域の境界上のみ存在し、常にその多面体のある頂点に存在します。時々、目的関数の等高線集合が多面体の境界の一部に平行であり、最適解は解の線や面、超平面に存在し、一意的に決まらない場合があります。しかしながら、このような場合においても、最適解を作る決定変数値が連続であるのは明らかですから、最適解の値は実行可能領域の頂点のみを考えれば発見することができます。一般に、頂点は制約や変数の上下限が問題内の等号で成立するところに存在します。シンプレックス法は、通常、頂点での解や基底（基底解）を考慮し、最適解が見つかるまで、あるいはその問題が実行不能や非有界であることが分かるまで、1 つの頂点から別の頂点へ移動したり繰り返したりします。必要な反復回数は、モデルのサイズと共に増加しますが、通常、制約条件の数により大きく依存します。

Xpress-Optimizer Reference Manual Simplex Performance 20

Performance Issues 4

主、双対シンプレックス法では、考慮する頂点や反復方法が違います。LP 問題では双対シンプレックス法がデフォルトですが、`XPRSminim` (MINIM)、`XPRSmaxim` (MAXIM) をフラグ `d` と共に用いることで、明示的に呼びだせます。

逆行列計算

シンプレックス法での最適化の際、Optimizer は極めてよく逆行列計算という処理を実施します。その頻度は制御 `INVERTFREQ` と `INVERTMIN` により決まります。これは現在の解を計算し直して、より簡潔な表現を探し、その精度を調べようとします。しかし、Optimizer は現在の解の新しい表現を探せないことがあります。これは、計算精度の問題やクラッシュ、`XPRSreadbasis` (READBASIS) により作成される不安定な初期条件によるもので、この場合、いくつかのベクト

ルは基底から除かれたり、スラックや人為変数に相当するユニットベクトルにより置き換えられます。逆行列計算により、Optimizerはアルゴリズムを継続する前により安定した解を見つけるために、現在の解を調整しようとします。

部分プライシングとDevexプライシング

双対シンプレックスOptimizerは、制御PRICINGALGにより部分プライシングとPaul Harris' Devexプライシングを組み合わせて使います。一般に、部分プライシングは高速の繰返し計算を大量に行い、Devexプライシングは遅い反復計算を少し実施します。どちらがよいかは、大いに問題によります。PRICINGALGを0に設定すると、Optimizerは部分プライシングで開始し、自動的にDevexプライシングに切り替える時期を決めます。OptimizerがDevexプライシングに切り替わらないようにするには、PRICINGALGを-1に設定しますOptimizerをDevexプライシングに強制的に切り替えるにはPRICINGALG制御を1に設定してください。

出力

LP最適解を探している間、コンソールOptimizerは画面に繰返しログを表示します。ライブラリユーザの場合、同じ情報をXPRSsetlogfile, XPRSsetcblplog関数により得られます。このログはLPLOG回の反復ごとに作られます。LPLOGを正の値に設定すると、概要出力を作成します。一方、負の値に設定すると、詳細なログを作成します。0に設定すると、ログは解き終わった時のみ表示されます。

Xpress-Optimizer Reference Manual Barrier Performance 21

Performance Issues 4

4.3 バリア法の性能

ニュートンバリア法

シンプレックス法とは違い、ニュートンバリア法はLPやQP問題を解くための内点法です。名前が示しているように、これは実行可能領域内の点から点へ次々と繰り返し移動する方法です。その実行可能領域の境界へ達するとペナルティが課されるため、この方法での反復は、実行可能領域を離れられません。しかしながら、LP問題の最適解は実行可能領域の境界上に存在するため、このペナルティは、最適解に収束するための反復計算を実行できるようにアルゴリズムが移行するのに合わせて、ダイナミックに減らされなくてはなりません。

内点法は、通常、厳密に実行可能領域内の解に到達した後、真の最適な頂点解に接近しようとするだけです。それゆえ、その接近に際しては、決定変数の数値ではなく、必要な反復回数を決めます。よって、シンプレックス法とは違って、バリア法はしばしばは問題の大きさに関わらず、似たような回数の繰返しで完了します。

このバリアソルバーはXPRSsminim (MINIM)かXPRSsmaxim (MAXIM)にbフラグを付けて、問題に対して呼びだします。これは、デフォルトでQP問題に使われることになっています。二次目的関数では、多面体の実行可能領域の頂点より面上に最適解が一般的には存在します。

バリアの性能制御

ニュートンバリア法は解探索が遅いときや数字上に問題があるとき、制御の値を変えることで、その動きを変えられます。CACHESIZEが正しく設定されているかの確認は、性能に大きな影響を与えます。これについては、インテルやAMDでないプラットフォーム上ではマニュアルで設定してください。同じく、コレスキー因子分解の順序アルゴリズムをBARORDERを用いて変更することは効果があります。これを2に設定すると、オーダリング自体は非常に遅くなりますが、しばしばよい結果となります。DENSECOLLIMITのような他の制御をマニュアルで設定すると、よい効果があります。例えば、木探索で密な列が見つかったような数値上の問題については、密な列の操作を不可能にすることで、なくすことができます。このためには

DENSECOLLIMITを大きな値に設定します。この場合、最適化速度は犠牲になりますが、数値上の振舞いは、普通、よくなります。

Xpress-Optimizer Reference Manual Barrier Performance 22

Performance Issues 4

クロスオーバー

一般に、バリア法は最適解が与えられた許容範囲内であれば終了します。この解は、実行可能領域の境界上にあるため、Optimizerは、この段階でクロスオーバーを実施し、シンプレックス法を使って最適化を完了するようにし、真の解である最適解を得ようとします。最適基底解が必要な場合には、本手法は最適化が開始される前に実施される必要があります。CROSSOVER制御はこれを管理し、デフォルトではLP問題には1が設定されています。CROSSOVERが0に設定されると、クロスオーバーは実施されず、純粹にバリア法で解を探索します。

収束

最適化は、通常、主問題、双対問題の解が実行可能で、相対値双対ギャップがBARGAPSTOPより小さいと終了します。言い換えると次の通りです。例えば、BARGAPSTOP = 1.0E-8のときは、最適解の目的関数値が8桁正しいことを意味します。BARPRIMALSTOPとBARDUALSTOPパラメータは、主問題と双対問題の実行可能性の終了基準を与えます。一般に、BARGAPSTOP, BARPRIMALSTOP, BARDUALSTOP制御を変える必要はありませんが、クロスオーバーが用いられ、シンプレックス法がクロスオーバー基底から最適基底を得るまで何度も繰返しを実施するような場合には、これらの制御パラメータの値を（例えば10から100倍）小さくするのは有効かも知れません。ニュートンバリア法は、各繰返しにおいて探索方向を計算し、その方向にステップを進みます。このステップサイズがBARSTEPSTOPより小さければ、探索を終了します。収束が非常に遅い場合には、BARSTEPSTOPをもっと大きな値に設定してシンプレックス法を早い段階で呼びだし、早く計算を終わらせる方がよいかもしれません。

出力

シンプレックス法のようにコンソールOptimizerは繰返しログを表示できます。同様にライブラリユーザはXPRSsetlogfile, XPRSsetcbbarlogにより同じ情報を得られます。どちらもBAROUTPUT制御の値に依存します。

```
primalobj dualobj-
1.0 dualobj+
```

```
----- BARGAPSTOP.
```

Xpress-Optimizer Reference Manual Integer Programming — The Global Search 23

Performance Issues 4

4.4 整数計画法の性能

分枝限定法

Xpress-Optimizerは混合型整数計画問題(MIP)を解くのに分枝限定法を用います。ここでは、整数解探索の制御を理解するのを助けるために、その簡単な概略を示します。本手法には3つの主要な概念があり、それは、分割、緩和、見切りです。

緩和とは整数計画問題の整数制約を取り除くことです。緩和問題は線形計画問題ですから、解くことができ、以下のどれかの結果となります。

- a) LPは実行不能であり、よってMIPもまた実行不能である
- b) LPは実行可能だが、いくつかの整数制約が満足されない、つまりMIPはまだ解けていない
- c) LPは実行可能で、全ての整数制約も満足されており、MIPは解けている

d) LP は非有界である

最後の d) はややこしい場合です。これは最初の緩和問題でしか起きず、問題がよく検討されていない場合にあり得ます。よって、この LP は非有界でないと考えられます。

a) と c) の場合は、更なる検討作業は不要で、その MIP 問題を「見切った」と言います。b) は満たされていない整数制約の 1 つを選んで分割を実施しなくてはならず、まだ作業途中です。

例えば、LP 最適解の整数変数 x の値が 1.34 とすると、これは整数制約に違反しています。原問題では $x < 1.0$ か $x > 2.0$ のどちらかが成立します。この 2 つの IP 問題が解ければ、 x の整数値を取り逃がすことはありません。そこで、問題を 2 つの部分問題に分割します。これらの部分問題の両方が解けて、2 つのうちの良い解を選ぶと、MIP が解けたこととなります。各部分問題を解くには、先と同様に緩和手法を使い、これを繰り返します。

これはある程度の任意性のある木探索アルゴリズムと言われるものです。木の各ノードは MIP 部分問題です。MIP は緩和され、LP 緩和問題が解かれます。その緩和が見切られなければ、その MIP は 2 つの部分問題に分割され、各子問題はノード MIP と同じ制約に 1 つの制約を追加されます。よって、各ノードは見切られるか、2 つの子 (部分問題) を持つかのどちらかです。

IP problem

$x2 . x1 .$

Sub-problem 2 Sub-problem 1

Xpress-Optimizer Reference Manual Integer Programming — The Global Search 24

Performance Issues 4

木を探索する途中、整数解が見つかったと、問題の解の限界が得られます。MIP の LP 緩和問題の最適目的関数値はその MIP 問題のそれより悪いことはありません。見つかった MIP ノードの最もよい値は、未解決ノードのカットオフ (足切り値) として使えます。LP 緩和問題の値が足切り値よりよくなければ、そのノード以下のいかなる MIP 子問題は、既に得られた MIP 解の値よりよくなることはありません。この場合もノードを見切ることができ、これ以上の考慮は必要ないのです。

足切り値の考え方は整数解がないと分かった場合のみならず、ある値よりよい最適解がある、あるいはあることが考えられる場合にも応用できます。緩和問題の値が足切り値より悪ければ、そのノードは捨てられます。しかしながら、過度に楽天的な足切り値を選ぶと、最適解を含む全ての整数実行可能解を見逃す危険があります。

整数最適解にある程度の許容範囲 (許容範囲) を考えられるのであれば、足切り値の考え方は更に強力になります。整数解が見つかったとき、例えば、これより 100 以上よい解がないのであれば、これを最適解として許容するとした場合です。このとき、その見つかった解より 100 だけよい値を足切り値に設定できます。

ノードと変数の選択

分枝限定の技術については、ユーザは色々を選択できます。但し、実際にそれをうまく行うことは 2 つの選択に大いに依存します。

a) いかなる段階においても、一般に見切られていない未解決のノードがいくつかあります。

このうち、どれを最初に解くかというノード選択問題

b) 攻略ノードを選んだ際、どの変数を分割するかという変数分割問題

Optimizerには、両選択について多くの問題でうまく行った戦略をデフォルトとして組み込まれ

ています。特定の問題に合った探索戦略ができるよう、いくつかの制御があります。Optimizer はノードの選択時ではなく、LP緩和問題を解いたときに変数を選択することから、変数選択問題について先に議論することになります。

Xpress-Optimizer Reference Manual Integer Programming — The Global Search 25

Performance Issues 4

分枝のための変数選択

各離散要素には、デフォルトの 500 かディレクティブファイルにユーザが設定した分枝の優先順位がつけられています。順位値が小さいと分枝に選ばれやすくなります。各離散要素が LP 解の値から離される際の単位劣化量として推定される押し上げ、押し下げに基づく擬コストを指定できます。

Optimizer は制約を満足していなくて、最も優先度の高いものの中から分枝要素を選びます。これらについては、制約を満たす(劣化)コストが最も高いと推定されるものを選びます。離散要素の推定コストの和により、そのノードから得られる整数最良解のかなり粗い推定ができます。この推定値が、ある問題クラスにおいていつも偏っているようであれば、擬コストをデフォルトの0.1から変えてみることは有効です。この操作はXPRSreaddirs (READDIRS)コマンドにより実施できます。

ノード選択

各アクティブノードには、LP 緩和解の値と整数解への推定劣化量が付いています。制御 NODESELECTION, BACKTRACK, VARSELECTION, BREADTHFIRST が次のノードの選択方法を決めます。

NODESELECTION の値は、ノード選択候補集合、つまり、選ばれるノードの集合を定義し、BACKTRACK の値が、その候補集合からのノードの選択に使われる基準を定義します。NODESELECTION が 1 (通常のデフォルト値) なら、2つの子ノードが候補集合を形作りますが、両方が見切られた場合には、全てのアクティブノードが候補集合を形作ります。NODESELECTION が 2 のときには、常に全てのノードが候補集合となり、最良、もしくは幅優先探索を実行します。NODESELECTION が 3 のときには、深さ優先探索を実行します。NODESELECTION が 4 のときには、優先順位の高い最初の BREADTHFIRST ノードは全てが候補となり、その後は、通常のデフォルト戦略に従います。

候補集合におけるノード決定には、BACKTRACK の値が選択基準を決めます。BACKTRACK が 1 で MIPTARGET が設定されていない場合 (直接ユーザにより、あるいは整数解を前もって探索して)、最もよい推定値のノードが選ばれます。BACKTRACK が 1 で MIPTARGET が設定されていると、Forrest-Hirst-Tomlin 基準が使われます。最小化問題の場合、次の値が最大のノードを選びます。

Xpress-Optimizer Reference Manual Integer Programming — The Global Search 26

Performance Issues 4

(MIPTARGET - 目的関数値 - 推定劣化量) / 推定劣化量

VARSELECTION の値は推定劣化量に影響します。VARSELECTION が 1 (デフォルト) のとき、推定劣化量は制約を満足していない要素ごとに分枝する 2つの分枝のよい方から推定されます。推定劣化量の他の計算方法は VARSELECTION を設定することで実施できます。

下の表は up_j と $down_j$ が離散要素 j において分枝する際の増減劣化量としてあり得る値を示しています。BACKTRACK が 2 のとき、推定値が最小のノードがいつも選ばれます。BACKTRACK が 3 のとき、最小下界値のノードがいつも選ばれます。

カットオフ値の調整

パラメータ MIPRELCUTOFF と MIPADDCUTOFF はデフォルト値が零でないため、MIPADDCUTOFF の設定が Optimizer のコマンドの集合内の別のところで起きる影響をよく注意して考慮しなくてはなりません。MIPADDCUTOFF を XPRSminim (MINIM)や XPRSmaxim (MAXIM)に先だって設定すると、その値は最適化処理において変えられるかもしれません。LP 最適化の最後に、MIPADDCUTOFF は次のように設定されます。

$$\max (\text{MIPADDCUTOFF}, 0.01 * \text{MIPRELCUTOFF} * \text{LP_value})$$

ここで LP_value とは LP Optimizer によって発見された LP 最適解の最適値です。この式が必要なく、MIPADDCUTOFF を指定した値にしたい場合には、MIPADDCUTOFF は LP Optimizer の動作後に設定されなくてはなりません。MIPRELCUTOFF に値を指定するのであれば、LP Optimizer の実行前に指定されなくてはなりません。

VARSELECTION Estimated Degradation (deg)

1

2

3

4

5

6

$$\min_j \text{upj downj}, ()$$

$$\overline{\text{upj downj}} + ()$$

$$2.0 \min_j \text{upj downj}, () \cdot \max_j \text{upj downj}, () + \{ \}$$

$$\overline{\max_j \text{upj downj}, ()}$$

$$\overline{\text{downj}}$$

$$\overline{\text{upj}}$$

Xpress-Optimizer Reference Manual Integer Programming — The Global Search 27

Performance Issues 4

整数事前処理

MIP 問題を解く前に MIPPRESOLVE が零でない値に設定されていると、整数事前処理が分枝限定木探索 (トップノードを含む) の各ノードで実行されます。これは、縮小コストの固定、バイナリ変数の固定やトップノードでの分析を含みます。変数は、あるノードにおいて固定されると、その全ての子ノードにおいても固定されたままですが、行列からは消去されません (事前分析による変数固定とは違います)。この整数事前処理は XPRSminim (MINIM)や XPRSmaxim (MAXIM)の線形(1)フラグには影響されません。

MIPPRESOLVE は値が次のようなビットマップです。

MIPPRESOLVE の値が 1+2=3 のときには、縮小費用と変数が固定されます。

ビット	値	アクション
0	1	縮小費用固定
1	2	変数固定
2	4	ルートノードにおける分析

Xpress-Optimizer Reference Manual Integer Programming — The Global Search 28

Performance Issues 4

Xpress-Optimizer Reference Manual Viewing and Modifying the Matrix 29

Implementing

Algorithms 5

5 アルゴリズムの実行

5.1 行列を見る , 修正する

以上に示したような簡単な操作は Xpress-Optimizer の多くのユーザに満足していただけるものと思いますが、モデルをロードした後、変更した問題を再度最適化するのに先だって問題の行列を見たり変更する必要があることが、時々あります。このことは、特に原問題が実行不能なことが分かり、実行可能領域を作成するために制約条件（行列）を除去する必要がある場合に重要です。このため、ライブラリユーザのために、Xpress-MP にはいくつかの関数がありますので、そのいくつかについて以下に示します。

行列を見る

Optimizer には、最適化の前後に目的関数、制約式右辺、上下界値、行列要素にアクセスすることのできる関数があります。前であれば、解のないことがはっきりしていても、その問題に関する全ての情報が得られます。後であれば、行列が事前分析されているかいないかに依存して構造が得られます。これについては以下の章で詳しく述べますので、ここでは詳しく述べませんが、完全な行列情報は、その行列が事前分析されていない場合にのみ得られることは重要です。事前分析されると、部分的な情報しか得られません。行列がどちらの状態が分からなければ、問題特性 `PRESOLVESTATE` で、その状態を確認できます。

ここでは、事前分析されていない状態の行列を対象とします。このような行列では、行は問題の制約を表し、`XPRSgetrows` により得られます。そのタイプやレンジについては関数 `XPRSgetrowtype` , `XPRSgetrowrange` で得られ、また各制約条件の名前は `XPRSgetnames` コマンドで返されます。右辺の値やそのレンジは `XPRSgetrhs` , `XPRSgetrhsrange` ルーチンにより得られます。行列の行に関して、目的関数の係数は `XPRSgetobj` ルーチンで同じように得られ、また関数 `XPRSgetqobj` は 2 次の目的関数の係数を返します。

Xpress-Optimizer Reference Manual Viewing and Modifying the Matrix 30

Viewing and

Modifying the Matrix 5

行列の列は問題の決定変数を表し、これについての情報も興味のあるところですが、列には一般に名前があり、`XPRSgetnames` 関数を再び使うことでアクセスできます。列の上下界値はコマンド `XPRSgetub` , `XPRSgetlb` によりアクセスでき、そのタイプやレンジは関数 `XPRSgetcoltype` , `XPRSgetcolrange` で行列列と同じように得られます。`XPRSgetcols` 関数を用いると、行列の列、そのものが得られます。

6 章「コンソールとライブラリ関数」にこれら関数の詳細な形式や各々の使い方を詳しく示しますので、ユーザが自分のプログラムで使用する前に詳細や例のページを参照ください。

行列の修正

時として、前に解かれたモデルについて、修正した行列が最適化のために再度表される前に、変更しなくてはならない場合があります、このためのルーチンがあります。行と列を(XPRSaddrows , XPRSaddcols を使って) 追加したり、(XPRSdelrows , XPRSdelcols を使って) モデルから取り除くことが可能です。行や列が追加されると、最大効果をあげるために、行列が EXTRAROWS, EXTRACOLS, EXTRAELMS, EXTRAMIPENTS 制御の設定により読まれる前に、それらのためのスペースを確保されなくてはなりません。そうしないと、自動的にリサイズされますが、実際に必要な量以上のスペースが割り当てられ、計算速度が遅くなるかもしれません。同様にして、行や列のタイプも (XPRSchgrowtype, XPRSchgcoltype を使って) 変更でき、行列の係数は (XPRSchgcoef, 複数の変更は XPRSchgmcoef を使って) 変更できます。右辺やレンジについては XPRSchgrhs , XPRSchgrhsrange によって、目的関数の係数は XPRSchgobj によって変えられます。2 次の目的関数についても必要に応じて XPRSchgqobj , XPRSchgmqobj で同様に変えられます。

上述のように、行列は事前分析後で事後分析前の場合、修正できませんが、変数の上下界値は (XPRSchgbounds を使って) 変えられます。次の節では、事前分析機能について、事前分析した行列を用いた作業の困難を克服するためのいくつかの提案と共に議論します。上述の全関数の例とそれらの具体的な構文については、6章「コンソールとライブラリ関数」のリファレンスのページを参照ください。Optimizerライブラリプログラムでの使い方を詳しく示しています。

Xpress-Optimizer Reference Manual Working with Presolve 31

Implementing

Algorithms 5

5.2 事前分析のはたらき

Optimizerには最適化処理に先だって問題を簡素化するための、いくつかのアルゴリズムがあります。この精巧な手法の集まりが事前分析で、問題の行列を修正して解きやすくし、Optimizerの性能を、しばしば大きく向上します。事前分析アルゴリズムは冗長な行や列を分析して取り除き、行列サイズを小さくし、多くのユーザにとって計算時間を減らすのに有効な頼りになるツールです。しかしながら、事前分析はオプションであり、PRESOLVE制御を0に設定すると動作しません。通常、これは1に設定されており、デフォルトで事前分析は動作します。事前分析の結果、問題が原モデルと全く異なるような形に見えることがあり、ユーザによっては混乱することがあるかも知れません。通常のOptimizerの使用では、このことは問題ではありません、但し、稀にエラーが起きたり、ある問題タイプの行列の追加部分にアクセスしようとしたとき、事前分析された値が返されることがあります。ここでは、このような混乱を避ける方法について示します。

線形計画問題

線形問題のために、事前分析はデフォルトで XPRSmaxim (MAXIM), XPRSminim (MINIM)ルーチンにより呼ばれ、最適化アルゴリズムが呼びだされる前に行列を整理します。最適化に続いて、原問題の解と保存されている原行列を戻すため、行列全体が自動的に事後分析されます。その結果、最適化前か解いてすぐに、行列全体を原形のままで見たり、上で説明したように変更することができます。

事前分析の間に新しい行列の係数と上下界値の要素は、行列内の追加フリースペースの必要とされるところに作成されます。制御EXTRAPRESOLVEを用いることで、準備されるこの要素数を手動で設定できます。設定されなければ、XPRSreadprob (READPROB)が呼ばれるときに、推定値が設定されます。要素数が十分でないと、警告が表示されて、最適化が続行されます。何らかの事由で最適化が終了する前に中断される場合は、CTRL-Cキーが用いられたか、

LPITERLIMITやMAXTIMEの設定が不十分なためであり、その場合、問題は事前分析された形のままとなります。この場合、LPITERLIMITやMAXTIMEの値を増やし、問題を再度ロードして最適化をもう1度実施したいと思うことでしょうか。しかしながら、問題のある部分が不適切な計算時間をもたらすような場合、事前分析済みの行列を修正するのではなく、最適化の前に行列をリロードして必要な修正を実施することを強くお勧めします。

Xpress-Optimizer Reference Manual Working with Presolve 32

Working with Presolve 5

(混合) 整数計画問題

モデルに離散要素が含まれると、上下界値強化や係数強化のような整数事前分析手法もまた、LP緩和の強化のために使われます。簡単な例として、行列にバイナリ変数 x があり、行列の制約の1つが $x \leq 0.2$ である場合です。ここで x は決して1にはなれないため、 x は0に固定されます。事前分析がこのように行列を変えるために離散要素を使う場合、LP緩和が強化されたといえます。コンソールユーザの場合、このことは画面に表示され、ライブラリユーザの場合、設定によってコールバック関数がログファイルに送られます。このとき、事前分析された行列のLP緩和問題の目的関数の最適値は、事前分析されていない行列のものと異なることがあります。

恐らく、整数問題に対しての大きな違いは、XPRSsmaxim (MAXIM), XPRSsminim (MINIM)による最適化の後、XPRSsglobal (GLOBAL) が続いて実施されるため、デフォルトでは事後分析が実行されないことです。但し、整数探索の前にLP緩和問題の解を見ることができます。事前分析が終わっていても、解は事前分析がLP緩和を強めたため、純粋な緩和問題の最適値でない数値かもしれません。先の場合、 x は0に固定されているので変数 x は最適値でないかもしれません。事前分析が使われると、LP緩和は異なるかもしれませんが、整数最適解は同じで、探索時間はかなり短くなるかもしれません。

離散要素のあるモデルのLP厳密解は、XPRSsminim (MINIM), XPRSsmaxim (MAXIM) コマンドにフラグ 1 を付ければ得られます。これは変数の離散制約を取り除き、LP緩和を厳しくせずに、行列を解きます。上記の場合、 x は0に固定されず、0から0.2までの範囲にあるものとします。LP緩和に興味がないければ、LP緩和問題を解いてすぐに整数探索を行うのがいくらか効果的で、これはXPRSsminim (MINIM), XPRSsmaxim (MAXIM) コマンドに g フラグを付けることで実施できます。

XPRSsglobal (GLOBAL)が整数解を見つけると、事後分析が行われ、解ファイルに書き込まれ、原問題の解が使えるようになります。SOLUTIONFILE 制御が0に設定されていると、XPRSgetsol はメモリから事前分析済みの解を読むか、解ファイルから事後分析された解を読みこみます。同じようにして、事後分析された解は解ファイルでなくメモリからも読みこまれます。

Xpress-Optimizer Reference Manual Using the Callbacks 33

Implementing

Algorithms 5

しかし、XPRSsglobal (GLOBAL) を呼んだ後には、行列全体の事後分析は実行されず、原行列を復帰できません。行列へ更にアクセスする必要がある場合は、解ルーチンの呼びだしの前に再度ロードするか(XPRSscopyprobを使って)コピーする必要があります。あるいは、例えばXPRSwriteprob (WRITEPROB) を呼んで事前分析された行列を書き出します。

混乱の原因

上記と6章「コンソールとライブラリ関数」にある行列を修正するライブラリルーチンのほとんどは、事前

分析された行列には作用しません。但し、カットプールマネージャによりカットを追加したり (XPRSchgboundsを使って) 変数上下界値を変えることだけは例外的に可能です。これらの関数は事前分析された問題を参照します。もし、行、列、上下界値やこれらの数を調べたい場合、これらの情報は原行列ではなく事前分析された行列から得られます。これらを使う際に注意していても、事前分析やスケールリングされた行列と動作するように、特に設計された関数が少し存在します。例として次のようなコマンドに含まれます(XPRsgetpresolvebasis, XPRsgetscaledinfeas, XPRsloadpresolvebasis, XPRsloadpresolvedirs)

5.3 コールバックを使う

Optimizerの出力

コンソールユーザは Optimizer が探索する現在の問題の解についての情報が、常に標準出力デバイスに提供されます。ライブラリユーザも同じ出力を XPRssetlogfile によってログファイルが設定されていれば、得ることができます。しかし、コンソールユーザはこの作成された情報に対して結果を返すことができ、セッションと関係できますが、ライブラリユーザについてはそうではありません。というのは、セッションが始まる前にプログラムは書かれてコンパイルされるからです。ライブラリユーザには上記の出力形式をよりインタラクティブに行うもう1つの方法としてコールバック機能が提供されます。

ライブラリコールバックはOptimizerに指示をできるユーザ定義のルーチンの関数の集合です。このようにして、ユーザは最適化処理における種々の場面で呼ばれるルーチンを定義でき、求解アルゴリズムを継続する前にユーザプログラムに戻るようにOptimizerに促せます。恐らく3つの最も一般的なコールバック機能は、LP解の探索に関連づけられています。しかし、このルーチンが呼ばれる状況の大半は整数探索に関連しており、以下の通りです。

Xpress-Optimizer Reference Manual Working with the Cut Manager 34

Working with the Cut

Manager 5

LP探索コールバック

標準出力を Optimizer から受け取ったり、ログファイルに保存する代わりに、コールバック XPRssetcbmessage により、Optimizer がテキスト列を出力する度に呼ばれるルーチンを定義できます。これは各メッセージ出力の状態を返すため、ユーザルーチンはエラーや警告メッセージをテストでき、それによって適切な行動をとることができます。

また、関数XPRssetcblogとXPRssetcbbarlogの組により、シンプレックス法やバリア法の各反復後に、ユーザは応答できます。制御LPLOG, BAROUTPUTを制御することで、このルーチンが呼ばれる頻度を減らすように設定できます。

整数探索コールバック

離散要素を含む問題が最適化されると、一般に多くのLP問題(ノード)が整数木探索の一部として解かれます。この処理における個々の点では、Optimizer にルーチンを指定するのに使われるコールバックによってユーザ定義ルーチンを呼ぶことができます。このルーチンについては、XPRssetcbprenode を使って新しいノードが選ばれたときはいつでも呼ぶことができ、ノードの選択を変更するのに使われます。ルーチンは、特定のノードで最適解か整数解が見つかったとき、あるいはXPRssetcbinfnode を用いて実行不能なノードが見つかったときに XPRssetcboptnode か XPRssetcbintsol コールバックを用いて呼ばれるように指定できます。よりよい整数解が見つかって、ノードが切り離されると、XPRssetcbnodecutoff により指定されていればルーチンが呼ばれます。XPRssetcbchgbranch か XPRssetcbchgnode を用いることで、新しい分枝変数が設定されたときや新しいコードを選択するためにコードを遡るときには、ルーチンが呼ばれるように指定できます。恐らく、もっと技術的には、XPRssetcbsepnode と XPRssetcbestimate は、ノードの分割方法の決定

方法を決めたり、各ノードでの推定劣化量をユーザ離散要素での分枝から得るルーチンを指定できます。

最終整数コールバック `XPRSsetcbgloballog` は、LP探索コールバックに似ており、グローバルログ文を出力されると、いつでもユーザルーチンが呼ばれるようになります。この頻度は制御 `MIPLOG` により設定できます。

5.4 カットマネージャのはたらき

カットとカットプール

(混合型) 整数計画問題の解の整数探索は、たくさんの LP 問題 (ノード) の最適化を呼びだします。この処理は、どのような整数最適解を含んだ状態で実行可能領域を狭くする追加行 (制約) を行列に供給することができるかによって、効果的になることがよくあります。このような追加行はカット面やカットと呼ばれます。

Xpress-Optimizer Reference Manual Working with the Cut Manager 35

Implementing

Algorithms 5

デフォルトでは、整数探索中に計算処理を高速化するため、Optimizer により自動的にカットが行列に追加されます。しかし、上級ユーザ向けに、Optimizer ライブラリは大きな自由度を提供し、特定のノードで追加されるカットを選択したり完全なカットを除去できます。カット面そのものはカットプールに展開され、ライブラリ関数により巧みに操作されます。

カットは特定のノードで行列に直接追加されるか、引き続き行列にロードされる前に最初に、カットプールに保存されます。一般に、カットを容易に作成できれば、各部分問題 (ノード) が最適化された後に行列にカットを直接追加し、余分なカットを取り除くことが好ましいのですが、これらの2つのどちらのアプローチが採用されたかによって、時にちょっとした違いを生じます。ノードで行列に追加され、またそのノードで消去されないカットは、自動的にカットプールに追加されます。もし、生成された全カットを保存したいなら、最初にカットをカットプールに追加してください。そうすると、カットは、カットプールから行列にロードできます。このアプローチには、カットプールルーチンが重複カットを見つけたり、強力なカットだけを保存するのに使うことのできるという利点があります。別のノードで行列に追加されたカットに関する経緯を保存するために、カットをユーザ定義のカットタイプに従って分類できます。カットタイプはノード番号のような数かビットマップで表せます。後者の場合、カットタイプの各ビットがカットの特性を示すのに使われます。例えば、現在のノードとその子ノードで適用可能なローカルカットとして、または全ノードで適用可能なグローバルカットとしてカットを分類できます。カットタイプの最初のビットが設定されるとローカルカットが指定され、2番目のビットが設定されるとグローバルカットが指定されます。カットタイプの他のビットは、カットの他の特性を示すのに使われます。ビットマップを使う利点は、例えば全てのローカルカットのような全カットの特性を簡単に選べることにあります。

カットマネジメントルーチン

`XPRSaddcuts` を用いて、現在のノードでカットは行列に直接追加されます。ノードで行列に追加されたカットは、自動的にカットプールに追加され、そのノードで `XPRSdelcuts` により特に消去されない限り、子ノードで復帰できます。カットは `XPRSaddcuts` を用いて現在のノードに追加されたり

`XPRSloadcuts` を使ってカットプールからロードされるのと同様、自動的に復元された親ノードから消去されます。

Xpress-Optimizer Reference Manual Working with the Cut Manager 36

Working with the Cut

Manager 5

普通は、基底スラックのカットだけを消去することが一番ですが、そうしないと、基底が有効でなくなり、

最適基底を回復するのに多くの反復が必要となります。XPRSdelcutsの第2引数が1に設定されると、他の制御に関係なく、非基底スラックのカットは決して消去されません。よって、これをいつも1にしておくことを強くお勧めします。

関数 XPRSstorecuts により、カットは直接カットプールに保存できます。カットプールに追加されたカットは、現在のノードで行列に自動的に追加されないため、このようなカットは、アクティブになる前に XPRSloadcuts を使って行列に明示的にロードされなくてはなりません。XPRSstorecuts の第3引数が1に設定されると、カットプールは、追加されたカットと同一なカットタイプの重複カットをチェックします。重複カットが見つかったら、右辺値がカットを強くする場合に新しいカットが追加されます。カットプール内のカットが、追加されたカットより弱ければ、木のアクティブノードに、既に適用されていない場合に限り取り除かれます。一方、この係数が2に設定されていれば、カットタイプは無視され、同じテストが全てのカットに実施されます。分枝限定木のアクティブノードに既に適用されていない限り、XPRSdelcpcuts ルーチンにより、カットをカットプールから取り除くことが、可能です。

カットプールのカットのリストは XPRSgetcpcutlist が、その添字リストを返す限り、コマンド XPRSgetcpcuts によって得られます。現在のノードでアクティブなカットのリストは XPRSgetcutlist によって得られます。

ユーザカットマネージャルーチン

ユーザはまた、分枝限定探索において個々のノードで呼ばれるユーザ自身のカットマネージャルーチンを作成できます。このルーチンは、前もってライブラリ関数の呼びだして定義され、コールバック同様、呼ばれる頻度に従って定義されます。この処理の最初に、カットマネージャ初期化ルーチンが呼ばれ、XPRSsetcbinitcutmgr により指定されます。同じように、最後には、XPRSsetcbfreecutmgr により終了ルーチンが指定できます。コマンド XPRSsetcbcutmgr により、木の各ノードで呼ばれるルーチンを定義できます。これらの関数の更に詳しいことについては、6章の関数リファレンスを参照ください。

Xpress-Optimizer Reference Manual Goal Programming 37

Implementing

Algorithms 5

5.5 目標計画法

概要

目標計画法はLPの拡張で、目標を制約の集合として与えます。目標計画法では2つの基本的なモデルとして優先(辞書式)モデルとアルキメデスモデルがあります。優先モデルでは、目標に優先順位がつけられます。あるレベルの目標は次のレベルの目標よりも絶対的に重要です。アルキメデスモデルでは、目標を達成できない場合の重みやペナルティが設定され、これらの最小化問題となります。

Optimizerでは、目標は制約か目的関数(N行)により構築されます。制約が目標の構築に使われる場合、目標は制約違反の最小化となります。制約が満たされれば目標達成となります。優先モデルでは、優先順位にしたがって、できるだけたくさんの目標を達成しようとします。アルキメデスモデルでは、各目標を満たせない際に発生するペナルティの重み付き総和を最小化します。目標がN行で構築されている場合、優先モデルでは各N行に対して、そのN行の最適値から目標が計算されます。これは、N行の最適値との許容格差を相対値か絶対値で指定して実施されます。アルキメデスモデルでは、問題は複数の目的関数を有するLPとなり、目的関数の重み付き総和を最小化します。ここでは、4つの異なる形の目標計画の例を示します。目標計画法はXPRSgoal (GOAL)コマンドを用いることで実施されます。構文は本マニュアルのリファレンスのページに詳しく説明してあります。

優先モデル

優先モデルでは、目標は最も重要なものから軽いものまでランク付けられます。最初に、最も重要な目標を満たすようにします。そして、最初の目標を満たしながら2つ目の目標を満たすようにします。この

ようなことを高い順位の目標の離れを増加しなくてはならなくなるまで続けて目標を満たすようにします。

例は以下の通りです

```
goal 1 (G1): 7*x + 3*y >= 40
goal 2 (G2): 10*x + 5*y = 60
goal 3 (G3): 5*x + 4*y <= 35
Xpress-Optimizer Reference Manual Goal Programming 38
```

Goal Programming 5

```
LIMIT : 100*x + 60*y <= 600
```

最初に、第1目標(G1)を満たそうとします。そして、 $x=5.0$, $y=1.6$ が得られますが、第2(G2)第3(G3)目標を満たしません。第1目標を満たしながら、第2目標を満たそうとすると、 $x=6.0$, $y=0.0$ は満たしますが、第3目標を満たせません。そこで、この過程を繰り返します。この場合、3つ全ての目標を満たす解は存在しません。

アルキメデスモデル

アルキメデスモデルでは、制約違反の重み付き総和を最小化します。次のような問題を考えます。ここでは、ペナルティが与えられています。

問題の答はこのようになります。

$7*x + 3*y$ が40より小さい場合には1単位につきペナルティ8単位を与えます。他の式も同様です。そして、解はこのペナルティの総和を最小化します。ペナルティは重みと言い換えてもよいでしょう。ここでの解は $x=6$, $y=0$, $d1=d2=d3=0$, $d4=5$ となり、これは1番目と2番目に重要な制約を満たしていますが、3番目の制約を満たすには右辺を5単位だけ減らさなくてはなりません。

問題が実行不能の場合には、全ての目標制約が緩和され、解は得られません。

Penalties

```
goal 1 (G1): 7*x + 3*y >= 40 8
goal 2 (G2): 10*x + 5*y = 60 3
goal 3 (G3): 5*x + 4*y >= 35 1
LIMIT : 100*x + 60*y <= 600
Minimize 8*d1 + 3*d2 + 3*d3 + 1*d4
Subject to: 7*x + 3*y + d1 >= 40
10*x + 5*y + d2 - d3 = 60
5*x + 4*y + d4 >= 35
100*x + 60*y <= 600
d1 >= 0, d2 >= 0, d3 >= 0, d4 >= 0
Xpress-Optimizer Reference Manual Goal Programming 39
```

Implementing

Algorithms 5

目的関数を使った優先モデル

重要度が分かっている目的関数の集合があるとします。制約条件のある優先モデルの場合、目標は最も重要なものから軽いものまでランク付けられています。まず、第1の目標の最適値を探します。最適値が見つかったら、この目的関数がある程度よりは悪くならない範囲で変えてもよいという制約式に変更します。これは最適解を得る前に固定値(絶対偏差)あるいは相対値(相対偏差)で与えます。そして、次の目標(2番目に重要な目的関数)を最適化します・・・以下同様です。

例えば、次のような問題を考えます。

各N行には最適化(最大, 最小)の方向と相対(P)または絶対(D)偏差が設定されています。OBJ1とOBJ3には各々10%, 20%の相対偏差が、またOBJ2には絶対偏差4単位が与えられています。

第1の目的関数を最大化すると最適値は4.615385となります。10%の偏差をつけて目的関数を制約

に変更します。

$$5*x + 2*y - 20 \geq -4.615385 - 0.1*4.615385$$

さて、先の目的関数の値は最良値から10%の範囲を出ないこととして、次に重要な目的関数(OBJ2)を最小化すると、最適値は51.133603となります。その結果、第2の目的(OBJ2)は次のような制約に書き換えられます。

$$-3*x + 15*y - 48 \leq 51.133603 + 4$$

この目的関数の値が最小値から4単位以上大きくなりませんとします。

Sense D/P Deviation

goal 1 (OBJ1): $5*x + 2*y - 20$ max P 10

goal 2 (OBJ2): $-3*x + 15*y - 48$ min D 4

goal 3 (OBJ3): $1.5*x + 21*y - 3.8$ max P 20

LIMIT : $42*x + 13*y \leq 100$

Xpress-Optimizer Reference Manual Goal Programming 40

Goal Programming 5

最後に、OBJ3を最大化します。最適値141.943668が得られます。20%の偏差が許されているので、この目的関数は次のような制約式に書き換えられます。

$$1.5*x + 21*y - 3.8 \geq 141.943995 - 0.2*141.943995$$

この問題の解は $x = 0.238062$, $y = 6.923186$ です。

目的関数を使ったアルキメデスモデル

最後に、重み付けられた目的関数値の総和を最適化します。言い換えると、複数の目的関数がある問題を解きます。次のような問題を考えます。

ここでは3つの異なる目的関数があり、それらを重み欄の値で重み付けして1つの目的関数にまとめます。このモデルの解は次の式を最小化したものとなります。

$$1*(-3*x + 15*y - 48) - 100*(5*x + 2*y - 20) - 0.01*(1.5*x + 21*y - 3.8)$$

解における目的関数の各値は次の通りです。

$$\text{OBJ1: } 5*x + 2*y - 20 = -4.615389$$

$$\text{OBJ2: } -3*x + 15*y - 48 = 67.384613$$

$$\text{OBJ3: } 1.5*x + 21*y - 3.8 = 157.738464$$

解は $x = 0.0$, $y = 7.692308$ です。

Weights Sense

goal 1 (OBJ1): $5*x + 2*y - 20$ 100 max

goal 2 (OBJ2): $-3*x + 15*y - 48$ 1 min

goal 3 (OBJ3): $1.5*x + 21*y - 3.8$ 0.01 max

LIMIT : $42*x + 13*y \leq 100$

Xpress-Optimizer Reference Manual Console and Library Functions 41

Console and Library

Functions 6

6 コンソールとライブラリ関数

Xpress-Optimizerのコンソールユーザ、ライブラリユーザとも行列ファイルからの問題の入力や解のための単純なルーチンから、複雑なコールバック関数や求解過程の大規模制御まで、多くのルーチンを使うことができます。これらのうち「コンソールモード」を含む中核となる機能については、両ユーザとも利用可能です。ライブラリユーザは更に高度な機能が使えます。この機能は、コンソールモードにより提供される機能を拡張し、またOptimizerとユーザプログラムとの間を更に制御したり、より複雑な問題の発展を提供します。

6.1 コンソールモード関数

本章には、コンソール及びアドバンスモードの関数を並べて説明してあるため、ライブラリユーザはOptimizerライブラリの全能力を使うためのクイックリファレンスとして使用できます。コンソール Xpressユーザについては、以下の関数しよう可能です。

コマンド	説明	ページ数
ALTER	行列要素や制約条件を変更する	53
FIXGLOBAL	離散要素を解ファイルの値に固定する	83
GLOBAL	整数解の探索を始める	141
GOAL	目標計画を実行する	144
HELP	コマンドの簡単な説明を表示する	146
IIS	既約実行不可能集合の探索を始める	147
MAXIM / MINIM	解探索ルーチン	175
QUIT	コンソールOptimizerを終了する	179
RANGE	問題のレンジ情報を計算する	180
READBASIS	前もって保存された基底をファイルから読みこむ	182
READDIRS	整数解探索の方向を示すディレクティブファイルを読みこむ	184
READPROB	ファイルから問題行列を読みこむ	186
RECURSE	非線形問題を解くための反復を実行する	189
RESTORE	ファイルからOptimizerのデータ構造を戻す	191
SAVE	現在のデータ構造をファイルに保存する	193
SCALE	現在の行列を再スケールする	194
SETPROBNAME	現在のデフォルト問題名を設定する	232

Xpress-Optimizer Reference Manual Console and Library Functions 42
Console and Library

Functions 6

関数についてのリストは、2.7章「クイックリファレンス」を参照してください。

6.2 関数の記述要領

本章に示した関数は、以下のように整理して示します。

関数名

各ルーチンの記述は分かりやすくするために新しいページから始まっています。関数のライブラリ名はページの左上に、コンソールXpress名は右上にあります。

目的

ルーチンの簡単な説明とその目的がインフォメーションセクションを始めます。

概要

ルーチンを使うための構文の概要を示します。オプションである引数及びフラグは必要なければNULLと指定されます。このような可能性のある場合については、引数のところかルーチンの説明の最後に補足として示してあります。関数がコンソールモードで使えるもの場合、ライブラリの構文を最初に示し、続いてコンソールXpressの構文を示します。

引数

ルーチンの引数の一覧とそれに設定可能な値が示されます。

STOP	コンソールOptimizerを終了させる	235
------	----------------------	-----

WRITEBASIS	ファイルに現在の基底を書きこむ	240
WRITEOMNI	バイナリOMNI形式ファイルに現在の解を書きこむ	242
WRITEPROB	行列ファイルに現在の問題を書きこむ	244
WRITEPRTRANGE	ファイルにレンジ解を出力する	246
WRITEPRTSOL	ファイルに現在の解を出力する	248
WRITERANGE	ファイルにレンジ情報を書きこむ	250
WRITESOL	ファイルに現在の解を書きこむ	252
コマンド	説明	ページ数

Xpress-Optimizer Reference Manual Console and Library Functions 43

Console and Library

Functions 6

エラー値

Optimizer リターンコードは9章「エラーメッセージ&リターンコード」に説明します。なお、ライブラリユーザの場合、リターンコード32は、追加のエラー情報が得られるかもしれないことを示し、エラーを起こした関数を特定します。これは

```
XPRSgetintattrib(prob, XPRS_ERRORCODE, &errorcode);
```

を呼び出すことで得られます。

各関数についてこうして得られるエラー値はエラー値セクションに示されます。

XPRSgetlasterror関数を使うことで、エラーの説明が得られます。エラー値に関心がなければ、本セクションを省略してください。

関係する制御

そのルーチンに関係する制御については、タイプ別に示されます。ライブラリユーザは、上記のエラー値セクションのXPRSgetintattribと同様、ここで与えられる制御名に接続語XPRS_を付けてください。コンソール XpressユーザはXpress-MP Essentialsにあるように、この接続語なしで制御を使います。これらの制御を使うにはルーチンを呼ぶ前に設定してください。

例

1 ~ 2の例を用いてルーチンの使い方を説明します。

補足

ルーチンの説明に含まれない追加情報を終わりに示します。

関連事項

関連するルーチンやトピックのリストを比較やリファレンスとして最後に示します。

Xpress-Optimizer Reference Manual Console and Library Functions 44

XPRSaddcols 6

XPRSaddcols

目的

入力ルーチンにより行列をOptimizerに引き渡した後、行列に列を加えます。

概要

```
int XPRSaddcols(XPRSprob prob, int newcol, int newnz,
double *objx, int *mstart, int *mrwind, double *dmatval,
double *bd1, double *bdu);
```

引数

関係する制御

Integer

Double

prob	現在の問題
newcol	新しい列の数
newnz	追加列における新たな非零要素数
objx	新しい列の目的関数の係数を含む長さnewcolの倍精度型配列
mstart	各列の要素の開始のmrwind, dmatval配列のオフセットを含む長さnewcol+1の整数型配列
mrwind	各列の要素の行添字を含む長さnewnzの整数型配列.
dmatval	要素値を含む長さnewnzの倍精度型配列
bdl	追加列の下限を含む長さnewcolの倍精度型配列
bdu	追加列の上限を含む長さnewcolの倍精度型配列
EXTRACOLS	追加できる列の数
EXTRAELEMS	追加できる行列要素数
EXTRAMIPENTS	追加できる離散要素数
MATRIXTOL	行列要素の零許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 45

XPRSaddcols 6

例

ここでは二つの問題を考えます。

XPRSaddcolsを使用して、(a)を(b)に変換し、次にXPRSaddnamesを使用して、新しい変数に名前をつけます。

```
obj[0] = 3;
mstart[] = {0, 3};
mrwind[] = {0, 1, 3};
matval[] = {2.0, 1.0, 3.0};
bdl[0] = 0.0; bdu[0] = 12.0;
...
XPRSaddcols(prob, 1, 3, obj, mstart, mrwind, matval, bdl, bdu);
XPRSaddnames(prob, 2, "z", 2, 2);
```

補足

1. 効果を最大化するため、問題をロードする前にEXTRACOLS, EXTRAELEMS, EXTRAMIPENTS 制御を設定して、追加する行や要素のためのスペースを確保してください。
2. ライブラリヘッダファイル内に定義される倍精度型定数XPRS_PLUSINFINITY, XPRS_MINUSINFINITYは、上下界値配列において各々プラス・マイナス無限大を表すのに使われます。

関連事項

XPRSaddnames, XPRSaddrows, XPRSalter, XPRSDelcols.

(a) Maximize (b) Maximize

Subject to: Subject to:

```
2x y + 2x y 3z ++
x 4y + 24 .
y 5 .
3x y + 20 .
x y + 9 .
x 4y 2z ++ + 24 .
y z + 5 .
3x y + 20 .
```

$xy3z++9.$

$z12.$

Xpress-Optimizer Reference Manual Console and Library Functions 46

XPRSaddcuts 6

XPRSaddcuts

目的

現在のノードで行列に直接カットを追加します。現在のノードで行列に加えられたカットと現在のノードで削除されなかったカットは自動的にカットプールに追加されます。カットプールに追加されたカットは子ノードでは、自動的に戻されます。

概要

```
int XPRSaddcuts(XPRSprob prob, int ncuts, int *mtype,
char *qrtype, double *drhs, int *mstart, int *mcols,
double *dmatval);
```

引数

prob	現在の問題
ncuts	追加されるカットの数。
mtype	カットタイプを含む長さncutsの整数型配列。ユーザはカットタイプとして任意の正の整数を選択でき、カットタイプはユーザ提供のパラメーターを使う他のカットマネージャルーチン内のカットを区別するのに使われます。カットタイプは整数やビットマップで設定できます。XPRSdelcutsを参照してください
qrtype	以下の行タイプを含む長さncutsの文字型配列 L 行を示す G 行を示す E 行を示す
drhs	カットの右辺要素を含む長さncutsの倍精度型配列
mstart	各カットの開始を示すmcols, dmatval配列へのオフセットを含む整数型配列。これは、カットncuts+1が開始点として、最終要素であるmstart[ncuts]を含む長さncuts+1の配列です
mcols	カット内の列添字を含む長さmstart[ncuts]-1の整数型配列
dmatval	カットの行列値を含む長さmstart[ncuts]-1の倍精度型配列
.	.
.	.
=	.

Xpress-Optimizer Reference Manual Console and Library Functions 47

XPRSaddcuts 6

関係する制御

Double

補足

1. カットの列と要素はXPRSaddcutsに渡されるmcols, dmatval配列に隣接して保存されます。各カットの開始点は mstart配列に保存されます。最後のカットの長さを決定するために、mstart配列はmcolsとdmatvalの中でカットがncuts+1を開始点としてその位置を含む配列の最後の要素を持つ長さncuts+1の配列です。mstart[ncuts]は追加されるカットにおける非零要素数を示します。

2. 行列に追加されるカットは、常に行列の最後に置かれ、行の数は元のカットの数に追加されて設定されます。ncutsが追加された場合、行0,....., ROWS-ncuts-1はもとの行であり、行ROWS-ncuts,....., ROWS-1は追加カットの分です。カットの数はCUTS問題特性により調べられます。

関連事項

XPRSaddrows, XPRSdelcpcuts, XPRSdelcuts, XPRSgetcpcutlist,
 XPRSgetcutlist, XPRSloadcuts, XPRSstorecuts, 4章「カットマネージャのはたらき」
 MATRIXTOL 行列要素の零許容範囲
 Xpress-Optimizer Reference Manual Console and Library Functions 48

XPRSaddnames 6

XPRSaddnames

目的

モデルをロードするとき、モデルの行や列は、それらと関係する名前を持たないかもしれませんが。これは行や列がそれらの通し番号により表されるためで、重要ではないかもしれませんが。しかし、ASCII形式解ファイルに列や行の名前が現れるようにしたい場合、XPRSaddnamesで行や列の範囲に名前をつけることができます。

概要

```
int XPRSaddnames(XPRSprob prob, int type, char *cnames,
int first, int last);
```

引数

関係する制御

Integer

例

変数名 (aとb), 目的関数(profit), 制約条件名(firstとsecond) を問題に追加する。

```
char rnames[] = "profit\0first\0second"
char cnames[] = "a\0b";
```

...

```
XPRSaddnames(prob, 1, rnames, 0, nrow-1);
XPRSaddnames(prob, 2, cnames, 0, ncol-1);
```

prob	現在の問題
type	1 行の名前 2 列の名前
cnames	最後はNULLで終わり、それぞれの名前が最大でMPSNAMELENGTH+1文字数のNULL文字を含む文字バッファ。この制御を変更する場合は、問題をロードする前に実施しなくてはなりません
first	行, 列のレンジの開始
last	行, 列のレンジの終了
MPSNAMELENGTH	最大名前文字長さ

Xpress-Optimizer Reference Manual Console and Library Functions 49

XPRSaddnames 6

関連事項

XPRSaddcols, XPRSaddrows, XPRSgetnames.
 Xpress-Optimizer Reference Manual Console and Library Functions 50

XPRSaddrows 6

XPRSaddrows

目的

入力ルーチンを用いてOptimizerに行列を引き渡した後、行列に行を追加します。

概要

```
int XPRSaddrows(XPRSprob prob, int newrow, int newnz,
char *qrtype, double *rhs, double *range, int *mstart,
int *mclind, double *dmatval);
```

引数

prob	現在の問題
newrow	新しい行の数
newnz	追加行における非零要素数
qrtype	行タイプを含む長さnewrowの文字配列
	L 制約条件の不等号関係 < を示す
	E 制約条件の等号関係 = を示す
	G 制約条件の不等号関係 > を示す
	R 範囲の制約条件を示す
	N 非束縛制約を示す
rhs	右辺要素を含む長さnewrowの倍精度型配列
range	各行の要素の開始のmclind, dmatval配列のオフセットを含む長さnewrowの整数型配列。範囲制約がない場合は、NULLになります。range配列内の値はRタイプ行のみ読まれます。他のタイプ行の要素は無視されます
mstart	各行の要素の開始の配列mclindとdmatvalのオフセットを含む長さnewrow+1の整数型配列
mclind	各行の要素の(隣接した)列添字を含む長さnewnzの整数型配列
dmatval	(隣接した)要素値を含む長さnewnzの倍精度型配列
.	
=	
.	

Xpress-Optimizer Reference Manual Console and Library Functions 51

XPRSaddrows 6

関係する制御

Integer

Double

例

問題に行を追加し、それにNewRow:という名前をつけます。

```
qrtype[0] = 'L';
rhs[0] = 25.0;
mstart[] = {0, 3};
mclind[] = {0, 1, 2};
dmatval[] = {8.0, 9.0, 10.0};
...
XPRSaddrows(prob, 1, 3, qrtype, rhs, NULL, mstart, mclind,
dmatval);
XPRSaddnames(prob, 1, "NewRow", 4, 4);
```

補足

最大の効果をあげるために、問題をロードする前にEXTRAROWS , EXTRAELEMENTS制御を設定して追加行や要素のためのスペースを確保してください。

関連事項

XPRSaddcols, XPRSaddcuts, XPRSaddnames, XPRSdelrows.

EXTRAELEMENTS	追加できる行列要素の数
EXTRAROWS	追加できる行の数
MATRIXTOL	行列要素の零許容範囲

Maximize:

Subject to:

2x y 3z ++

x 4y 2z ++ 24 .

y z + 5 .

3x y 20 . +

x y 3z ++ 9 .

8x 9y 10z ++ 25 .

Xpress-Optimizer Reference Manual Console and Library Functions 52

XPRSaddsetnames 6**XPRSaddsetnames****目的**

離散要素をもつモデルをロードしたとき、特殊順序集合は、それに関係した名前を持たないかもしれません。ASCII形式解ファイルに名前が現れるようにしたい場合、この関数により集合の範囲に名前をつけることができます。

概要

```
int XPRSaddsetnames(XPRSprob prob, char *names, int first,
int last);
```

引数**関係する制御**

Integer

例

問題に set1 , set2 という集合名をつけます。

```
char snames[] = "set1\0set2"
```

```
...
```

```
XPRSaddsetnames(prob, snames, 0, 1);
```

関連事項

XPRSaddnames, XPRSloadglobal, XPRSloadqglobal.

prob	現在の問題
names	名前の最後がNULLで終わる最大でMPSNAMELENGTH+1文字数のNULL文字を含むバッファ。この制御を変更する場合は、問題をロードする前に実施しなくてはなりません
first	集合範囲の開始
last	集合範囲の終了
MPSNAMELENGTH	最大名前文字長さ

Xpress-Optimizer Reference Manual Console and Library Functions 53

XPRSalter

ALTER 6

XPRSalter ALTER

目的

現在の問題における行列要素や右辺，制約を変更します。

概要

```
int XPRSalter(XPRsprob prob, const char *filename);
ALTER [filename]
```

引数

関係する制御

Integer

Double

例 1 (ライブラリ)

以下の呼び出しはファイル名を指定していないので、ファイル`problem_name.alt`が読み込まれ、そこから、現在の行列を変更するためのコマンドが取り込まれます。

```
XPRSalter(prob, "");
```

例 2 (コンソール)

次の例は、ファイル`fred.alt`が読み込まれ、そこから現在の行列を変更するための指示が取り込まれます。

```
ALTER fred
```

補足

1. ファイル`filename.alt`が読まれます。それは付録A.6「行列変更ファイル(.alt)」に説明されている形式で、行列の修正についての記述が含まれるASCIIファイルです。MPS REVERSEデータのMODIFY形式もサポートされています。

<code>prob</code>	現在の問題
<code>filename</code>	読みこむファイルを指定する200文字以内の文字列。省略した場合は、デフォルト <code>problem_name</code> に拡張子 <code>.alt</code> を付けたものになります
<code>EXTRAELEMS</code>	追加できる行列要素の数
<code>MATRIXTOL</code>	行列要素の零許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 54
XPRSalter

ALTER 6

2. コマンド`XPRSalter (ALTER)`と制御`EXTRAELEMS`を使うとメモリ内の問題の値や制約の意味を変えられます。効果を最大化するため、追加する行列要素のスペースを確保してください。追加できる要素数の最大値の定義は`XPRsreadprob(READPROB)`の前に実施してください。
3. 事前分析済みの整数モデルを変更することはできません。最適化後に変更したい場合は、最適化前に `PRESOLVE`を0に設定して事前分析をしないか`XPRsreadprob (READPROB)`でモデルを読みなおしてください。

関連事項

付録A.6「行列変更ファイル(.alt)」

Xpress-Optimizer Reference Manual Console and Library Functions 55

XPRsbranchcut 6

XPRsbranchcut

目的

ユーザ離散要素上で分岐するために適用されるカットプールにおいてカットと（または）上下界値の添字を指定します。このルーチンは、ユーザ分割コールバック関数XPRSsetcbsepnodeからのみ呼び出すことができます。

概要

```
int XPRSbranchcut(XPRSprob prob, int ncuts, int *mindex);
```

引数**関係する制御**

なし

例

この例は、整数解探索のためのユーザ分割コールバック関数を定義します。

```
XPRSsetcbsepnode(prob, sepNode, NULL);
```

ここで、関数sepNodeは次のように定義されます。

```
int sepNode(XPRSprob prob, void *my_object, int ibr,
int iglsel, int ifup, double curval)
```

```
{
int index;
double dbd;
if(ifup)
{
dbd = floor(xval);
XPRSstorebounds(prob, 1, &iglsel, "U", &dbd, &index);
}
else
{
dbd = ceil(xval);
```

prob	現在の問題
ncuts	適用されるカットと（あるいは）上下界値の数
mindex	（XPRSstoreboundsで返された）上下界値、または適用されるカットプール内のカットの添字を含む配列

Xpress-Optimizer Reference Manual Console and Library Functions 56

XPRSbranchcut 6

```
XPRSstorebounds(prob, 1, &iglsel, "L", &dbd, &index);
}
XPRSbranchcut(prob, 1, &index);
return 0;
}
```

関連事項

XPRSgetcpcutlist, XPRSloadcuts, XPRSsetcbestimate,

XPRSsetcbsepnode, XPRSstorecuts, 5.4章「カットマネージャーのはたらき」

Xpress-Optimizer Reference Manual Console and Library Functions 57

XPRSbtran 6**XPRSbtran****目的**

現在の基底の逆数によってユーザに与えられる(行)ベクトルの事後掛け算。

概要

```
int XPRsbtran(XPRsprob prob, double *vec);
```

引数**関係する制御**

Double

例

全ての配列が次元化されているとして、制約番号*irow*の（スケーリングされていない）タブロー行*z*を求めます。

```
/* Minimum size of arrays:
y: nrow + ncol;
mstart: 2;
mrowind, dmatval: nrow. */
/* set up the unit vector y to pick out row irow */
for(i = 0; i < nrow; i++)
y[i] = 0.0;
y[irow] = 1.0;
rc = XPRsbtran(prob,y); /* y = e*B^{-1} */
/* Form z = y * A */
for(j = 0; j < ncol, j++){
rc = XPRsgetcols(prob,mstart,mrowind,dmatval,nrow,
&nelt,j,j);
for(d = 0.0, ielt = 0, ielt < nelt; ielt++)
```

prob

現在の問題

vec

逆基底が掛けられる値を含む長さROWSの倍精度型配列。変換された値は配列内に現れます

ETATOL

イータ要素の零許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 58

XPRsbtran 6

```
d += y[mrowind[ielt]] * dmatval[ielt];
y[nrow + j] = d;
}
```

補足

行列が事前分析済みの場合、XPRsbtranはその事前分析済み問題の基底と動作します。

関連事項

XPRsftran.

Xpress-Optimizer Reference Manual Console and Library Functions 59

XPRSchgbounds 6**XPRSchgbounds****目的**

行列の列の上下界値を変更します。

概要

```
int XPRSchgbounds(XPRsprob prob, int nbnds, int *mindex,
char *qbtype, double *bnd);
```

引数**関係する制御**

なし

例

現在の問題の列0について上限0.5となるように変更します。

```
mindex[0] = 0;
qbtype[0] = 'U';
bnd[0] = 0.5;
XPRSchgbounds(prob, 1, mindex, qbtype, bnd);
```

補足

1. mindex配列に列添字が2回現れると、一度で変数の上限と下限の両方を変更することができます。
2. XPRSchgboundsは事前分析後の問題に適用され、事前分析された問題を参照します。

prob	現在の問題
nbnds	変更する上下界値の数
mindex	上下界値が変更される列添字を含む大きさnbndsの整数型配列
qbtype	変更される上下界値タイプを示す長さnbndsの文字配列
	U 上限を変更
	L 下限を変更
	B 両方の上下界値を変更。列は固定
bnd	新しい上下界値値を与える長さnbndsの倍精度型配列

Xpress-Optimizer Reference Manual Console and Library Functions 60

XPRSchgbounds 6

3. ライブラリヘッダファイルに定義された倍精度型定数XPRS_PLUSINFINITY, XPRS_MINUSINFINITYは、上下界値(bnd)配列内で各々プラス・マイナス無限を表すのに使うことができます。

関連事項

XPRSgetlb, XPRSgetub, XPRSstorebounds.

Xpress-Optimizer Reference Manual Console and Library Functions 61

XPRSchgcoef 6**XPRSchgcoef****目的**

行列内の1つの係数を変更します。その係数が存在しなければ、新しい係数を行列に追加します。多くの係数が行列の行に追加される場合は、既存の行を削除し、新しい行を追加することが効果的です。

概要

```
int XPRSchgcoef(XPRSprob prob, int irow, int icol,
double dval);
```

引数**関係する制御**

Double

例

行列の2行目, 1列目の要素を0.33に変更します。

```
XPRSchgcoef(prob, 2, 1, 0.33);
```

補足

XPRSchgcoefを何度も呼びだすよりもXPRSchgmcoefを使う方が効率的で、そのような場合にはそうすべきです。

関連事項

XPRSaddcols, XPRSaddrows, XPRSchgmcoef, XPRSchgmqobj,
XPRSchgobj, XPRSchgqobj, XPRSchgrhs, XPRSgetcols, XPRSgetrows.

prob	現在の問題
irow	係数の行添字
icol	係数の列添字
dval	係数の新しい値。dvalが零の場合、係数は削除されます
MATRIXTOL	行列要素の零許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 62

XPRSchgcoltype 6**XPRSchgcoltype****目的**

行列内の列のタイプを変更します。

概要

```
int XPRSchgcoltype(XPRSprob prob, int nels, int *mindex,
char *qctype);
```

引数**関係する制御**

なし

例

行列の3列目と5列目を整数、バイナリに各々変更します。

```
mindex[0] = 3; mindex[1] = 5;
qctype[0] = 'I'; qctype[1] = 'B';
XPRSchgcoltype(prob, 2, mindex, qctype);
```

補足

列のタイプは、MIP探索の開始前でのみ変更できます。問題が事前分析済みのときに XPRSchgcoltype を呼び出すと、事前分析された列番号が示されます。列を部分整数、半連続、半連続整数変数に変更することはできません。

関連事項

XPRSaddcols, XPRSchgrowtype, XPRSdelcols, XPRSgetcoltype.

prob	現在の問題
nels	変更する列の数
mindex	列の添字を含む長さnelsの整数型配列
qctype	新しい列タイプを与える長さnelsの文字配列
	C 連続列
	B バイナリ列
	I 整数列

Xpress-Optimizer Reference Manual Console and Library Functions 63

XPRSchgmcoef 6**XPRSchgmcoef****目的**

行列内の複数の係数を変更します。係数が存在しない場合は、行列に追加します。行列の1つの行に多くの係数を追加する場合は、行列の古い行を削除して新しい行を追加する方が効果的です。

概要

```
int XPRSchgmcoef(XPRsprob prob, int nels, int *mrow,
int *mcol, double *dval);
```

引数**関係する制御**

Double

例

```
mrow[0] = 0; mrow[1] = 3;
mcol[0] = 1; mcol[1] = 5;
dval[0] = 2.0; dval[1] = 0.0;
XPRSchgmcoef(prob, 2, mrow, mcol, dval);
これにより 2 つの要素が 2.0, 0.0 に変わります。
```

補足

多くの係数を変更する場合、XPRSchgmcoef を使う方が XPRSchgcoef を繰り返し呼び出すより効果的です。

prob	現在の問題
nels	新しい係数の数
mrow	変更する係数の行添字を含む長さ nels の整数型配列
mcol	変更する係数の列添字を含む長さ nels の整数型配列
dval	新しい係数値を含む長さ nels の倍精度型配列。 dval の要素が零のときは、係数は削除されます
MATRIXTOL	行列要素の零許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 64

XPRSchgmcoef 6**関連事項**

XPRSchgcoef, XPRSchgmqobj, XPRSchgobj, XPRSchgqobj, XPRSchgrhs, XPRsgetcols, XPRsgetrhs.

Xpress-Optimizer Reference Manual Console and Library Functions 65

XPRSchgmqobj 6**XPRSchgmqobj****目的**

目的関数の複数の二次係数を変更します。係数が存在しない場合は、新しい係数を目的関数に追加します。

概要

```
int XPRSchgmqobj(XPRsprob prob, int nels, int *mqcol1,
int *mqcol2, double *dval);
```

引数**関係する制御**

なし

例

次のコードは、目的関数の項を決定します。

```
mqcol1[0] = 0; mqcol2[0] = 0; dval[0] = 6.0;
mqcol1[1] = 1; mqcol2[1] = 0; dval[1] = 3.0;
XPRSchgmqobj(prob, 2, mqcol1, mqcol2, dval);
```


補足

1. 配列 `mqcol1`, `mcol2` 内の列は行列に存在しなくてはなりません。存在しない場合は `XPRSaddcols` で列を追加してください。
2. いくつかの係数を変更する場合には、`XPRSchgqobj` を繰り返し呼び出すよりも `XPRSchgmqobj` を使う方が効果的です。

<code>prob</code>	現在の問題
<code>nels</code>	変更する係数の数
<code>mcol1</code>	各二次の項の最初の変数の列添字を含む大きさ <code>ncol</code> の整数型配列
<code>mcol2</code>	各二次の項の 2 番目の変数の列添字を含む大きさ <code>ncol</code> の整数型配列
<code>dval</code>	係数の新しい値。 <code>dval</code> 内の要素が 0 の場合、相当する要素は削除されます。これらは二次のヘシアン行列の係数です

`6x1()23x1()x2()·3x2()x1()·++()2.`
 Xpress-Optimizer Reference Manual Console and Library Functions 66

XPRSchgmqobj 6**関連事項**

`XPRSchgcoef`, `XPRSchgmcoef`, `XPRSchgobj`, `XPRSchgqobj`, `XPRSgetqobj`.
 Xpress-Optimizer Reference Manual Console and Library Functions 67

XPRSchgobj 6**XPRSchgobj****目的**

目的関数の係数を変更します。

概要

```
int XPRSchgobj(XPRSprob prob, int nels, int *mindex,
double *obj);
```

引数**関係する制御**

なし

例

```
XPRSchgobjを使って目的関数の3つの係数を変更します。
mindex[0] = 0; mindex[1] = 2; mindex[2] = 5;
obj[0] = 25.0; obj[1] = 5.3; obj[2] = 0.0;
XPRSchgobj(prob, 3, mindex, obj);
```

補足

目的関数の定数部分の値は、問題特性 `OBJFIXED` によって得られます。

関連事項

`XPRSchgcoef`, `XPRSchgmcoef`, `XPRSchgmqobj`, `XPRSchgqobj`,
`XPRSgetobj`.

<code>prob</code>	現在の問題
<code>nels</code>	変更する目的関数の係数要素の数
<code>mindex</code>	レンジ要素が変わる列の添字を含む長さ <code>nels</code> の整数型配列。 -1 の添字は、右側の目的関数の定数部分が変わることを示します
<code>obj</code>	新しい目的関数の係数を与える長さ <code>nels</code> の倍精度型配列

Xpress-Optimizer Reference Manual Console and Library Functions 68

XPRSchgqobj 6

XPRSchgqobj

目的

Hessian行列の変数の組(icol, jcol)に相当する目的関数の二次係数の1つを変更します。

概要

```
int XPRSchgqobj(XPRsprob prob, int icol, int jcol,
double dval);
```

引数

関係する制御

なし

例

次のコードは目的関数に項を追加します。

```
icol = jcol = 0; dval = 6.0;
XPRSchgqobj(prob, icol, jcol, dval);
icol = 0; jcol = 1; dval = 3.0;
XPRSchgqobj(prob, icol, jcol, dval);
```

補足

1. 列icol, jcolは行列に存在しなくてはなりません。存在しない場合は、ルーチンXPRSaddcolsを使って追加しなくてはなりません。
2. icolがjcolと等しくない場合、行列要素(icol, jcol)と(jcol, icol)はHessian対称でないように変更されます。

関連事項

XPRSchgcoef, XPRSchgmcoef, XPRSchgmqobj, XPRSchgobj,
XPRSgetqobj.

prob	現在の問題
icol	二次項の第1変数の列添字
jcol	二次項の第2変数の列添字
dval	二次のHessian行列における係数の新しい値。dval内の要素が0のとき、相当する要素は削除されます

$6 \times 1 () 2 3 \times 1 () \times 2 () \cdot 3 \times 2 () \times 1 () \cdot + + () 2 .$

Xpress-Optimizer Reference Manual Console and Library Functions 69

XPRSchgrhs 6

XPRSchgrhs

目的

行列の右辺要素を変更します。

概要

```
int XPRSchgrhs(XPRsprob prob, int nels, int *mindex,
double *rhs);
```

引数

関係する制御

なし

例

行2, 6, 8の3つの右辺を新しい値に変更する。

```
mindex[0] = 2; mindex[1] = 8; mindex[2] = 6;
rhs[0] = 5.0; rhs[1] = 3.8; rhs[2] = 5.7;
XPRSchgrhs(prob, 3, mindex, rhs);
```

関連事項

XPRSchgcoef, XPRSchgmcoef, XPRSchgrhsrange, XPRSgetrhs, XPRSgetrhsrange.

prob	現在の問題
nels	変更する右辺の要素の数
mindex	右辺要素が変わる行の添字を含む長さnelsの整数型配列
rhs	右辺の値を与える長さnelの倍精度型配列

Xpress-Optimizer Reference Manual Console and Library Functions 70

XPRSchgrhsrange 6**XPRSchgrhsrange****目的**

問題の行列の行の範囲を変更します。

概要

```
int XPRSchgrhsrange(XPRSprob prob, int nels, int *mindex,
double *rng);
```

引数**関係する制御**

なし

例

問題の制約条件を変更します。

```
mindex[0] = 5; rng[0] = 2.0;
XPRSchgrhsrange(prob, 1, mindex, rng);
```

補足

行の指定範囲が*r*の場合、*r*の行のタイプと値に依存して実施されます。このルーチンを使用すれば、範囲のない行に変更できます。

関連事項

XPRSchgcoef, XPRSchgmcoef, XPRSchgrhs, XPRSgetrhsrange.

prob	現在の問題
nels	変更する範囲要素の数
mindex	範囲要素が変更される行の添字を含む長さnelsの整数型配列
rng	範囲値を与える長さnelsの倍精度型配列

***r*の値 行タイプ 効果**

```
,
,
xy + 10 . 8 xy + 10 ..
r0 . b = b . b r - ajxj _ b ..
r0 . b . b ajxj _ b r + ..
r0 < b = b . b ajxj _ b r - ..
r0 < b . b r + ajxj _ b ..
```

Xpress-Optimizer Reference Manual Console and Library Functions 71

XPRSchgrowtype 6**XPRSchgrowtype**

目的

行列の行のタイプを変更します。

概要

```
int XPRSchgrowtype(XPRSprb prob, int nels, int *mindex,
char *qrtype);
```

引数**関係する制御**

なし

例

行 4 が等式行に変更されます。

```
mindex[0] = 4; qrtype[0] = 'E';
XPRSchgrowtype(prob, 1, mindex, qrtype);
```

補足

ある行において初めに`R`または`L`タイプに変更し、`XPRSchgrhsrange`を使用して範囲を変更することで、レンジタイプの行に変更できます。

関連事項

`XPRSaddrows`, `XPRSchgcoltype`, `XPRSchgrhs`, `XPRSchgrhsrange`,
`XPRSDelrows`, `XPRSgetrowrange`, `XPRSgetrowtype`.

<code>prob</code>	現在の問題
<code>nels</code>	変更する行の数
<code>mindex</code>	行の添字を含む長さ <code>nels</code> の整数型配列
<code>qrtype</code>	新しい行のタイプを与える長さ <code>nels</code> の文字配列
	<code>L</code> 行を示す
	<code>E</code> 行を示す
	<code>G</code> 行を示す
	<code>R</code> レンジ行を示す
	<code>N</code> 自由行を示す

.

=

.

Xpress-Optimizer Reference Manual Console and Library Functions 72

XPRScopycallbacks 6**XPRScopycallbacks****目的**

ある問題のコールバック関数を別の問題にコピーする。

概要

```
int XPRScopycallbacks(XPRSprb dest, XPRSprb src);
```

引数**関係する制御**

なし

例

問題`prob1`のメッセージコールバック関数`callback`を設定し、それを問題`prob2`にコピーします。

```
XPRScreateprob(&prob1);
XPRSSetcbmessage(prob1, callback, NULL);
```

```
XPRSccreateprob(&prob2);
XPRScopycallbacks(prob2,prob1);
```

関連事項

XPRScopycontrols, XPRScopyprob.

dest コールバックがコピーされる先の問題

src コールバックがコピーされる元の問題

Xpress-Optimizer Reference Manual Console and Library Functions 73

XPRScopycontrols 6

XPRScopycontrols

目的

ある問題の制御を別の問題にコピーします。

概要

```
int XPRScopycontrols(XPRScprob dest, XPRScprob src);
```

引数

関係する制御

なし

例

問題prob1の事前分析をオフにして、これと他の制御値を問題prob2にコピーします。

```
XPRSccreateprob(&prob1);
XPRScsetintcontrol(prob1,XPRS_PRESOLVE,0);
XPRSccreateprob(&prob2);
XPRScopycontrols(prob2,prob1);
```

関連事項

XPRScopycallbacks, XPRScopyprob.

dest 制御をコピーする先の問題名

src 制御をコピーする元の問題名

Xpress-Optimizer Reference Manual Console and Library Functions 74

XPRScopyprob 6

XPRScopyprob

目的

ある問題に定義された情報を別の問題にコピーします。

概要

```
int XPRScopyprob(XPRScprob dest, XPRScprob src,
char *probname);
```

引数

関係する制御

なし

例

問題prob1から問題prob2に問題, 制御, コールバックをコピーします。

```
XPRScprob prob1, prob2;
...
XPRSccreateprob(&prob2);
XPRScopyprob(prob2,prob1,"MyProb");
XPRScopycontrols(prob2,prob1);
XPRScopycallbacks(prob2,prob1);
```

補足

XPRScopyprobは問題だけをコピーし、問題に関するコールバックや制御についてはコピーしません。これらについては、XPRScopycallbacks, XPRScopycontrolsを使って各々コピーしてください。

関連事項

XPRScopycallbacks, XPRScopycontrols, XPRScreateprob.

dest	情報のコピー先となる新しい問題のポインタ
src	情報のコピー元となる古い問題のポインタ
probname	コピーされた問題の名前を含む200文字以下の文字列。ファイルへの出力が見込まれる場合、特に整数問題の場合、これはユニークでなくてはなりません

Xpress-Optimizer Reference Manual Console and Library Functions 75

XPRScreateprob 6**XPRScreateprob****目的**

Optimizer内に新しい問題を設定します。

概要

```
int XPRScreateprob(XPRScprob *prob);
```

引数**関係する制御**

なし

例

myprobを含む問題を作成します。

```
XPRScprob prob;
XPRScinit(NULL);
XPRSccreateprob(&prob);
XPRScreadprob(prob, "myprob", "");
```

補足

1. XPRSccreateprobは XPRScinitの後、他のOptimizerルーチンを使用する前に呼びだしてください。
2. ライセンスによっては、この方法で問題をいくらかでも作成できます。作業が終了したら、XPRScdestroyprobにより全ての問題を削除してください。

関連事項

XPRScdestroyprob, XPRSccopyprob, XPRScinit.

prob	新しい問題が保持する変数のポインタ
------	-------------------

Xpress-Optimizer Reference Manual Console and Library Functions 76

XPRScdelcols 6**XPRScdelcols****目的**

行列から列を削除します。

概要

```
int XPRScdelcols(XPRScprob prob, int ncols, int *mindex);
```

引数**関係する制御**

なし

例

行列から列 3 を削除します。

```
mindex[0] = 3;
XPRSdelcols(prob, 1, mindex);
```

補足

問題から列が削除されると、残る列の数は減少し、番号 0 ~ COLS-1 になります。ここで、COLS は行列内で削除されなかった列の数を含む問題特性です。

関連事項

XPRSaddcols, XPRSchgcoltype, XPRSdelrows.

prob	現在の問題
ncols	削除する列数
mindex	削除する列を含む長さncolsの整数型配列

Xpress-Optimizer Reference Manual Console and Library Functions 77

XPRSdelcpcuts 6**XPRSdelcpcuts****目的**

分枝限定探索では、カットは子ノードで適用されるためにカットプールに保存されます。これらのカットは、与えられたノードからXPRSdelcpcutsにより取り除かれますが、多くの場合、適用の際、カットプールから完全に切り除くことが望ましいです。これにはXPRSdelcpcutsを使います。

概要

```
int XPRSdelcpcuts(XPRSprob prob, int itype, int interp,
int ncuts, int *mcutind);
```

引数**関係する制御**

なし

関連事項

XPRSaddcuts, XPRSdelcuts, XPRSloadcuts, 5.4章「カットマネージャのはたらき」

prob	現在の問題
itype	カットタイプ
interp	カットタイプの解釈
	-1 カットを全て除きます
	1 カットタイプを数で扱います
	2 カットタイプをビットマップで扱います。ビットのいずれかがitypeに設定されたビットと一致する場合、削除します
	3 カットタイプをビットマップで扱います。全てのビットがitypeに設定されたビットと一致する場合、削除します
ncuts	削除するカット数。値-1は全てのカットを削除することを示します
mcutind	削除するカットの添字番号を含む整数型配列。ncutsが-1の場合、こ

の配列はNULLになります。そうでなければ長さncutsになります
Xpress-Optimizer Reference Manual Console and Library Functions 78

XPRScelcuts 6

XPRScelcuts

目的

現在のノードで行列からカットを削除します。自動的に戻された親ノードからのカットも、現在のノードにXPRSaddcuts, XPRSloadcutsを用いて追加されたカットが削除されます。削除されるカットは様々な方法で指定できます。基準により対象外とされるカットは削除されません。

概要

```
int XPRScelcuts(XPRSprob prob, int ibasis, int itype,
int interp, double delta, int num, int *mcutind);
```

引数

関係する制御

なし

prob	現在の問題
ibasis	1 が設定されると、その基底を有効とします。0 が設定されると、非基底スラックのカットは削除されます
itype	削除されるカットのタイプ
interp	カットitypeの解釈
	-1 全カットタイプを削除します
	1 カットタイプを数で扱います
	2 カットタイプをビットマップで扱います。ビットのいずれかがitypeに設定されたビットと一致する場合、削除されます
	3 カットタイプをビットマップで扱います。全てのビットがitypeに設定されたビットと一致する場合、削除されます
delta	相対スラック値がdeltaより大きいカットのみ削除されます。全カットを削除するとき、この引数をXPRS_MINUSINFINITYに設定します
num	カットリストによって削除されるカットの数。-1が設定されると、全てのカットが対象となります
mcutind	削除するカットの添字番号を含む整数型配列。numが-1に設定されると、この配列はNULLになりますが、そうでなければ長さnumです

Xpress-Optimizer Reference Manual Console and Library Functions 79

XPRScelcuts 6

補足

1. 基底スラックのカットのみを取り除くことは、通常最もよいですが、そうしない場合、基底は有効でなくなり、最適基底を回復するために何度か繰返しが必要です。ibasisパラメータが1に設定されると、非基底スラックのカットは他のパラメータがそれらを削除するように設定してあっても、削除されません。ibasisパラメータをいつも1にしておくことを強くお勧めします。
2. 削除されるカットは、カットのスラック値のサイズによっても指定されます。deltaパラ

メータより大きいスラック値のカットのみ削除されます。

- 削除されるカットの添字のリストも示すことができます。自動的に戻されたカットは $0, \dots, \text{ncuts}-1$ の添字を持ちます (ここで ncuts は親ノードでのカットの数で、追加されたカットは $\text{ncuts}, \dots, \text{ncuts}+\text{nadded}-1$ の添字を持ち、 nadded は現在のノードに追加されたカットの数です)。行列内のカットの数は問題特性 CUTS により決められます。

関連事項

XPRSaddcuts, XPRSdelcpcuts, XPRSloadcuts, 5.4章「カットマネージャのはたらき」.
Xpress-Optimizer Reference Manual Console and Library Functions 80

XPRSdelnode 6

XPRSdelnode

目的

分枝限定木探索において未探索ノードのリストから特定のノードを削除します。

概要

```
int XPRSdelnode(XPRSprob prob, int inode, int ifboth);
```

引数

関係する制御

なし

例

```
XPRSdelnode(prob, 10, 0);
```

木探索におけるノード番号10とその次の子ノードを削除します。

補足

このルーチンは分枝限定探索のコールバックから最も効果的に呼びだされます。

関連事項

なし

prob	現在の問題
inode	削除するノードの数
ifboth	以下のどちらか1つのフラグ
	0 次の子ノードを削除します
	1 両方の子ノードを削除します

Xpress-Optimizer Reference Manual Console and Library Functions 81

XPRSdelrows 6

XPRSdelrows

目的

入力ルーチンを使ってOptimizerに渡された行列から行を削除します。

概要

```
int XPRSdelrows(XPRSprob prob, int nrows, int *mindex);
```

引数

関係する制御

なし

例

行0と10を行列から削除します。

```
mindex[0] = 0; mindex[1] = 10;
```

```
XPRSdelrows(prob, 2, mindex);
```

補足

問題から行が削除されると、残された行の数は減少し、行は番号0~ROWS-1になります。ここでROWSは行列で削除されなかった行の数を含む問題特性です。

関連事項

XPRSaddrows, XPRSchgrowtype, XPRSdelcols.

prob	現在の問題
nrows	削除する行の数
mindex	削除する行を含む長さnrowsの整数型配列

Xpress-Optimizer Reference Manual Console and Library Functions 82

XPRStdestroyprob 6**XPRStdestroyprob****目的**

指定された問題を削除し、操作、最適化後、その問題に関連するメモリを開放します。

概要

```
int XPRStdestroyprob(XPRStprob prob);
```

引数**関係する制御**

なし

例

myprobと呼ばれる問題を作成、ロードし、解いた後、割り当てられたリソースを開放します。

```
XPRStcreateprob(&prob);
XPRStreadprob(prob, "myprob", "");
XPRStmaxim(prob, "");
XPRStdestroyprob(prob);
```

補足

作業終了後は全ての問題を削除しなくてはなりません。XPRStdestroyprobにNULL問題ポインタが渡されると、エラーは起きません。

関連事項

XPRStcreateprob, XPRStfree, XPRStinit.

prob	削除される問題
------	---------

Xpress-Optimizer Reference Manual Console and Library Functions 83

XPRStfixglobal**FIXGLOBAL 6****XPRStfixglobal FIXGLOBAL****目的**

すべての離散要素を解ファイルの値に固定します。これは離散変数が最適値に固定された後、連続変数の被約費用を求めるのに有用です。

概要

```
int XPRStfixglobal(XPRStprob prob);
FIXGLOBAL
```

引数**関係する制御**

なし

例 1 (ライブラリ)

線形問題を解くのに先だって問題`myprob`を読みなおしたり`XPRSfixglobal`を使用する前に、問題`myprob`の整数解を探索します。

```
XPRSreadprob(prob, "myprob", "");
XPRSminim(prob, "g");
XPRSreadprob(prob, "myprob", "");
XPRSfixglobal(prob);
XPRSminim(prob, "");
XPRSwriteprtsol(prob);
```

例 2 (コンソール)

コンソールで同様のことを実施するのは以下の通りです。

```
READPROB
MINIM -g
READPROB
FIXGLOBAL
MINIM
PRINTSOL

prob                現在の問題
Xpress-Optimizer Reference Manual Console and Library Functions 84
XPRSfixglobal
```

FIXGLOBAL 6**補足**

1. `XPRSfixglobal (FIXGLOBAL)`は事前分析された問題には使用できません。
2. このコマンドを呼び出すことにより、離散要素は解ファイルの値に固定されます。(デフォルト)事前分析オプションを使って`XPRSglobal (GLOBAL)`を呼ぶ場合には、`XPRSfixglobal (FIXGLOBAL)`によって呼ぶ前に問題を入力し直す必要があります。
3. このコマンドは離散要素が固定された後、行列内の連続変数のリデュースコストを調べるのに有用です。`XPRSfixglobal (FIXGLOBAL)`を呼んだ後、`XPRSrange (RANGE)`を使ってMIP問題中の連続変数に関する感度分析が実行可能です。

関連事項

`XPRSglobal (GLOBAL)`, `XPRSrange (RANGE)`.
Xpress-Optimizer Reference Manual Console and Library Functions 85

XPRSfree 6**XPRSfree****目的**

割り当てられたメモリを開放し、全ての開いているファイルを閉じます。

概要

```
int XPRSfree(void);
```

引数

なし

関係する制御

なし

例

問題`prob`に割り当てられたリソースを開放し、整理してから終了します。

```
XPRSdestroyprob(prob);
XPRSfree();
```

```
return 0;
```

補足

XPRsfreeを呼んだ後、最初にXPRSinitを再度呼びださずにライブラリ関数を使うことはできません。

関連事項

XPRsdestroyprob, XPRSinit.

Xpress-Optimizer Reference Manual Console and Library Functions 86

XPRsftran 6

XPRsftran

目的

現在の行列の逆行列によりユーザに与えられる(列)ベクトルの事前掛け算。

概要

```
int XPRsftran(XPRSprob prob, double *vec);
```

引数

関係する制御

Double

例 1

全ての配列が次元化されているとして、構造変数番号jcolの(スケーリングされていない)タブロー列を求めます。

```
/* Min size of arrays: mstart: 2; mrowind, dmatval & y:
nrow. */
/* Get column as loaded originally, in sparse format */
rc = XPRSgetcols(prob, mstart, mrowind, dmatval, nrow,
&nelt, jcol, jcol);
/* Unpack into the zeroed array */
for(i = 0; i < nrow; i++)
y[i] = 0.0;
for(ielt = 0; ielt < nelt; ielt++)
y[mrowind[ielt]] = dmatval[ielt];
rc = XPRsftran(prob,y);
```

例 2

全ての配列の次元が定められているとして、行番号irowのスラック値の(スケーリングされていない)タブロー列を求めます。

prob	現在の問題
vec	逆基底が乗じられる値を含む長さROWSの倍精度型配列。変換された値は配列内に現れます。

ETATOL	イータ要素の零許容範囲
--------	-------------

Xpress-Optimizer Reference Manual Console and Library Functions 87

XPRsftran 6

```
/* Min size of arrays: y: nrow */
/* Set up the original slack column in full format */
for(i = 0; i < nrow; i++)
y[i] = 0.0;
y[irow] = 1.0;
rc = XPRsftran(prob,y);
```

補足

行列が事前分析済みの場合、この関数は事前分析済みの問題の基底として動作します。

関連事項

XPRSbtran.

Xpress-Optimizer Reference Manual Console and Library Functions 88

XPRSgetbanner 6**XPRSgetbanner****目的**

バナーと著作権メッセージを返します。

概要

```
int XPRSgetbanner(char *banner);
```

引数**関係する制御**

なし

例

プログラムの始めにバナー情報を返すためにXPRSgetbannerを呼びだします。

```
char banner[256];
...
if(XPRSinit(NULL))
{
XPRSgetbanner(banner);
printf("%s\n",banner);
return 1;
}
XPRSgetbanner(banner);
printf("%s\n",banner);
```

補足

この関数は、XPRSinitについて問題が起こる場合、更なる情報を返すのに有効です。

関連事項

XPRSinit.

banner バナーを保持するのに十分な長さのバッファ (NULL終了部分を追加)。
 最大256文字

Xpress-Optimizer Reference Manual Console and Library Functions 89

XPRSgetbasis 6**XPRSgetbasis****目的**

ユーザーのデータエリア内に現在の基底を返します。

概要

```
int XPRSgetbasis(XPRSprob prob, int *rstatus,
int *cstatus);
```

引数**関係する制御**

Integer

例

後のために基底を保存する前に問題を最小化します。

```

int rows, cols, *rstatus, *cstatus;
...
XPRSminim(prob, "");
XPRSgetintattrib(prob, XPRS_ROWS, &rows);
XPRSgetintattrib(prob, XPRS_COLS, &cols);
rstatus = (int *) malloc(sizeof(int)*rows);
cstatus = (int *) malloc(sizeof(int)*cols);
prob          現在の問題
rstatus       各行に関するスラック, サープラス, 人為変数の基底状態を表す
              長さROWSの整数型配列。状態は次のいずれかです
              0   スラック, サープラス, 人為変数が下限において非基底
              1   スラック, サープラス, 人為変数が基底
              2   スラック, サープラス, 人為変数が上限において非基底
              必要なければNULLとなります
cstatus       制約行列における列の基底状態を保持する長さCOLSの整数型配列。
              状態はいずれかです
              0   変数は下限において非基底。または変数に下限がない場合に零
                  で基底
              1   変数は基底
              2   変数は上限で基底
              必要なければNULLとなります
SOLUTIONFILE 解が読み書きされる制御
Xpress-Optimizer Reference Manual Console and Library Functions 90

```

XPRSgetbasis 6

```
XPRSgetbasis(prob, rstatus, cstatus);
```

関連事項

XPRSgetpresolvebasis, XPRSloadbasis, XPRSloadpresolvebasis.
Xpress-Optimizer Reference Manual Console and Library Functions 91

XPRSgetcolrange 6

XPRSgetcolrange

目的

XPRSrangeによって計算された列範囲を返します。

概要

```
int XPRSgetcolrange(XPRSprob prob, double *upact,
double *loact, double *uup, double *udn, double *ucost,
double *lcost);
```

引数

関係する制御

なし

例

概要に示すように、列範囲が配列内に戻されます。

```
int cols, *upact, *loact, *uup, *udn, *ucost, *lcost;
...
XPRSgetintattrib(prob, XPRS_COLS, &cols);
upact = malloc(cols*(sizeof(int)));
loact = malloc(cols*(sizeof(int)));
```

```

uup = malloc(cols*(sizeof(int)));
udn = malloc(cols*(sizeof(int)));
ucost = malloc(cols*(sizeof(int)));
lcost = malloc(cols*(sizeof(int)));
XPRStrange(prob);
XPRSgetcolrange(prob, upact, loact, uup, udn, ucost, lcost);

```

prob	現在の問題
upact	上側列アクティビティの長さCOLSの倍精度型配列
loact	下側列アクティビティの長さCOLSの倍精度型配列
uup	上側列ユニットコストの長さCOLSの倍精度型配列
udn	下側列ユニットコストの長さCOLSの倍精度型配列
ucost	上側コストの長さCOLSの倍精度型配列
lcost	下側コストの長さCOLSの倍精度型配列

Xpress-Optimizer Reference Manual Console and Library Functions 92

XPRSgetcolrange 6

補足

アクティビティとユニットコストはレンジファイル(*problem_name.rng*)から得られます。レンジファイルにおける上下側列アクティビティと上下側ユニットコストは付録A「ログとファイル形式」に説明があります。

関連事項

XPRSgetrowrange, XPRStrange.

Xpress-Optimizer Reference Manual Console and Library Functions 93

XPRSgetcols 6

XPRSgetcols

目的

与えられた範囲で列の制約行列中の非零要素数を返します。

概要

```

int XPRSgetcols(XPRSprob prob, int *mstart, int *mrwind,
double *dmatval, int size, int *nels, int first,
int last);

```

引数

関係する制御

なし

例

```

int nels, cols, first = 0, last;

```

prob	現在の問題
mstart	各要求列についてのmrwindとdmatval配列の開始オフセットを示す添字が入れられる整数型配列。長さは少なくともlast-first+2です。列iは、mrwindとdmatval配列において位置mstart[i]で始まり、mstart[i+1]-mstart[i]要素含まれます。必要なければNULLとなります
mrwind	各列の非零要素の行添字が入れられる長さsizeの整数型配列。必要なければNULLとなります
dmatval	非零要素値が入れられる長さsizeの倍精度型配列。必要なければNULLとなります

size 戻される要素の最大数
 nels mrwind,dmatval 配列内の非零要素の数が返される整数のポインタ。
 非零要素の数がsizeより大きい場合は、size要素だけ返されます。
 nelsがsizeより小さい場合はnelsだけ返されます
 first その範囲での最初の列
 last その範囲での最後の列

Xpress-Optimizer Reference Manual Console and Library Functions 94

XPRSgetcols 6

```
...
XPRSgetintattrib(prob,XPRS_COLS,&cols);
last = cols-1;
XPRSgetcols(prob,NULL,NULL,NULL,0,&nels,first,last);
```

行列のすべての列における非零行列要素の数をnelsに返します。

補足

例に示すように、mstart,mrwind,dmatvalをNULLに置換することで、その列範囲での要素の数を得られます。この場合、渡された配列の長さが零であることを示すためにsizeには0が設定されなくてはなりません。これは上記の例に示されている通りです。

関連事項

XPRSgetrows.

Xpress-Optimizer Reference Manual Console and Library Functions 95

XPRSgetcoltype 6

XPRSgetcoltype

目的

与えられた範囲での列の列タイプを返します。

概要

```
int XPRSgetcoltype(XPRSprob prob, char *coltype, int first,
int last);
```

引数

関係する制御

なし

例

行列内の全ての列についてタイプを調べ、コンソールに出力します。

```
int cols, i;
char *types;
...
XPRSgetintattrib(prob,XPRS_COLS,&cols);
types = (char *)malloc(sizeof(char)*cols);
XPRSgetcoltype(prob,types,0,cols-1);
for(i=0;i<cols;i++) printf("%c\n",types[i]);
```

関連事項

XPRSchgcoltype, XPRSgetrowtype.

prob	現在の問題
coltype	列タイプが返される長さlast-first+1の文字配列
	C 連続変数
	I 整数変数

	B	バイナリ変数
	S	半連続変数
	R	半連続整数変数
	P	部分整数変数
first		その範囲での最初の列
last		その範囲での最後の列

Xpress-Optimizer Reference Manual Console and Library Functions 96

XPRSgetpcutlist 6

XPRSgetpcutlist

目的

カットプールからカット添字のリストを返します。

概要

```
int XPRSgetpcutlist(XPRSprob prob, int itype, int interp,
double delta, int *ncuts, int size, int *mcutind,
double *dviol);
```

引数

関係する制御

なし

補足

1. deltaパラメータに必要な違反サイズを設定することによって違反カットが得られます。同様に、違反していないカットが必要であれば、ライブラリヘッダファイルに定義された XPRS_MINUSINFINITYをdeltaに設定します。

prob	現在の問題
itype	返されるカットのカットタイプ
interp	カットタイプの解釈
	-1 全てのカットを得ます
	1 カットタイプを数で扱います
	2 カットタイプをビットマップで扱います。ビットのいずれかが itypeに設定されたビットと一致する場合、カットを得ます
	3 カットタイプをビットマップで扱います。全てのビットが itypeに設定されたビットと一致する場合、カットを得ます
delta	deltaよりも大きい絶対スラック値のカットだけが返されます
ncuts	カットプール内のタイプitypeのカットの数で返される整数のポインタ
size	返されるカットの最大数
mcutind	カットの添字番号が返される長さsizeの整数型配列
dviol	カットのスラック変数値が返される長さsizeの倍精度型配列

Xpress-Optimizer Reference Manual Console and Library Functions 97

XPRSgetpcutlist 6

2. アクティブカットの数がsizeより多い場合、sizeカットだけが返され、ncutsはアクティブカットの数に設定されます。ncutsがsizeよりも小さい場合は、ncutsの位置にmcutindが入るだけです。

関連事項

XPRSgetpccuts, XPRSgetcutlist, 5.4章「カットマネージャのはたらき」
Xpress-Optimizer Reference Manual Console and Library Functions 98

XPRSgetpccuts 6**XPRSgetpccuts****目的**

カットプールからカットのリストが返されます。配列mindex内のカットのリストはルーチンに渡されなくてはなりません。カットの列と要素は、mcols, dmatvalパラメータにより示された領域に返されます。列と要素は続けて保存され、各カットの開始点はmstartパラメータで示された領域に返されます。

概要

```
int XPRSgetpccuts(XPRSprob prob, int *mindex, int ncuts,
int size, int *mtype, char *qrtype, int *mstart,
int *mcols, double *dmatval, double *drhs);
```

引数**関係する制御**

なし

prob	現在の問題
mindex	カットの添字番号を含む長さncutsの整数型配列
ncuts	返されるカットの数
size	返されるカットの列添字の最大数
mtype	カットタイプが返される長さが少なくともncutsの整数型配列
qrtype	カットの意味(L, G, E)が返される長さが少なくともncutsの文字配列
mstart	mcols, dmatval配列へのオフセットを含む長さが少なくともncuts+1の整数型配列。最後の要素は、カットncuts+1の開始位置を示します
mcols	カットの列添字が返される長さsizeの整数型配列
dmatval	行列値が返される長さsizeの倍精度型配列
drhs	カットの右辺要素が返される長さが少なくともncutsの倍精度型配列

Xpress-Optimizer Reference Manual Console and Library Functions 99

XPRSgetpccuts 6**関連事項**

XPRSgetpcutlist, XPRSgetcutlist, 5.4章「カットマネージャのはたらき」
Xpress-Optimizer Reference Manual Console and Library Functions 100

XPRSgetcutlist 6**XPRSgetcutlist****目的**

現在のノードでのアクティブカットの添字のリストを戻します。

概要

```
int XPRSgetcutlist(XPRSprob prob, int itype, int interp,
int *ncuts, int size, int *mcutind);
```

引数**関係する制御**

なし

補足

アクティブカット数がsizeよりも大きい場合は、sizeカットが返され、ncutsはアクティブカット数に設定されます。ncutsがsizeよりも小さい場合は、ncutsの位置にはmcutindが入ります。XPRSaddcutsを使って行列にカットが加えられ、カットプールに保存されていなければ、-1の添字が返されます。

関連事項

XPRSgetcpcutlist, XPRSgetcpcuts, 5.4章「カットマネージャのはたらき」

prob	現在の問題
itype	返されるカットのカットタイプ。-1は全てのアクティブカットを返すことを示します
interp	カットタイプの解釈 -1 全てのカットを得ます 1 カットタイプを数字で扱います 2 カットタイプをビットマップで扱います。ビットのいずれかがitypeに設定されるビットと一致する場合にカットを得ます 3 カットタイプをビットマップで扱います。全てのビットがitypeに設定されるビットと一致する場合にカットを得ます
ncuts	タイプitypeのアクティブカットの数が返される整数のポインタ
size	戻されるカットの最大数
mcutind	カットの添字番号が返される長さsizeの整数型配列

Xpress-Optimizer Reference Manual Console and Library Functions 101

XPRSgetdblattrib 6**XPRSgetdblattrib****目的**

様々な倍精度型の問題の特性値を戻します。問題特性は、問題のロード中や最適化中に設定されます。

概要

```
int XPRSgetdblattrib(XPRSprob prob, int ipar,
double *dval);
```

引数**関係する制御**

なし

例

目的関数の最適値を得てコンソールに出力します。

```
double lpobjval;
...
XPRSmxim(prob, "");
XPRSgetdblattrib(prob, XPRS_LPOBJVAL, &lpobjval);
printf("The maximum profit is %f\n", lpobjval);
```

関連事項

XPRSgetintattrib, XPRSgetstrattrib.

prob	現在の問題
ipar	値が返される問題特性。全ての問題特性のリストは8章「問題特性」あるいは、 <code>xprs.h</code> ヘッダファイル中のリストを参照ください
dval	返される問題特性の値の倍精度型のポインタ

Xpress-Optimizer Reference Manual Console and Library Functions 102

XPRSgetdblcontrol 6

XPRSgetdblcontrol

目的

与えられた倍精度型の制御パラメータの値を戻します。

概要

```
int XPRSgetdblcontrol(XPRSprob prob, int ipar,
double *dval);
```

引数

関係する制御

7章「制御パラメータ」参照

例

整数の実行可能許容範囲を戻します。

```
XPRSgetdblcontrol(prob, XPRS_MIPTOL, &miptol);
```

関連事項

XPRSsetdblcontrol, XPRSgetintcontrol, XPRSgetstrcontrol.

prob	現在の問題
ipar	値が返される制御パラメータ。全制御パラメータのリストは7章「制御パラメータ」か <code>xprs.h</code> ヘッダファイル中のリストを参照してください
dval	返される制御値の位値を示すポインタ

Xpress-Optimizer Reference Manual Console and Library Functions 103

XPRSgetdirs 6

XPRSgetdirs

目的

行列にロードされた指示を戻します。優先度や分枝方向, 擬コストが返されます。事前分析後に呼びだされた場合、XPRSgetdirsは事前分析済みの問題のための指示を得ます。

概要

```
int XPRSgetdirs(XPRSprob prob, int *ndir, int *mcols,
int *mpri, char *qbr, double *dupc, double *ddpc);
```

引数

関係する制御

なし

補足

1. 値`ndir`は最大でMIPENTSの指示の数を示し、`XPRSgetintattrib(prob, XPRS_MIPENTS, &mipents)`で得られます。
2. `prob`と`ndir`以外のいかなる引数については、必要なければNULLとなります。

prob	現在の問題
ndir	指示の数が返される整数のポインタ

mcols	列番号 (0,1, 2,...) が (1つ目が-1, 2つ目が-2,...)の特殊順序集合に一致する負の値を含む長さ <code>ndir</code> の整数型配列
mpri	列, 集合の優先順位を含む長さ <code>ndir</code> の整数型配列
qbr	各列や集合の分枝方向を指定する長さ <code>ndir</code> の文字配列
	U 押し上げ要素
	D 押し下げ要素
	N 指定なし
dupc	列や集合の上側擬コストを含む長さ <code>ndir</code> の倍精度型配列
ddpc	列や集合の下側擬コストを含む長さ <code>ndir</code> の倍精度型配列
Xpress-Optimizer Reference Manual Console and Library Functions 104	

XPRSgetdirs 6

関連事項

XPRsloaddirs, XPRsloadpresolvedirs.
Xpress-Optimizer Reference Manual Console and Library Functions 105

XPRSgetglobal 6

XPRSgetglobal

目的

問題についてのグローバル情報を戻します。事前分析オプションが使用される場合は、XPRsminim, XPRsmaximの前に呼びだしてください。

概要

```
int XPRSgetglobal(XPRsprob prob, int *nglents, int *sets,
char *qgtype, int *mgcols, double *dlim, char *qstype,
int *msstart, int *mscols, double *dref);
```

引数

prob	現在の問題
nglents	バイナリ, 整数, 半連続, 半連続整数, 部分整数の要素の数が返される整数のポインタ。これは問題特性MIPENTSと同じです
sets	SOS1, SOS2集合の数が返される整数のポインタ。問題特性SETSから得ることが可能です
qgtype	要素タイプが返される長さ <code>nglents</code> の文字配列。タイプは次のどれかです
	B バイナリ変数
	I 整数変数
	P 部分整数変数
	S 半連続変数
	R 半連続整数変数
mgcols	離散要素の列添字が返される長さ <code>nglents</code> の整数型配列
dlim	部分整数変数の限度や半連続変数, 半連続整数変数の下限が返される長さ <code>nglents</code> の倍精度型配列 (バイナリ変数や整数変数に相当する位値の要素は意味をなしません)
qstype	集合タイプが返される長さ <code>sets</code> の文字配列。集合タイプは次のどれかです
	1 SOS1 タイプ集合
	2 SOS2 タイプ集合

XPRSgetglobal 6**関係する制御**

なし

例

配列mgcols と qrtype内の離散変数とそのタイプを得ます。

```
int nglents, nsets, *mgcols;
char *qrtype;
...
XPRSgetglobal(prob, &nglents, &nsets, NULL, NULL, NULL, NULL,
NULL, NULL, NULL);
mgcols = malloc(nglents*sizeof(int));
qrtype = malloc(nglents*sizeof(char));
XPRSgetglobal(prob, &nglents, &nsets, qrtype, mgcols, NULL,
NULL, NULL, NULL, NULL);
```

補足

必要なければ、prob, nglents, nsetsを除き、引数はNULLとなります。

関連事項

XPRSloadglobal, XPRSloadqglobal.

msstart	集合の開始を示すmscols, dref配列へのオフセットが返される整数型配列。これは最後の要素が集合 sets+1で始まり、mscols, dref配列、SETMEMBERSの長さと同じオフセットを含む長さsets+1の配列でなくてはなりません
mscols	各集合の列が返される長さSETMEMBERSの整数型配列
dref	集合の各要素の参照行要素が返される長さSETMEMBERSの倍精度型配列

XPRSgetiis 6**XPRSgetiis****目的**

IIS探索において見つかった既約実行不可能集合(IIS)の1つを検索します。

概要

```
int XPRSgetiis(XPRSprob prob, int *colnumber,
int *rownumber, int *miiscol, int *miisrow);
```

引数**関係する制御***Integer***例**

問題probにおいてIISを発見し、検索します。

```
int ncols, nrows, *miiscol, *miisrow;
...
XPRSgetiis(prob, "");
XPRSgetiis(prob, &ncols, &nrows, NULL, NULL);
miiscol = malloc(ncols*sizeof(int));
miisrow = malloc(nrows*sizeof(int));
XPRSgetiis(prob, &ncols, &nrows, miiscol, miisrow);
```

補足

関数 `XPRiis` の後でのみ `XPRSgetiis` を呼ぶことができ、問題中の IIS を計算します。

関連事項

`XPRSiis`.

<code>prob</code>	現在の問題
<code>colnumber</code>	IIS の列の数
<code>rownumber</code>	IIS の行の数
<code>miiscol</code>	IIS 集合の列添字を含む長さ <code>colnumber</code> の整数型配列。必要なければ NULL となります
<code>miisrow</code>	IIS 集合の行添字を含む長さ <code>rownumber</code> の整数型配列。必要なければ NULL となります
<code>MAXIIS</code>	発見される既約実行不可能集合の数

Xpress-Optimizer Reference Manual Console and Library Functions 108

XPRSgetindex 6**XPRSgetindex****目的**

指定した行、列名の添字を返します。

概要

```
int XPRSgetindex(XPRSprob prob, int type, char *name,
int *seq);
```

引数**関係する制御**

Integer

例

`problem` をロードし、「n 0203」が行もしくは列の名前かを調べます。

```
int seqr, seqc;
...
XPRSreadprob(prob, "problem", "");
XPRSgetindex(prob, 1, "n 0203", &seqr);
XPRSgetindex(prob, 2, "n 0203", &seqc);
if (seqr == -1 && seqc == -1) printf("n 0203 not there\n");
if (seqr != -1) printf("n 0203 is row %d\n", seqr);
if (seqc != -1) printf("n 0203 is column %d\n", seqc);
```

関連事項

`XPRSaddnames`.

<code>prob</code>	現在の問題
<code>type</code>	1 行の添字が要求される時 2 列の添字が要求される時
<code>name</code>	行名もしくは列名を持つ長さ <code>MPSNAMELENGTH</code> (NULL 終了分を加える) の文字列
<code>seq</code>	行または列の添字番号が返される整数のポインタ。その行、列が存在しない場合は、-1 が返されます
<code>MPSNAMELENGTH</code>	最大名前文字長さ

Xpress-Optimizer Reference Manual Console and Library Functions 109

XPRSgetinfeas 6

XPRSgetinfeas

目的

実行不可能な主、双対変数のリストを返します。

概要

```
int XPRSgetinfeas(XPRSprob prob, int *npv, int *nps,
int *nds, int *ndv, int *mx, int *mslack, int *mdual,
int *mdj);
```

引数

エラー値

関係する制御

Integer

Double

prob	現在の問題
npv	主実行不能変数の数
nps	主実行不能行の数
nds	双対実行不能行の数
ndv	双対実行不能変数の数
mx	主実行不能変数が返される長さnpvの整数型配列。必要なければNULLとなります
mslack	主実行不能行が返される長さnpsの整数型配列。必要なければNULLとなります
mdual	双対実行不能行が返される長さndsの整数型配列。必要なければNULLとなります
mdj	双対実行不能変数が返される長さndvの整数型配列。必要なければNULLとなります
91	現在の問題は有効ではありません
422	解は有効ではありません
SOLUTIONFILE	解が読み書きされる制御
FEASTOL	RHSの零許容範囲
OPTIMALITYTOL	被約費用許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 110

XPRSgetinfeas 6

例

実行不能要素の数を得るために、最初にNULLの整数型配列を伴ってXPRSgetinfeasが呼びだされます。そして、その配列のためにスペースが割り当てられ、その関数が再度それらを入れるために呼びだされます。

```
int npv, nps, nds, ndv, *mx, *mslack, *mdual, *mdj;
...
XPRSgetinfeas(prob, &npv, &nps, &nds, &ndv, NULL, NULL,
NULL, NULL);
mx = malloc(npv * sizeof(*mx));
mslack = malloc(nps * sizeof(*mslack));
mdual = malloc(nds * sizeof(*mdual));
mdj = malloc(ndv * sizeof(*mdj));
XPRSgetinfeas(prob, &npv, &nps, &nds, &ndv, mx, mslack,
mdual, mdj);
```


補足

1. SOLUTIONFILEに0が設定されると、メモリに解の実行不能性が返されます。SOLUTIONFILEに1が設定されると、.solファイルに解の実行不能性が返されます。
2. 最後の4つの引数のいずれかにNULLが設定されると、実行不能の番号に相当する値が返されます。

関連事項

XPRSgetiis, XPRSgetscaledinfeas, XPRSiis.
Xpress-Optimizer Reference Manual Console and Library Functions 111

XPRSgetintattrib 6**XPRSgetintattrib****目的**

様々な整数問題特性の値を戻します。問題のロード中、最適化中に問題特性が設定されます。

概要

```
int XPRSgetintattrib(XPRSprob prob, int ipar, int *ival);
```

引数**関係する制御**

なし

例

行列内の列の数を得て、各列の下限を得るためのスペースを割り当てます。

```
int cols;
double *lb;
...
XPRSgetintattrib(prob, XPRS_COLS, &cols);
lb = (double *) malloc(sizeof(double)*cols);
XPRSgetlb(prob, lb, 0, cols-1);
```

関連事項

XPRSgetdblattrib, XPRSgetstrattrib.

prob	現在の問題
ipar	値が戻される問題特性。全問題特性のリストは8章「問題特性」か xprs.hヘッダファイルを参照ください
ival	問題特性の値が返される整数のポインタ

Xpress-Optimizer Reference Manual Console and Library Functions 112

XPRSgetintcontrol 6**XPRSgetintcontrol****目的**

様々な整数制御パラメータの値を戻します。

概要

```
int XPRSgetintcontrol(XPRSprob prob, int ipar, int *igval);
```

引数**関係する制御**

7章「制御パラメータ」参照

例

defaultalgの値を得て画面に表示します。

...

```
XPRsmaxim(prob, "");
XPRsgetintcontrol(prob, XPRS_DEFAULTALG, &defaultalg);
printf("DEFAULTALG is %d\n", defaultalg);
```

補足

SCALINGのような制御パラメータはビットマップです。各ビットのはたらきは異なります。例えばビット0は値1，ビット1は値2，ビット2は値4などです。

関連事項

XPRssetintcontrol, XPRsgetdblcontrol, XPRsgetstrcontrol.

prob	現在の問題
ipar	値が返される制御パラメータ。全制御のリストは7章「制御パラメータ」がxprs.hヘッダファイルを参照ください
igval	制御の値が返される整数のポインタ

Xpress-Optimizer Reference Manual Console and Library Functions 113

XPRsgetlasterror 6**XPRsgetlasterror****目的**

最適化計算中、最後に発生したエラーを返します。

概要

```
int XPRsgetlasterror(XPRsprob prob, char *errmsg);
```

引数**関係する制御**

なし

例

エラーチェックにおけるこの関数の使われ方を示します。

```
void error(XPRsprob myprob, char *function)
{
    char errmsg[256];
    XPRsgetlasterror(myprob, errmsg);
    printf("Function %s did not execute correctly: %s\n",
        function, errmsg);
    XPRsdestroyprob(myprob);
    XPRsfree();
    exit(1);
}
```

ここで、主関数は以下のような行を含みます。

```
XPRsprob prob;
...
if(XPRscreateprob(&prob)) error(prob, "XPRscreateprob");
```

関連事項

9章「エラーメッセージ&リターンコード」, ERRORCODE, XPRssetcbmessage, XPRssetlogfile.

prob	現在の問題
errmsg	最後のエラーメッセージが返される256文字分のバッファ

Xpress-Optimizer Reference Manual Console and Library Functions 114

XPRsgetlb 6**XPRsgetlb**

目的

与えられた範囲における列の下限を返します。

概要

```
int XPRSgetlb(XPRSprob prob, double *lb, int first,
int last);
```

引数**関係する制御**

なし

例

現在の問題の列の下限を戻します。

```
int cols;
double *lb;
...
XPRSgetintattrib(prob, XPRS_COLS, &cols);
lb = (double *) malloc(sizeof(double)*cols);
XPRSgetlb(prob, lb, 0, cols-1);
```

補足

XPRS_PLUSINFINITY以上の値の場合は、無限大となります。XPRS_MINUSINFINITY以下の値の場合はマイナスに無限小となります。

関連事項

XPRSchgbounds, XPRSgetub.

prob	現在の問題
lb	下限が置かれる長さlast-first+1の倍精度型配列
first	その範囲での最初の列
last	その範囲での最後の列

Xpress-Optimizer Reference Manual Console and Library Functions 115

XPRSgetnames 6**XPRSgetnames****目的**

与えられた範囲における行や列の名前を返します。その名前は文字バッファに返され、各名前はNULL文字で区切られます。

概要

```
int XPRSgetnames(XPRSprob prob, int type, char *names,
int first, int last);
```

引数**関係する制御**

Integer

例 1

現在の問題の行と列の名前を戻します。

```
int cols, rows, nl;
...
XPRSgetintattrib(prob, XPRS_COLS, &cols);
XPRSgetintattrib(prob, XPRS_ROWS, &rows);
XPRSgetintcontrol(prob, XPRS_MPSNAMELENGTH, &nl);
cnames = (char *) malloc(sizeof(char)*(nl+1)*cols);
```

prob	現在の問題
------	-------

type	1 行の名前が要求された場合 2 列の名前が要求された場合
names	名前を保存するのに十分な長さのバッファ。それぞれの名前は MPSNAMELENGTH の文字長さ (NULL 文字終了分を加える) なので、配列 names は少なくとも $(\text{first} - \text{last} + 1) * (\text{MPSNAMELENGTH} + 1)$ 文字分必要となります。行/列 first+i の名前は、位値 $i * \text{MPSNAMELENGTH} + i$ で始まるバッファ names に書きこまれます
first	その範囲での最初の行や列
last	その範囲での最後の行や列
MPSNAMELENGTH	最大名前文字長さ

Xpress-Optimizer Reference Manual Console and Library Functions 116

XPRSgetnames 6

```
rnames = (char *) malloc(sizeof(char)*(nl+1)*rows);
XPRSgetnames(prob, 1, rnames, 0, rows-1);
XPRSgetnames(prob, 2, cnames, 0, cols-1);
```

例 2

C で names[i] を表示するには以下のようにします。

```
int namelength;
...
XPRSgetintcontrol(prob, XPRS_MPSNAMELENGTH, &namelength);
printf("%s", names + i*(namelength+1));
```

関連事項

XPRSaddnames.
Xpress-Optimizer Reference Manual Console and Library Functions 117

XPRSgetobj 6

XPRSgetobj

目的

与えられた範囲において、列の目的関数係数を返します。

概要

```
int XPRSgetobj(XPRSprob prob, double *obj, int first,
int last);
```

引数

関係する制御

なし

例

現在の問題の目的関数係数を返します。

```
int cols;
double *obj;
...
XPRSgetintattrib(prob, XPRS_COLS, &cols);
obj = (double *) malloc(sizeof(double)*cols);
XPRSgetobj(prob, obj, 0, cols-1);
```

関連事項

XPRSchgobj.
prob 現在の問題

obj 目的関数係数が置かれる長さlast-first+1の倍精度型配列
 first その範囲での最初の列
 last その範囲での最後の列
 Xpress-Optimizer Reference Manual Console and Library Functions 118

XPRSgetpivotorder 6

XPRSgetpivotorder

目的

基底変数のピボット順序を返します。

概要

```
int XPRSgetpivotorder(XPRSprob prob, int *mpiv);
```

引数

関係する制御

なし

例

変数のピボット順序を配列pPivotに返します。

```
XPRSgetintattrib(prob, XPRS_ROWS, &rows);
pPivot = malloc(rows*(sizeof(int)));
XPRSgetpivotorder(prob, pPivot);
```

補足

行添字は範囲0 ~ ROWS-1です。列の範囲はROWS+SPAREROWS ~ ROWS+SPAREROWS+COLS-1です。

関連事項

XPRSgetpivots, XPRSpivot.

prob 現在の問題
 mpiv ピボット順序が返される長さROWSの整数型配列
 Xpress-Optimizer Reference Manual Console and Library Functions 119

XPRSgetpivots 6

XPRSgetpivots

目的

変数inが基底に入る場合、基底を出る可能性のある変数のリストを返します。

概要

```
int XPRSgetpivots(XPRSprob prob, int in, int *outlist,
double *x, double *dobj, int *npiv, int maxpiv);
```

引数

エラー値

関係する制御

なし

例

7番目の変数が基底に入る場合に基底を出る可能性のある変数のリストを探索します。

```
int npiv, outlist[5];
double dobj;
...
XPRSgetpivots(prob, 6, outlist, NULL, &dobj, &npiv, 5);
```

prob 現在の問題
 in 基底に入る行や列の添字

<code>outlist</code>	出る可能性のある変数のリストを保持する長さ <code>maxpiv</code> 以上の整数型配列。必要なければ <code>NULL</code> となります
<code>x</code>	<code>in</code> が基底に入れられる場合、全変数の値を保持する長さ <code>ROWS+SPAREROWS+COLS</code> の倍精度型配列。必要なければ <code>NULL</code> となります
<code>dobj</code>	<code>in</code> が基底に入れられる場合、目的関数値が返される倍精度型のポインタ
<code>npiv</code>	出る可能性のある変数の実際の数に返される整数ポインタ
<code>maxpiv</code>	出る可能性のある変数が返される数の最大数
425	<code>in</code> が無効であることを示します（範囲外かすでに基底のとき）

Xpress-Optimizer Reference Manual Console and Library Functions 120

XPRSgetpivots 6

補足

- 変数`in`が基底に入り問題が退化する場合、いくつかの基底変数は基底を離れる候補となり、その可能性のある候補の数は`npiv`に返されます。これらの候補は多くても `maxpiv`で、少なくとも長さ`maxpiv`はある`outlist`にリストが返されます。変数`in`がピボット化された場合は、問題が退化するため、目的関数値や全ての変数は、`outlist`からの候補で基底を去るために選ばれたものには依存しません。目的関数の値は`dobj`に返され、変数の値は`x`に返されます。
- 行添字は範囲`0 ~ ROWS-1`、列添字は範囲`ROWS+SPAREROWS ~ ROWS+SPAREROWS+COLS-1`です。

関連事項

`XPRSgetpivotorder`, `XPRSpivot`.

Xpress-Optimizer Reference Manual Console and Library Functions 121

XPRSgetpresolvebasis 6

XPRSgetpresolvebasis

目的

メモリからユーザーのデータエリアに現在の基底を返します。問題が事前分析されていると、事前分析された基底が返されます。そうでなければ、原基底が返されます。

概要

```
int XPRSgetpresolvebasis(XPRSprob prob, int *rstatus,
int *cstatus);
```

引数

関係する制御

なし

例

整数解探索に先だって事前分析された問題の基底情報を得て出力します。

```
XPRSprob prob;
```

```
int i, cols, *cstatus;
```

```
...
```

```
XPRSreadprob(prob, "myglobalprob", "");
```

```
XPRSmnim(prob, "");
```

```
prob
```

現在の問題

```
rstatus
```

各行に関するスラック，サープラス，人為変数の基底状態を示す

長さROWSの整数型配列。状態は次のうちのどれか一つです

- 0 スラック, サープラス, または人為変数が下限で非基底状態
- 1 スラック, サープラス, または人為変数が基底状態
- 2 スラック, サープラスが上限で非基底状態

必要なければNULLとなります

cstatus 制約行列における列の基底状態を保持するための長さCOLSの整数型配列。状態は次のいずれかです

- 0 変数が下限で非基底状態か、下限がなければ零で基底状態
- 1 変数が基底状態
- 2 変数が上限にある

必要なければNULLとなります

Xpress-Optimizer Reference Manual Console and Library Functions 122

XPRSgetpresolvebasis 6

```
XPRSgetintattrib(prob, XPRS_COLS, &cols);
cstatus = malloc(cols * sizeof(int));
XPRSgetpresolvebasis(prob, NULL, cstatus);
for(i=0; i<cols; i++)
printf("Column %d: %d\n", i, cstatus[i]);
XPRSglobal(prob);
```

関連事項

XPRSgetbasis, XPRSloadbasis, XPRSloadpresolvebasis.
Xpress-Optimizer Reference Manual Console and Library Functions 123

XPRSgetpresolvesol 6

XPRSgetpresolvesol

目的

メモリから事前分析された問題の解を返します。

概要

```
int XPRSgetpresolvesol(XPRSprob prob, double *x,
double *slack, double *dual, double *dj);
```

引数

関係する制御

なし

例

(混合)整数計画問題を読みこみ、線形緩和問題の最適化に続いて事前分析された問題の解を表示します。

```
XPRSreadprob(prob, "MyProb", "");
XPRSmaxim(prob, "");
XPRSgetintattrib(prob, XPRS_COLS, &cols);
x = malloc(cols * sizeof(double));
XPRSgetpresolvesol(prob, x, NULL, NULL, NULL);
for(i=0; i<cols; i++)
printf("Presolved x(%d) = %g\n", i, x[i]);
XPRSglobal(prob);
```

prob 現在の問題

x 主変数の値が返される長さCOLSの倍精度型配列。必要なければNULLとなります

slack	スラック値が返される長さROWSの倍精度型配列。必要なければNULLとなります
dual	双対変数の値が返される長さROWSの倍精度型配列。必要なければNULLとなります
dj	各変数の被約費用が返される長さCOLSの倍精度型配列。必要なければNULLとなります

Xpress-Optimizer Reference Manual Console and Library Functions 124

XPRSgetpresolvesol 6

補足

1. 問題が事前分析されていなければ、メモリにある解が返されます。
2. 原問題の解は関連する関数XPRSgetsolにより返されます。

関連事項

XPRSgetsol, 章5.2 「事前分析のはたらき」

Xpress-Optimizer Reference Manual Console and Library Functions 125

XPRSgetprobname 6

XPRSgetprobname

目的

現在の問題名を返します。

概要

```
int XPRSgetprobname(XPRSprob prob, char *probname);
```

引数

関係する制御

なし

例

The following returns the problem name into probname:

```
char probname[200];
...
XPRSgetprobname(prob, probname);
```

関連事項

XPRSsetprobname.

prob

現在の問題

probname

現在の問題名を含むための200文字以内の文字列.

Xpress-Optimizer Reference Manual Console and Library Functions 126

XPRSgetqobj 6

XPRSgetqobj

目的

Hessian 行列の変数の組(icol, jcol)に相当する二次目的関数係数の1つを返します。

概要

```
int XPRSgetqobj(XPRSprob prob, int icol, int jcol,
double *dval);
```

引数

関係する制御

なし

例

目的関数の項の係数を返し、変数valueに置きます。

```
double value;
...
XPRSgetqobj(prob, 0, 0, &value);
```

補足

dval は二次Hessian 行列の係数です。例えば、目的関数が項を持つ場合、XPRSgetqobj によって戻された値は3.0で、また目的関数が項を持つ場合、XPRSgetqobjによって戻された値はの6.0です。

関連事項

XPRSchgqobj, XPRSchgmqobj.

prob	現在の問題
icol	二次の項における第 1 変数の列添字
jcol	二次の項における第 2 の変数の列添字
dval	目的関数の係数がおかれる倍精度型ポインタ

x_0^2

$3x_1 x_2 + 3x_2 x_1 + x_2^2$

$6x_1^2 x_2^2$

Xpress-Optimizer Reference Manual Console and Library Functions 127

XPRSgetrhs 6**XPRSgetrhs****目的**

与えられた範囲にある行の右辺要素を返します。

概要

```
int XPRSgetrhs(XPRSprob prob, double *rhs, int first,
int last);
```

引数**関係する制御**

なし

例

問題の右辺値を返します。

```
int rows;
double *rhs;
...
XPRSgetintattrib(prob, XPRS_ROWS, &rows);
rhs = (double *) malloc(sizeof(double)*rows);
XPRSgetrhs(prob, rhs, 0, rows-1);
```

関連事項

XPRSchgrhs, XPRSchgrhsrange, XPRSgetrhsrange.

prob	現在の問題
rhs	右辺要素が置かれる長さlast-first+1 の倍精度型配列
first	その範囲での最初の行
last	その範囲での最後の行

Xpress-Optimizer Reference Manual Console and Library Functions 128

XPRSgetrhsrange 6

XPRSgetrhrange

目的

与えられた範囲にある行の右辺範囲値を返します。

概要

```
int XPRSgetrhrange(XPRSprob prob, double *range,
int first, int last);
```

引数

関係する制御

なし

例

行列内の全ての行の右辺範囲値を返します。

```
int rows;
double *range;
...
XPRSgetintattrib(prob, XPRS_ROWS, &rows);
range = malloc(rows*sizeof(double));
XPRSgetrhrange(prob, range, 0, rows);
```

関連事項

XPRSchgrhs, XPRSchgrhrange, XPRSgetrhs, XPRSrange.

prob	現在の問題
range	右辺範囲値がおかれる長さlast-first+1 の倍精度型配列
first	その範囲での最初の行
last	その範囲での最後の行

Xpress-Optimizer Reference Manual Console and Library Functions 129

XPRSgetowrange 6

XPRSgetowrange

目的

XPRSrange によって計算された行範囲を返します。

概要

```
int XPRSgetowrange(XPRSprob prob, double *upact,
double *loact, double *uup, double *udn);
```

引数

関係する制御

なし

例

行範囲を計算して返します。

```
int rows;
double *upact, *loact, *uup, *udn;
...
XPRSrange(prob);
XPRSgetintattrib(prob, XPRS_ROWS, &rows);
upact = malloc(rows*sizeof(double));
...
XPRSgetowrange(prob, upact, loact, uup, udn);
```

補足

アクティビティとユニットコストはレンジファイル(*problem_name.rng*)から得られます。レ

ンジファイルにおける上下側列アクティビティと上下側ユニットコストの意味については、付録A「ログとファイル形式」を参照ください。

関連事項

XPRSchgrhsrange, XPRSgetcolrange.

prob	現在の問題
upact	上側行アクティビティのための長さROWSの倍精度型配列
loact	下側行アクティビティのための長さROWSの倍精度型配列
uup	上側行ユニットコストのための長さROWSの倍精度型配列
udn	下側行ユニットコストのための長さROWSの倍精度型配列

Xpress-Optimizer Reference Manual Console and Library Functions 130

XPRSgetrows 6

XPRSgetrows

目的

与えられた範囲での制約行列行の中の非零要素を返します。

概要

```
int XPRSgetrows(XPRSprob prob, int *mstart, int *mclind,
double *dmatval, int size, int *nels, int first,
int last);
```

引数

関係する制御

なし

例

最初の2つの行に、最大6つまでの非零行列要素を返し、表示します。:

prob	現在の問題
mstart	要求された各行についてのmclind、dmatval配列のオフセットの開始を示す添字が入った整数型配列。長さは少なくともlast-first+2です。列iはmrwind、dmatval配列のmstart[i]の位置で始まり、mstart[i+1]-mstart[i]要素が含まれます。必要なければNULLとなります
mclind	各行の非零要素の列添字が入る長さsizeの整数型配列。必要なければNULLとなります
dmatval	非零要素の値が入る長さsizeの倍精度型配列。必要なければNULLとなります
size	戻される要素の最大数
nels	mclind、dmatval配列における非零要素の数を返す整数のポインタ。非零要素の数がsizeよりも大きい場合は、size要素だけ返されます。nelsがsizeよりも小さい場合、nelsだけ返されます
first	その範囲での最初の行
last	その範囲での最後の行

Xpress-Optimizer Reference Manual Console and Library Functions 131

XPRSgetrows 6

```
int size=6, nels, mstart[3], mclind[6];
double dmatval[6];
...
```

```
XPRSgetrows(prob,mstart,mclind,dmatval,size,&nels,0,1);
for(i=0;i<nels;i++) printf("\t%2.1f\n",dmatval[i]);
```

補足

`mstart`, `mclind`, `dmatval` を NULL におきかえることで、列のその範囲内の要素の数が得られます。この場合、`size` には、渡される配列の長さは 0 であることを示す 0 が設定されます。

関連事項

XPRSgetcols, XPRSgetrowrange, XPRSgetrowtype.
Xpress-Optimizer Reference Manual Console and Library Functions 132

XPRSgetrowtype 6**XPRSgetrowtype****目的**

与えられた範囲における行のタイプを返します。

概要

```
int XPRSgetrowtype(XPRSprob prob, char *qrtype, int first,
int last);
```

引数**関係する制御**

なし

例

配列 `qrtype` に行タイプを戻します。

```
int rows;
char *qrtype;
...
XPRSgetintattrib(prob,XPRS_ROWS,&rows);
qrtype = (char *) malloc(sizeof(char)*rows);
XPRSgetrowtype(prob,qrtype,0,rows-1);
```

関連事項

XPRSchgrowtype, XPRSgetrowrange, XPRSgetrows.

<code>prob</code>	現在の問題
<code>qrtype</code>	行タイプが返される長さ <code>last-first+1</code> 文字の文字配列
	N 制約がないことを示します
	L 制約 < を示します
	E 制約 = を示します
	G 制約 > を示します
	R レンジ制約を示します
<code>first</code>	その範囲での最初の行
<code>last</code>	その範囲での最後の行
.	
=	
.	

Xpress-Optimizer Reference Manual Console and Library Functions 133

XPRSgetscaledinfeas 6**XPRSgetscaledinfeas****目的**

原問題についてスケーリングされた実行不能な主，双対変数のリストを返します。問題が事前

分析済みであれば、この関数がリストを返す前に事後分析がなされます。

概要

```
int XPRSgetscalinfeas(XPRSprob prob, int *npv, int *nps,
int *nds, int *ndv, int *mx, int *mslack, int *mdual,
int *mdj);
```

引数

エラー値

関係する制御

Double

prob	現在の問題
npv	主実行不能変数の数
nps	主実行不能行の数
nds	双対実行不能行の数
ndv	双対実行不能変数の数
mx	主実行不能変数が返される長さnpvの整数型配列。必要なければNULLとなります
mslack	主実行不能行が返される長さnpsの整数型配列。必要なければNULLとなります
mdual	双対実行不能行が返される長さndsの整数型配列。必要なければNULLとなります
mdj	双対実行不能変数が返される長さndvの整数型配列。必要なければNULLとなります
422	解は使えません
FEASTOL	RHSの零許容範囲
OPTIMALITYTOL	被約費用許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 134

XPRSgetscalinfeas 6

例

初めに実行不能要素数を得るためにNULLの整数型配列と共に XPRSgetscalinfeas が呼びだされます。そして配列用にスペースを割り当て、それらを入れるために関数が再び呼びだされます。

```
int *mx, *mslack, *mdual, *mdj, npv, nps, nds, ndv;
...
XPRSgetscalinfeas(prob, &npv, &nps, &nds, &ndv, NULL,
NULL, NULL, NULL);
mx = malloc(npv * sizeof(int));
mslack = malloc(nps * sizeof(int));
mdual = malloc(nds * sizeof(int));
mdj = malloc(ndv * sizeof(int));
XPRSgetscalinfeas(prob, &npv, &nps, &nds, &ndv, mx,
mslack, mdual, mdj);
```

補足

最後の4つの引数のいずれかにNULLが設定されると、実行不能に相当する数が返されます。

関連事項

XPRSgetiis, XPRSgetinfeas, XPRSiis.

Xpress-Optimizer Reference Manual Console and Library Functions 135

XPRSgetsol 6

XPRSgetsol

目的

最適化に続いて解の値を得ます。

概要

```
int XPRSgetsol(XPRSprob prob, double *x, double *slack,
double *dual, double *dj);
```

引数

関係する制御

Integer

例

次のコマンド列は、トップノードでの解(x)と最適MIP解(y)を得ます。

```
int cols;
double *x, *y;
...
XPRSmaxim(prob, "");
XPRSgetintattrib(prob, XPRS_COLS, &cols);
x = malloc(cols*sizeof(double));
XPRSgetsol(prob, x, NULL, NULL, NULL);
XPRSglobal(prob);
y = malloc(cols*sizeof(double));
XPRSgetsol(prob, y, NULL, NULL, NULL);
```

prob	現在の問題
x	主変数の値が返される長さCOLSの倍精度型配列。必要なければNULLとなります
slack	スラック変数が返される返す長さROWSの倍精度型配列。必要なければNULLとなります
dual	双対変数の値が返される長さROWSの倍精度型配列。必要なければNULLとなります
dj	各変数の被約費用が返される長さCOLSの倍精度型配列。必要なければNULLとなります
REL10STYLE	双対変数に旧型形式を使うかを表します
SOLUTIONFILE	解が読み書きされる場所の制御

Xpress-Optimizer Reference Manual Console and Library Functions 136

XPRSgetsol 6

補足

1. Optimizerはデフォルトでは、XPRSminim や XPRSmaximから出るとき、また、XPRSglobalによって整数解が得られるとき、解ファイルに解を書き込みます。XPRSgetsolがXPRSminim, XPRSmaxim, XPRSglobalの後に呼びだされる場合は、デフォルトでは、解ファイルに解が返されます。
2. XPRSglobalの後に呼びだされる場合、XPRSgetsolは最後の整数解かトップノードでのLP緩和問題の値を返します。しかし、コールバックルーチンの1つからXPRSglobalの実行中に呼びだされた場合は、現在のノードの解が返されます。これについては、SOLUTIONFILE制御を設定することで変えられます。
3. 行列がXPRSminimやXPRSmaximを呼びだした後に更新されると、解は使えません。

4. 問題が事前分析済みの場合、XPRSgetsolは原問題の解を返します。事前分析された解を得るためにはXPRSgetpresolvesolを呼び出すしかありません。
5. 線形問題があり、XPRSsminimかXPRSsmaximを何度も呼び出す場合、解ファイルに読み書きできないようにすることが有効です。XPRSsminimあるいはXPRSsmaximが終了し、行列が離散要素を含まないなら、メモリ内の解は解ファイルにある解と同じです。
6. 行列が離散要素を持つ場合は、解ファイルに書きこめないようにしてください。というのは、XPRSGlobalが終了すると、メモリにある解は最後のノードの解で、MIP最適解ではないためです。また、事前分析を実施する場合、XPRSsminim, XPRSsmaximあるいはXPRSGlobalが終了した後、メモリにある解は事前分析済みの問題の解です。解ファイルに書きこめないようにするのが基本であれば、MIP最適解は、MIP解が見つかる度に呼ばれる整数解コールバック(XPRSsetcbintsolによって設定されます) により得られます。デフォルトでは、XPRSgetsolがコールバックルーチンにより呼びだされる場合、解はメモリから読みこまれます。

関連事項

XPRSgetpresolvesol, XPRSwriteprtsol, XPRSwritesol.
Xpress-Optimizer Reference Manual Console and Library Functions 137

XPRSgetstrattrib 6

XPRSgetstrattrib

目的

種々の文字列問題特性の値を戻します。問題特性は問題のロード中、最適化中に設定されます。

概要

```
int XPRSgetstrattrib(XPRSprob prob, int ipar, char *cval);
```

引数

関係する制御

なし

例

ロードされた行列の名前を戻します。

```
char matrixname[256];
...
XPRSreadprob(prob, "myprob", "");
XPRSgetstrattrib(prob, XPRS_MATRIXNAME, matrixname);
```

関連事項

XPRSgetdblattr, XPRSgetintattr.

prob	現在の問題
ipar	値が返される問題特性。全問題特性のリストは8章「問題特性」かヘッダファイルxprs.h内を参照ください
cval	特性の値 (NULL終了分を加える) が返される文字列のポインタ

Xpress-Optimizer Reference Manual Console and Library Functions 138

XPRSgetstrcontrol 6

XPRSgetstrcontrol

目的

与えられた文字列制御パラメータの値を戻します。

概要

```
int XPRSgetstrcontrol(XPRSprob prob, int ipar,
```

```
char *cgval);
```

引数

関係する制御

7章「制御パラメータ」参照

例

MPSBOUNDNAME の値を戻されて表示されます。

```
char mpsboundname[256];
```

```
...
```

```
XPRSgetstrcontrol(prob, XPRS_MPSBOUNDNAME, mpsboundname);
```

```
printf("Name = %s\n", mpsboundname);
```

関連事項

XPRSgetdblcontrol, XPRSgetintcontrol, XPRSsetstrcontrol.

prob

現在の問題

ipar

値が返される制御パラメータ。全制御のリストは7章「制御パラメータ」か `xprs.h` ヘッダファイル内のリストを参照ください

cgval

制御の値(NULL終了分を加える) が返される文字列のポインタ

Xpress-Optimizer Reference Manual Console and Library Functions 139

XPRSgetub 6

XPRSgetub

目的

与えられた範囲での列の上限を返します。

概要

```
int XPRSgetub(XPRSprob prob, double *ub, int first,
int last);
```

引数

関係する制御

なし

例

現在の問題の列の上限を戻します。

```
int cols;
```

```
double *ub;
```

```
...
```

```
XPRSgetintattrib(prob, XPRS_COLS, &cols);
```

```
ub = (double *) malloc(sizeof(double)*ncol);
```

```
XPRSgetub(prob, ub, 0, ncol-1);
```

補足

値が `XPRS_PLUSINFINITY` 以上のときは無限大となり、`XPRS_MINUSINFINITY` 以下のときは無限小となります。

関連事項

XPRSchgbounds, XPRSgetlb.

prob

現在の問題

ub

上限が置かれてある長さ `last-first+1` の倍精度型配列

first

その範囲での最初の列

last

その範囲での最後の列

Xpress-Optimizer Reference Manual Console and Library Functions 140

XPRSgetunbvec 6**XPRSgetunbvec****目的**

シンプレックス法や双対アルゴリズムによって行列が非有界である場合の基底に入るベクトルの添字を返します。

概要

```
int XPRSgetunbvec(XPRSProb prob, int *junb);
```

引数**エラー値****関係する制御**

なし

補足

双対シンプレックス法を用いて解くとき、問題が主実行不可能な場合は、XPRSgetunbvecは双対非有界を検出したピボット行を返します。

関連事項

XPRSgetinfeas, XPRSmxim / XPRSmnim.

prob

現在の問題

junb

非有界の行列の原因となる基底に入るベクトルが返される整数のポイント。シンプレックス法では、ベクトルは制約(行) $junb$ ($junb$ が $0 \sim row-1$ の場合)か、変数(列) $junb-ROWS-SPAREROWS$ ($junb$ が $ROWS+SPAREROWS \sim ROWS+SPAREROWS+COLS-1$ の場合)のスラック変数です

91

現在の問題は使えません

Xpress-Optimizer Reference Manual Console and Library Functions 141

XPRSGlobal**GLOBAL 6****XPRSGlobal GLOBAL****目的**

XPRSmxim (MAXIM) , XPRSmnim (MINIM)を用いてLP緩和問題を解いた後、整数解探索を開始するか、中断された整数解探索を継続します。

概要

```
int XPRSGlobal(XPRSProb prob);
```

GLOBAL

引数**関係する制御**

Integer

prob

現在の問題

BACKTRACK

ノード選択基準

BREADTHFIRST

ノード選択基準の限度

COVERCUTS

トップノードにおいて導入された不等式の使用回数

CPUTIME

1 CPUタイム 0 経過時間

CUTDEPTH

カットが生成される木における最大の深さ

CUTFREQ

木探索でカットが生成される頻度

CUTSTRATEGY	カット戦略の指定
DEFAULTALG	木探索で使うアルゴリズム
GOMCUTS	トップノードでのGomory カット使用回数
KEEPMIPSOL	保持する整数解の数
MAXMIPSOL	探索するMIP解の最大数
MAXNODE	分枝限定法探索でのノードの最大数
MAXSLAVE	並列MIP探索に使われるスレーブプロセッサの数
MAXTIME	最大許容時間
MIPLOG	整数解印刷用フラグ
MIPPRESOLVE	整数事前分析のタイプ
NODESELECTION	ノード選択制御
REFACTOR	最適基底の再因子分解の実施
SBBEST	強い分枝を実行する実行不能離散要素の数
SBITERLIMIT	強い分枝を実行する双対反復の数

Xpress-Optimizer Reference Manual Console and Library Functions 142
XPRSglobal

GLOBAL 6

Double

例 1 (ライブラリ)

問題fred.mat,を入力し、解ファイルを出力する前にLPと整数問題を解きます。

```
XPRSreadprob(prob, "fred", "");
XPRSmxim(prob, "");
XPRSglobal(prob);
XPRSwriteprtsol(prob);
```

例 2 (コンソール)

コンソールOptimizerのコマンドの集合に同じです。

```
READPROB fred
MAXIM
GLOBAL
WRITEPRTSOL
```

補足

1. XPRSmxim (MAXIM) かXPRSmnim (MINIM)によりLP最適解が見つっていると、整数解探索をXPRSglobal (GLOBAL)により開始します。
2. 整数解探索を中断して、引き続きXPRSglobal (GLOBAL)が再度呼びだされる場合、終了点から探索が再開されます。トップノードで探索を再開するには、関数XPRSinitglobalを使います。

TREECOVERCUTS	木の中で導入された不等号の使用回数
TREEGOMCUTS	木におけるGomoryカットの使用回数
VARSELECTION	劣化推定制御ノード選択
MIPADDCUTOFF	新しいカットオフを与えるために目的関数に加える量
DEGRADEFACTOR	推定劣化量を乗する要素
MIPABSCUTOFF	LP最適化コマンドの後に設定するカットオフ
MIPABSSTOP	最適化停止絶対値基準
MIPRELCUTOFF	カットオフ%
MIPRELSTOP	最適化停止相対値基準

MIPTARGET	整数目的関数目標
MIPTOL	整数解の実行可能許容範囲
PSEUDOCOST	ノードの劣化量推定でのデフォルト擬コスト

Xpress-Optimizer Reference Manual Console and Library Functions 143
XPRSglobal

GLOBAL 6

- MIPLOG ノード値 n ごとに 6 列の情報の概要ログが出力されます。(付録 A.7「グローバル・ログ」参照)
- 多くの場合は、XPRSmaxim (MAXIM)/XPRSminim (MINIM) の後に XPRSglobal (GLOBAL) は直接呼ばれます。このような場合、`g` フラグを XPRSmaxim (MAXIM)/XPRSminim (MINIM) に付けて使うことで、より効果的に実行できます。
- (コンソール) 整数解探索は CTRL-C を入力すれば、次の入力を始めていない限り中断します。
- Optimizer ライブラリユーザは、MIPSTATUS 問題特性を使うことで、整数解探索の最終状態を確認できます。
- Optimizer は、複数ファイルにわたりデータが展開された 2 GB を超えるグローバル(アクティブノードリスト)ファイルをサポートしています。グローバルファイルは最初、`name probname.glb` と名付けられ、それ以降のファイル名は、`probname.glb.1`, `probname.glb.2`, ... と名付けられます。1 つのファイルが 2GB を超えることはありません。
- XPRSminim (MINIM) と XPRSmaxim (MAXIM) について説明された制御はまた、XPRSglobal (GLOBAL) アルゴリズムを制御することにも使えます。
- いくつかの MIP 解を保存する場合は、MIP 制御 `KEEPMIPSOL` を要求する解の数として 1 ~ 10 の範囲で設定します。整数解を比較検討したい場合に、この機能が役立ちます。普通、最初に見つかった MIP 解は、`probname.sol` に保存されます。`KEEPMIPSOL` が 1 よりも大きく設定されていると、2 番目により MIP 解が見つかると、最初の解のファイルの名前を `probname.sol0` と名づけ、2 番目の解を `probname.sol1` に書きこみます。3 番目、4 番目があれば同様にします。よって、10 個の整数解が保存された場合は、最も良い MIP 解は常に `probname.sol` の中に見られ、次に良い解は、`probname.sol0`、それ以降は `probname.sol9` まで次々と解を見ることができます。これらの異なる解は、Optimizer の `XPRSwritesol` (`WRITESOL`) コマンドに `e` フラグを付けることで ASCII 形式で出力できます。

関連事項

XPRsfixglobal (FIXGLOBAL), XPRSinitglobal, XPRSmaxim (MAXIM) / XPRSminim (MINIM), 付録 A.7「グローバルログ」
Xpress-Optimizer Reference Manual Console and Library Functions 144
XPRSgoal

GOAL 6

XPRSgoal GOAL

目的

目標計画を実行します。

概要

```
int XPRSgoal(XPRSprob prob, char *filename, char *flags);
GOAL [filename] [-flags]
```

引数**関係する制御***Integer***例 1 (ライブラリ)**

目標計画を実行します。

Goalexはファイル gbl.golから指示を得ます。

```
XPRsreadprob(prob, "goalex", "");
```

```
XPRsgoal(prob, "gbl", "fo");
```

例 2 (コンソール)

目的関数OBJ1のウェイトが未知の問題があり、行を最大化し、制約を最適値の5%まで緩和してよいとする目標計画を実行する場合は、以下のように問題を解きます。

```
READPROB
```

```
GOAL
```

```
P
```

```
O
```

```
prob
```

現在の問題

```
filename
```

指示が読みこまれるファイル名を含む200文字以内の文字列(拡張子.golがつきます)

```
flags
```

XPRsgoal (GOAL)に渡すフラグ

o 表示される最適化プロセスのログ

l 整数変数を線形に扱います

f ファイルfilename.grpに書き出します

```
KEEPMIPSOL
```

優先モデルにおいて保持する解の数

Xpress-Optimizer Reference Manual Console and Library Functions 145

XPRsgoal

GOAL 6

```
OBJ1
```

```
MAX
```

```
P
```

```
5
```

<empty line>

補足

- 複数の目的関数を持つ問題の解をXPRsgoal (GOAL)コマンドを用いることで探すことができます。XPRsgoal (GOAL)を制約と用いることで、実行不可能な問題の解を見つけられません。目標は最初に問題を実行可能にするために緩和した制約をになります。この緩和した制約をどれだけ満たすかを調べ、問題を実行可能にするために必要なペナルティが分かる(アルキメデスモデル)か、どの緩和制約を満たせないかが分かります(優先モデル)
- (コンソール) GOALの使用時にオプションfilenameを指定すると、プロンプトへの応答はfilename.golから読みこまれます。プロンプトに正常な応答がない場合、目標計画は終了し、制御がOptimizerに返されます。
- KEEPMIPSOLは、優先モデルにおける(制約条件か目的関数)各ステップ解を保持するのに使われます。XPRswritesol (WRITESOL)コマンドにeフラグを付けて用いることで、全ての解を出力できます。
- 目標計画問題の1つの出力を、更なる検討の入力として使うことは、常にはできません。というのは、目的関数の係数、右辺、行タイプは全て変更されることがあるためです。
- 目的関数を使うアルキメデスモデルでは、結果の目的関数の定数項は、もとの目的関数の

右辺の一次結合です。

関連事項

5.5章 「目標計画法」

Xpress-Optimizer Reference Manual Console and Library Functions 146

HELP 6

HELP

目的

Optimizerのコンソールユーザにコマンドを簡単に説明します。

概要

HELP

引数

なし

関係する制御

なし

例 (コンソール)

このコマンドはコンソールOptimizerのコマンドラインで呼びだして使います。

HELP

関連事項

なし

Xpress-Optimizer Reference Manual Console and Library Functions 147

XPRSiis

6

XPRSiis IIS

目的

線形実行不能問題の中で既約実行不可能集合(IIS)の探索を開始します。

概要

```
int XPRSiis(XPRSProb prob, char *flags);
```

```
IIS [-flags]
```

引数

関係する制御

Integer

例 1 (ライブラリ)

IISを探索し、問題特性IISを調べてその数を示します。

```
int iis;
```

```
...
```

```
XPRSiis(prob, "f");
```

```
XPRSgetintattrib(prob, XPRS_IIS, &iis);
```

```
printf("IIS = %d\n", iis);
```

例 2 (コンソール)

制御パラメータMAXIIS で指定した数のIISを探します。

```
IIS
```

```
prob
```

```
flags
```

現在の問題

XPRSiis (IIS) に渡すフラグ。次のように設定できます

- IIS情報と最適化過程ログを画面に表示します

f ファイル `problem_name.iis` に IIS 情報を書きこみます
 フラグがない場合は、IIS 情報だけが画面に表示されます
 見つける既約実行不可能集合の数

MAXIIS
 Xpress-Optimizer Reference Manual Console and Library Functions 148
 XPRSiiis

IIS 6

補足

1. モデルには実行不能な要素がいくつか存在する場合があります。1 つの IIS を修正しても、モデルが実行可能にならないかもしれません。よって、Optimizer はモデルが実行不能になる場合の IIS を見つけます。制御 MAXIIS を正の整数に設定すると、XPRSiiis (IIS) コマンドは IIS が MAXIIS 個見つかるまで終了します。デフォルトでは、制御 MAXIIS は -1 に設定されていて、実行不能になるたびに独立な IIS を見つけます。
2. 問題特性 IIS は特定の探索で見つかった IIS の個数を戻すことができます。詳細は例 1 を参照してください。

関連事項

XPRSgetiis, 3.2 章「実行不能」
 Xpress-Optimizer Reference Manual Console and Library Functions 149

XPRSinit 6

XPRSinit

目的

Optimizer ライブラリを初期化します。これは、他のライブラリルーチンよりも前に呼ばれなければいけません。

概要

```
int XPRSinit(char *xpress);
```

引数

関係する制御

なし

例

XPRSinit を呼ぶ通常の方法です。

```
if(XPRSinit(NULL)) printf("Problem with XPRSinit\n");
```

補足

1. エラーチェックは全てのライブラリ関数の呼び出しで常に使用されるべきですが、大多数のエラーは初期化段階で起こるため、初期化関数を使ってエラーチェックを行うのは特に重要です。零でないリターンコードは、ライセンスが見つからなかったことを表します。このような場合には、アプリケーションは終了します。但し、リターンコード 3 2 の場合は、学生用ライセンスが見つかったとしてソフトは動きますが、機能や容量に制限があります。
2. 全てのスレッドが同じであるマルチスレッドアプリケーションでは、XPRSinit はライブラリに先だって各スレッドにより呼びだされます。初期化は一度だけ実行される、各スレッドが必要なだけライブラリ関数を使えることを保証します。他のスレッドを作り出すクリアマスタースレッドをもつアプリケーションの場合、マスタースレッドによって初期化操作を呼びだす必要があります。

関連事項

XPRScreateprob, XPRSfree.

Xpress

Xpresss-MP のパスワードファイルのあるディレクトリ。指定がなければNULLとし、標準の初期化ディレクトリがチェックされます。

Xpress-Optimizer Reference Manual Console and Library Functions 150

XPRSinitglobal 6**XPRSinitglobal****目的**

整数木探索を再初期化します。デフォルトでは、XPRSglobalが中断されて再度呼びだされた場合、整数解探索は中断したところから再開されます。最初にXPRSglobal を呼び、その後XPRSinitglobalが呼ばれた場合は、XPRSglobalが再度呼ばれた時、整数解探索はトップノードからスタートします。

概要

```
int XPRSinitglobal(XPRSprob prob);
```

引数**関係する制御**

なし

例

問題を再度解こうとする前に整数解探索を初期化します。

```
XPRSinitglobal(prob);
```

```
XPRSmxim(prob, "g");
```

関連事項

XPRSglobal, XPRSmxim / XPRSmnim.

prob

現在の問題

Xpress-Optimizer Reference Manual Console and Library Functions 151

XPRSlloadbasis 6**XPRSlloadbasis****目的**

ユーザエリアに基底をロードします。

概要

```
int XPRSlloadbasis(XPRSprob prob, int *rstatus,
int *cstatus);
```

引数**関係する制御**

なし

例

問題をロードし、最適化を高速化するために類似問題から(事前に最適化済の)基底を再ロードします。

```
XPRSreadprob(prob, "problem", "");
```

```
XPRSlloadbasis(prob, rstatus, cstatus);
```

```
XPRSmnim(prob, "");
```

補足

新しい基底の保存により問題が変更された場合、それをロードする前に以下のように基底を変更できます。

prob	現在の問題
rstatus	各行に関係するスラック, サープラス, 人為変数の基底状態を含む長さROWSの整数型配列。状態は以下のいずれかです 0 スラック, サープラス, 人為変数は下限で基底ではありません 1 スラック, サープラス, 人為変数は基底です 2 スラック, サープラス変数は上限で基底ではありません
cstatus	制約行列内の各列の基底状態を含む長さCOLSの整数型配列。状態は以下のいずれかです 0 変数に下限値がなければ、変数は下限で基底でない、あるいは零で超基底です 1 変数は基底です 2 変数は上限にあります

Xpress-Optimizer Reference Manual Console and Library Functions 152

XPRSloadbasis 6

- 変数が無限の下限か有限の上限を持たなければ、下限(cstatus[icol]=0)で新しい非基底変数を作ります。この場合、上限で非基底変数を作ります(cstatus[icol]=2)
- 新しい制約基底を作ります (rstatus[jrow]=1)
- 基底変数、非基底制約を削除しないようにします

関連事項

XPRSgetbasis, XPRSgetpresolvebasis, XPRSloadpresolvebasis.
Xpress-Optimizer Reference Manual Console and Library Functions 153

XPRSloadcuts 6

XPRSloadcuts

目的

カットプールから行列にカットをロードします。XPRSloadcutsを呼ばないと、カットはカットプールに残りますが、ノードではアクティブでなくなります。XPRSdelcutsで削除しなければ、ノードでロードされたカットは全ての子ノードでアクティブなままです。

概要

```
int XPRSloadcuts(XPRSprob prob, int itype, int interp,
int ncuts, int *mcutind);
```

引数

関係する制御

なし

関連事項

XPRSaddcuts, XPRSdelcpcuts, XPRSdelcuts, 5.4章「カットマネージャのはたらき」

prob	現在の問題
itype	カットタイプ
interp	カットタイプの解釈

- 1 全てのカットをロードします
- 1 カットタイプを数字で扱います
- 2 カットタイプをビットマップで扱います。いずれかのビットがitypeに設定されたビットと一致する場合、カットをロードします

ncuts 3 カットタイプをビットマップで扱います。全てのビットがitypeに設定されるビットと一致する場合、カットをロードします。ロードするカットの数。値-1は、タイプitypeの全てのカットをロードすることを示します

mcutind ロードするカットの添字番号を含む整数型配列。ncutsが-1の場合、この配列はNULLです。そうでなければ、長さncutsです。-1の添字は無視され、XPRSstorecutsから返される配列mindexは、直接XPRsloadcutsに引き渡されます

Xpress-Optimizer Reference Manual Console and Library Functions 154

XPRsloaddirs 6

XPRsloaddirs

目的

指示を行列にロードします。

概要

```
int XPRsloaddirs(XPRSprob prob, int ndir, int *mcols,
int *mpri, char *qbr, double *dupc, double *ddpc);
```

引数

関係する制御

なし

関連事項

XPRSgetdirs, XPRsloadpresolvedirs, XPRSreaddirs.

prob	現在の問題
ndir	指示の数
mcols	列番号を含む長さndirの整数型配列。負の値は集合番号を示します (第1集合は-1、第2集合は-2と続きます)
mpri	列、集合の優先度を含む長さndirの整数型配列。必要なければNULLとなります
qbr	各列、集合の分枝方向を示す長さndirの文字配列 U 押し上げ要素 D 押し下げ要素 N 特になし 必要なければNULLとなります
dupc	列、集合の上側擬コストを含む長さndirの倍精度型配列。必要なければNULLとなります
ddpc	列、集合の下側擬コストを含む長さndirの倍精度型配列。必要なければNULLとなります

Xpress-Optimizer Reference Manual Console and Library Functions 155

XPRsloadglobal 6

XPRsloadglobal

目的

整数問題をOptimizerデータ構造にロードします。整数、バイナリ、部分整数、半連続、半連続整数変数は、タイプ1と2の集合と一緒に定義されます。集合要素の参照行の値は、参照行を指定するのではなく配列として渡されます。

概要

```
int XPRSloadglobal(XPRSprob prob, char *probname, int ncol,
int nrow, char *qrtype, double *rhs, double *range,
double *obj, int *mstart, int *mnel, int *mrwind,
double *dmatval, double *dlb, double *dub, int ngents,
int nsets, char *qgtype, int *mgcols, double *dlim,
char *qstype, int *msstart, int *mscols, double *dref);
```

引数

prob	現在の問題
probname	問題名を含む200文字以内の文字列
ncol	行列の構造列の数
nrow	目的関数行を含まない行列の行の数
qrtype	行タイプを含む長さnrowの文字配列 L 制約を示します E 制約を示します G 制約を示します R 範囲制約を示します N 非束縛制約を示します
rhs	右辺の係数を含む長さnrowの倍精度型配列
range	レンジ行のレンジ値を含む長さnrowの倍精度型配列。必要なければNULLとなります
obj	目的関数の係数を含む長さncolの倍精度型配列
mstart	各列要素の開始のmrwind, dmatval配列のオフセットを含む整数型配列。この配列の長さはncolであるか、mnelがNULLの場合は長さncol+1です
.	
=	
.	

Xpress-Optimizer Reference Manual Console and Library Functions 156

XPRSloadglobal 6

mnel	各列の非零要素の数を含む長さncolの整数型配列。必要なければNULLとなります。
mrwind	各列の非零要素の行添字を含む整数型配列。添字が昇順に入力されている場合、長さはmstart[ncol-1]+mnel[ncol-1]であるか、mnelがNULLの場合は長さmstart[ncol]です
dmatval	mrwindの非零要素値長さを含む倍精度型配列
dlb	列の下限を含む長さncolの倍精度型配列
dub	列の上限を含む長さncolの倍精度型配列
ngents	バイナリ, 整数, 半連続, 半連続整数, 部分整数要素の数
nsets	SOS1, SOS2集合の数
qgtype	要素タイプを含む長さngentsの文字配列 B バイナリ変数 I 整数変数 P 部分整数変数 S 半連続変数

	R	半連続整数変数
mgcols		離散要素の列添字を含む長さngetsの整数型配列
dlim		部分整数変数と半連続、半連続整数変数の下限を含む長さngentsの倍精度型配列。(バイナリや整数変数に相当する位置の要素は無視されます)。必要なければNULLとなります
qstype		集合タイプを含む長さnsetsの文字配列 1 SOS1 タイプ集合 2 SOS2 タイプ集合 必要なければNULLとなります
msstart		集合のスタートを示すmscols, dref配列のオフセットを含む整数型配列。この配列は長さnsets+1で、最後の要素は集合nsets+1の開始位置のオフセットを含みます。必要なければNULLとなります。

Xpress-Optimizer Reference Manual Console and Library Functions 157

XPRsloadglobal 6

関係する制御

Integer

Double

例

整数x,yを用いた整数問題globalExです。

```
char probname[] = "globalEx";
int ncol = 2, nrow = 2;
char qrtype[] = {'L', 'L'};
double rhs[] = {400.0, 200.0};
int mstart[] = {0, 2, 4};
int mrwind[] = {0, 1, 0, 1};
double dmatval[] = {3.0, 1.0, 2.0, 3.0};
double objcoefs[] = {1.0, 2.0};
double dlb[] = {0.0, 0.0};
double dub[] = {200.0, 200.0};
```

mscols 各集合の列を含む長さmsstart[nsets]-1の整数型配列。必要なければNULLとなります。

dref 集合の各要素の参照行要素を含む長さmsstart[nsets]-1の倍精度型配列。必要なければNULLとなります

EXTRAELEMS 追加できる行列要素数

EXTRACOLS 追加できる列の数

EXTRAMIPENTS 追加できる離散要素数

EXTRAPRESOLVE 事前分析で追加できる要素数

EXTRAROWS 追加できる行の数

KEEPNROWS 非束縛行の状態

SCALING スケーリングのタイプ

MATRIXTOL 行列要素の零許容範囲

SOSREFTOL 参照行要素間の最小ギャップ

Maximize

Subject to:

$x + 2y +$

$3x + 2y + 400 .$

$x3y + 200$.
Xpress-Optimizer Reference Manual Console and Library Functions 158

XPRSloadglobal 6

```
int ngents = 2;
int nsets = 0;
char qrtype[] = {'I', 'I'};
int mgcols[] = {0, 1};
...
XPRSloadglobal(prob, probname, ncol, nrow, qrtype, rhs,
NULL, objcoefs, mstart, NULL, mrwind, dmatval, dlb,
dub, ngents, nsets, qrtype, mgcols, NULL, NULL,
NULL, NULL, NULL);
```

補足

1. 行と列の添字は、通常、0からnrow-1まで、0からncol-1でCの表記に各々従います。
2. ライブラリヘッダファイルで定義される2つの定数XPRS_PLUSINFINITYとXPRS_MINUSINFINITYは、上下界値配列内で各々プラス・マイナス無限（大・小）を示すのに使われます。
3. レンジ配列の値はRタイプ行についてのみ読みこまれます。他のタイプの行については無視されます。

関連事項

XPRSaddsetnames, XPRSloadlp, XPRSloadqglobal, XPRSloadqp,
XPRSreadprob.
Xpress-Optimizer Reference Manual Console and Library Functions 159

XPRSloadlp 6

XPRSloadlp

目的

ファイルから行列を読むのではなく、行列を直接Optimizerに渡せます。

概要

```
int XPRSloadlp(XPRSprob prob, char *probname, int ncol,
int nrow, char *qrtype, double *rhs, double *range,
double *obj, int *mstart, int *mnel, int *mrwind,
double *dmatval, double *dlb, double *dub);
```

引数

prob	現在の問題
probname	問題名を含む200文字以内の文字列
ncol	行列の構造列の数
nrow	行列の行の数(目的関数を含みません)
qrtype	行タイプを含む長さnrowの文字配列
	L 制約を示します
	E 制約を示します
	G 制約を示します
	R レンジ制約を示します
	N 非束縛制約を示します
rhs	行の右辺係数を含む長さnrowの倍精度型配列
range	レンジ列のレンジ値を含む長さnrowの倍精度型配列。必要なければ

obj NULLとなります
 目的関数の係数を含む長さncolの倍精度型配列
 mstart 各列の要素の開始のmrwind, dmatval配列のオフセットを含む整数型配列。この配列は長さncolであるか、mnelがNULLのときに長さncol+1です
 mnel 各行における非零要素の数を含む長さncolの整数型配列。必要なければNULLとなります

.
 =
 .

Xpress-Optimizer Reference Manual Console and Library Functions 160

XPRloadlp 6

関係する制御

Integer

Double

例

LP問題をXPRloadlpを使ってOptimizerにロードします。

```
char probname[] = "small";
```

```
int ncol = 2, nrow = 4;
```

```
char qrtype[] = {'N', 'G', 'G', 'G'};
```

```
double rhs[] = { 0 , 3 , 3 , 1 };
```

mrwind 各列の非零要素の行添字を含む整数型配列。添字が昇順に列に入力されている場合、mrwindの長さは、mstart[ncol-1]+mnel[ncol-1]であるか、mnelがNULLの場合は、mstart[ncol]となります

dmatval dmatval 非零要素値を含む倍精度型配列で、長さはmrwindによります

dlb 列の下限を含む長さncolの倍精度型配列

dub 列の上限を含む長さncolの倍精度型配列

EXTRAELEMENTS 追加できる行列要素の数

EXTRACOLS 追加できる列の数

EXTRAPRESOLVE 事前分析において追加できる要素の数

EXTRAROWS 追加できる行の数

KEEPNROWS 非束縛行の状態

SCALING スケーリングのタイプ

MATRIXTOL 行列要素の零許容範囲

Maximize

Subject to:

$xy +$

$2x3 >$

$x2y + 3 >$

$xy + 1 >$

Xpress-Optimizer Reference Manual Console and Library Functions 161

XPRloadlp 6

```
double obj[] = { 1 , 1 };
```

```
int mstart[] = { 0 , 4 , 7 };
```

```
int mrwind[] = { 0 , 1 , 2 , 3 , 0 , 2 , 3 };
double dmatval[] = { 1 , 2 , 1 , 1 , 1 , 2 , 1 };
double dlb[] = { 0 , 0 };
double dub[] = { XPRS_PLUSINFINITY, XPRS_PLUSINFINITY };
XPRSloadlp(prob, probname, ncol, nrow, qrtype, rhs, NULL,
obj, mstart, NULL, mrwind, dmatval, dlb, dub)
```

補足

1. 行列の添字は通常、0からnrow-1, 0からncol-1のCの表記に各々従います。
2. ライブラリヘッダファイルで定義される2つの定数 XPRS_PLUSINFINITYと XPRS_MINUSINFINITYは、限度配列においてプラス・マイナス無限（大・小）を表すのに使われます。
3. range配列の値はRタイプの行についてのみ読まれます。他のタイプの行については無視されます。
4. レンジ制約について、rhs配列の値は上限制約を表し、rng配列の値は制約のレンジを表します。よってレンジ制約は次のようになります。

関連事項

XPRSloadglobal, XPRSloadqglobal, XPRSloadqp, XPRSreadprob.

$rhs_j - a_{ij}x_i$

__ rhsj..

Xpress-Optimizer Reference Manual Console and Library Functions 162

XPRSloadmodelcuts 6**XPRSloadmodelcuts****目的**

行列の行の集合をモデルカットとして扱うように指定します。

概要

```
int XPRSloadmodelcuts(XPRSprob prob, int nmod, int *mrows);
```

引数**エラー値****関係する制御**

なし

例

最初の6行を整数問題myprobのモデルカットとして設定します。

```
int mrows[] = {0,1,2,3,4,5}
...
XPRSloadmodelcuts(prob,6,mrows);
XPRSminim(prob,"g");
```

補足

1. 事前分析の間、モデルカットは行列から取り除かれます。最適化に続いて、違反モデルカットは行列の後ろに追加され、行列は再度最適化されます。これは、違反カットが存在する限り続きます。
2. モデルカットは「真の」モデルカットでなければならず、その点で、それらは最適MIP解で冗長です。

関連事項

5.4章「カットマネージャのはたらき」参照

prob 現在の問題
 nmod モデルカットの数
 mrows カットとして扱われる行添字の配列
 268 事前分析された行列には操作できません
 Xpress-Optimizer Reference Manual Console and Library Functions 163

XPRSloadpresolvebasis 6

XPRSloadpresolvebasis

目的

ユーザエリアから事前分析された基底をロードします。

概要

```
int XPRSloadpresolvebasis(XPRSprob prob, int *rstatus,
int *cstatus);
```

引数

関係する制御

なし

例

ある問題の事前分析された基底を保存し、他の問題にロードします。

```
int rows, cols, *rstatus, *cstatus;
...
XPRSreadprob(prob, "myprob", "");
XPRSminim(prob, "");
XPRSgetintattrib(prob, XPRS_ROWS, &rows);
XPRSgetintattrib(prob, XPRS_COLS, &cols);
rstatus = malloc(rows*sizeof(int));
cstatus = malloc(cols*sizeof(int));
XPRSgetpresolvebasis(prob, rstatus, cstatus);
```

prob	現在の問題
rstatus	各行に關係するスラック, サープラス, 人為変数の基底状態を含む長さROWSの整数型配列。状態は以下のいずれかです 0 スラック, サープラス, 人為変数は下限で基底ではありません 1 スラック, サープラス, 人為変数は基底です 2 スラックあるいはサープラス変数は上限で基底ではありません
cstatus	行列の各列の基底状態を含む長さCOLSの整数型配列。状態は次のいずれかです 0 変数は下限で非基底。あるいは変数に下限がない場合は零で基底です 1 変数は基底です 2 変数は上限にあります

Xpress-Optimizer Reference Manual Console and Library Functions 164

XPRSloadpresolvebasis 6

```
XPRSreadprob(prob2, "myotherprob", "");
XPRSminim(prob2, "");
XPRSloadpresolvebasis(prob2, rstatus, cstatus);
```

関連事項

XPRSgetbasis, XPRSgetpresolvebasis, XPRSloadbasis.

Xpress-Optimizer Reference Manual Console and Library Functions 165

XPRsloadpresolvedirs 6

XPRsloadpresolvedirs

目的

指示を事前分析された行列にロードします。

概要

```
int XPRsloadpresolvedirs(XPRsprob prob, int ndir,
int *mcols, int *mpri, char *qbr, double *dupc,
double *ddpc);
```

引数

関係する制御

なし

例

行列の列0に優先指示をロードします。

```
int mcols[] = {0}, mpri[] = {1};
```

```
...
```

```
XPRsminim(prob, "");
```

```
XPRsloadpresolvedirs(prob, 1, mcols, mpri, NULL, NULL, NULL);
```

prob	現在の問題
ndir	指示の数
mcols	列番号を含む長さndirの整数型配列。負の値は集合番号を示します。 (-1は最初の集合、-2は2番目の集合...と続きます)
mpri	各列や集合の優先度を含む長さndirの整数型配列。必要なければ NULLとなります
qbr	各列あるいは集合の分枝方向を指定する長さndirの文字配列 U 押し上げ要素 D 押し下げ要素 N 特に指定無し 必要なければNULLとなります
dupc	列や集合の押し上げ擬コストを含む長さndirの倍精度型配列。必要な ければNULLとなります
ddpc	列や集合の押し下げ擬コストを含む長さndirの倍精度型配列。必要な ければNULLとなります列や集合の押し下げ擬コストを含む長さ ndirの倍精度型配列。必要なければNULLとなります

Xpress-Optimizer Reference Manual Console and Library Functions 166

XPRsloadpresolvedirs 6

```
XPRsminim(prob, "g");
```

関連事項

XPRsgetdirs, XPRsloadadds.

Xpress-Optimizer Reference Manual Console and Library Functions 167

XPRsloadqglobal 6

XPRsloadqglobal

目的

二次目的係数を持つ整数問題をOptimizerデータ構造にロードします。整数, バイナリ, 部分

整数, 半連続, 半連続整数変数はタイプ 1 と 2 の集合とともに定義されます。集合要素の参照行の値は参照行を指定するのではなく、配列として引き渡されます。

概要

```
int XPRsloadqglobal(XPRsprob prob, char *probname,
int ncol, int nrow, char *qrtype, double *rhs,
double *range, double *obj, int *mstart, int *mnel,
int *mrwind, double *dmatval, double *dlb, double *dub,
int nqtr, int *mqc1, int mqc2, double *dqe, int ngents,
int nsets, char *qgtype, int *mgcols, double *dlim,
char *qstype, int *msstart, int *mscols, double *dref);
```

引数

prob	現在の問題
probname	問題名を含む200文字以内の文字列
ncol	行列の構造列の数
nrow	目的行以外の行列の行の数 目的行以外の行列の行の数
qrtype	行タイプを含む長さnrowの文字配列 L 制約を示します E 制約を示します G 制約を示します R レンジ制約を示します N 非束縛制約を示します
rhs	右辺の係数を含む長さnrowの倍精度型配列
range	レンジ行のレンジ値を含む長さnrowの倍精度型配列。レンジ配列の値はRタイプの行についてのみ読まれます。他のタイプの行については無視されます。必要なければNULLとなります
obj	目的関数係数を含む長さncolの倍精度型配列
.	.
=	.
.	.

Xpress-Optimizer Reference Manual Console and Library Functions 168

XPRsloadqglobal 6

mstart	各列の要素の開始のmrwind, dmatval 配列のオフセットを含む整数型配列。この配列の長さはncolであるか、mnelがNULLの場合、長さncol+1です
mnel	各列の非零要素数を含む長さncolの整数型配列。必要なければNULLとなります
mrwind	各列の非零要素の行添字を含む整数型配列。添字が列について昇順に入力されている場合、mrwindの長さは、mstart[ncol-1]+mnel[ncol-1] であるか、mnelがNULLの場合にはmstart[ncol]です
dmatval	mrwind の非零要素値の長さを含む倍精度型配列 mrwind の非零要素値の長さを含む倍精度型配列
dlb	列の下限を含む長さncolの倍精度型配列
dub	列の上限を含む長さncolの倍精度型配列
nqtr	二次項の数

mqc1	各二次項の第1変数の列添字を含むサイズnqtrの整数型配列
mqc2	各二次項の第2の変数の列添字を含むサイズnqtrの整数型配列
dqe	二次係数を含むサイズnqtrの倍精度型配列
ngents	バイナリ, 整数, 半連続, 半連続整数, 部分整数の要素数
nsets	SOS1, SOS2集合の数
qgtype	要素タイプを含む長さngentsの文字配列
	B バイナリ変数
	I 整数変数
	P 部分整数変数
	S 半連続整数
	R 半連続整数変数
mgcols	離散要素の列添字を含む長さngentsの整数型配列

Xpress-Optimizer Reference Manual Console and Library Functions 169

XPRsloadqglobal 6

関係する制御

Integer

Double

例

以下の整数条件での最小化問題です。

```
int nrow = 1, ncol = 2, nquad = 3;
int mstart[] = {0, 1, 2};
```

dlim	部分整数変数の整数限度や半連続変数、半連続整数変数の下限値が含まれる長さngentsの倍精度型配列（バイナリ変数や整数変数に相当する位値の要素は無視されます）。必要なければNULLとなります
qstype	長さnsetsの文字配列で次を含みます 1 SOS1 タイプの集合 2 SOS2 タイプの集合 必要なければNULLとなります
msstart	集合の開始を示すmscols, dref配列のオフセットを含む整数型配列。この配列は、長さがnsets+1で、最後の要素は集合nsets+1の開始位置オフセットを含みます。必要なければNULLとなります
mscols	各集合の列を含む長さmsstart[nsets]-1の整数型配列。必要なければNULLとなります
dref	集合の各要素の参照行要素を含む長さmsstart[nsets]-1の倍精度型配列。必要なければNULLとなります
EXTRAELEMS	追加できる行列要素の数
EXTRACOLS	追加できる列の数
EXTRAMIPENTS	追加できる離散要素の数
EXTRAPRESOLVE	事前分析で追加できる要素の数
EXTRAROWS	追加できる行の数
KEEPNROWS	非束縛行の状態
SCALING	スケーリングのタイプ
MATRIXTOL	行列要素の零許容範囲
SOSREFTOL	参照行要素間の最小ギャップ

```

6x1 - 2x1 2
2x1 x2 - 2x2 2
++ x1 x2 + 1.9 . x1
Xpress-Optimizer Reference Manual Console and Library Functions 170

```

XPRsloadqglobal 6

```

int mrwind[] = {0, 0};
double dmatval[] = {1, 1};
double rhs[] = {1.9};
char qrtype[] = {'L'};
double lbound[] = {0, 0};
double ubound[] = {XPRS_PLUSINFINITY, XPRS_PLUSINFINITY};
double obj[] = {-6, 0};
int mqc1[] = {0, 0, 1};
int mqc2[] = {0, 1, 1};
double dquad[] = {4, -2, 4};
int ngents = 1, nsets = 0;
int mgcols[] = {0};
char qgtype[]={'I'};
double *primal, *dual;
primal = malloc(ncol*sizeof(double));
dual = malloc(nrow*sizeof(double));
...
XPRsloadqglobal(prob, "myprob", ncol, nrow, qrtype, rhs,
NULL, obj, mstart, NULL, mrwind, dmatval, lbound,
ubound, nquad, mqc1, mqc2, dquad, ngents, nsets,
qgtype, mgcols, NULL, NULL, NULL, NULL, NULL)

```

補足

1. 行と列の添字は通常、0からnrow-1、0からncol-1のCの条件に各々従います。
2. ライブラリヘッダファイルで定義される2つの定数 XPRS_PLUSINFINITYと XPRS_MINUSINFINITYは、プラス・マイナス無限(大・小)を上下界値配列において各々表すのに使われます。

関連事項

XPRSaddsetnames, XPRsloadqglobal, XPRsloadlp, XPRsloadqp, XPRsreadprob.

Xpress-Optimizer Reference Manual Console and Library Functions 171

XPRsloadqp 6

XPRsloadqp

目的

二次問題をOptimizerデータ構造にロードします。この問題では、二次項は制約ではなく、目的関数にあります。

概要

```

int XPRsloadqp(XPRsprob prob, char *probname, int ncol,
int nrow, char *qrtype, double *rhs, double *range,
double *obj, int *mstart, int *mnel, int *mrwind,
double *dmatval, double *dlb, double *dub, int nqtr,
int *mqc1, int *mqc2, double *dqe);

```

引数

prob	現在の問題
------	-------

probname	問題名を含む200文字以内の文字列
ncol	行列の構造列の数
nrow	行列の行の数 (目的行を含みません)
qrtype	行タイプを含む長さnrowの文字配列 L 制約 < を示します E 制約 = を示します G 制約 > を示します R レンジ制約を示します N 非束縛制約を示します
rhs	行の右辺係数を含む長さnrowの倍精度型配列
range	レンジ行のレンジ値を含む長さnrowの倍精度型配列。レンジ制約がなければNULLとなります
obj	目的関数の係数を含む長さncolの倍精度型配列
mstart	各列の要素の開始のmrwind, dmatval配列のオフセットを含む整数型配列。この配列は長さncolであるか、mnelがNULLの場合には長さncol+1となります

.
=
.

Xpress-Optimizer Reference Manual Console and Library Functions 172

XPRSloadqp 6

関係する制御

Integer

Double

例

以下の条件での最小化です。

```
int nrow = 1, ncol = 2, nquad = 3;
```

mnel	各列の非零要素の数を含む長さncolの整数型配列。全ての要素が連続で、ncol+1列の要素の開始位置のオフセットをmstart[ncol]が含む場合は、NULLになります。
mrwind	各列の非零要素の行添字を含む整数型配列。添字が列の昇順に入力される場合、mrwindの長さはmstart[ncol-1]+mnel[ncol-1]であるか、mnelがNULLの場合はmstart[ncol]になります
dmatval	非零要素値を含む倍精度型の配列。長さはmrwindによります
dlb	列の下限を含む長さncolの倍精度型配列
dub	列の上限を含む長さncolの倍精度型配列
nqtr	二次項の数
mqc1	各二次項における第1変数の列添字を含むサイズnqtrの整数型配列
mqc2	各二次項における第2変数の列添字を含むサイズnqtrの整数型配列
dqe	二次係数を含むサイズnqtrの倍精度型配列
EXTRALEMS	追加できる行列要素の数
EXTRACOLS	追加できる列の数
EXTRAPRESOLVE	事前分析で追加できる要素の数
EXTRAROWS	追加できる行の数
KEEPNROWS	非束縛行の状態

SCALING スケーリングのタイプ
 MATRIXTOL 行列要素の零許容範囲
 $6 \times 1 - 2 \times 1 \ 2$
 $2 \times 1 \ x_2 - 2 \times 2 \ 2$
 $++ \ x_1 \ x_2 + 1.9 .$
 Xpress-Optimizer Reference Manual Console and Library Functions 173

XPRsloadqp 6

```
int mstart[] = {0, 1, 2};
int mrwind[] = {0, 0};
double dmatval[] = {1, 1};
double rhs[] = {1.9};
char qrtype[] = {'L'};
double lbound[] = {0, 0};
double ubound[] = {XPRS_PLUSINFINITY, XPRS_PLUSINFINITY};
double obj[] = {-6, 0};
int mqc1[] = {0, 0, 1};
int mqc2[] = {0, 1, 1};
double dquad[] = {4, -2, 4};
double *primal, *dual;
primal = malloc(ncol*sizeof(double));
dual = malloc(nrow*sizeof(double));
...
XPRsloadqp(prob, "example", ncol, nrow, qrtype, rhs,
NULL, obj, mstart, NULL, mrwind, dmatval, lbound,
ubound, nquad, mqc1, mqc2, dquad)
```

補足

1. 行と列の添字は通常、0からnrow-1, 0からncol-1のCの条件に各々従います。
2. ライブラリヘッダファイルで定義された2つの定数XPRS_PLUSINFINITYとXPRS_MINUSINFINITYは、上下界値配列で各々プラス・マイナス無限(大・小)を表すために使われます。
3. レンジ配列の値はRタイプの行だけ読み込まれます。他のタイプの行については無視されます。

関連事項

XPRsloadglobal, XPRsloadlp, XPRsloadqglobal, XPRsreadprob.
 Xpress-Optimizer Reference Manual Console and Library Functions 174

XPRsloadsecurevecs 6

XPRsloadsecurevecs

目的

行列から行や列を事前分析で取り除くのを防ぐために、行や列にマークします。

概要

```
int XPRsloadsecurevecs(XPRsprob prob, int nr, int nc,
int *mrow, int *mcol);
```

引数

関係する制御

なし

例

事前分析において最初の6行と4列を取り除かないようにします。

```

int mrow[] = {0,1,2,3,4,5};
int mcol[] = {0,1,2,3};
...
XPRSreadprob(prob, "myprob", "");
XPRSloadsecurevecs(prob, 6, 4, mrow, mcol);
XPRSminim(prob, "");

```

関連事項

5.2章 「事前分析のはたらき」

prob	現在の問題
nr	マークされた行の数
nc	マークされた列の数
mrow	マークされた行を含む長さnrの整数型配列。必要なければNULLとなります
mcol	マークされた列を含む長さncの整数型配列。必要なければNULLとなります

Xpress-Optimizer Reference Manual Console and Library Functions 175

XPRSmxim / XPRSminim

MAXIM / MINIM 6

XPRSmxim / XPRSminim MAXIM / MINIM

目的

LP最適解の探索を開始します。

概要

```

int XPRSmxim(XPRSprob prob, char *flags);
int XPRSminim(XPRSprob prob, char *flags);
MAXIM [-flags]
MINIM [-flags]

```

引数

関係する制御

Integer

prob	現在の問題
flags	XPRSmxim (MAXIM) か XPRSminim (MINIM)に渡すフラグ。デフォルトは"" あるいは NULLで、その場合、使われるアルゴリズムは制御DEFAULTALGで決められます。引数は以下の通りです b ニュートンバリア法を使って解きます p 主シンプレックス法を使ってモデルを解きます d 双対シンプレックス法を使ってモデルを解きます l (Lの小文字)離散要素の不連続性を無視した線形モデルとして解きます g XPRSGlobal(GLOBAL)を呼びだしてグローバルモデルを解きます オプションをいくつか組み合わせて使うことができ、例えば、pgは主シンプレックス法でLPを解き、続けて整数解探索を実行します 振動の実行 ニュートンバリア法の最大繰返し数 バリア法で使われるメモリの量
AUTOPERTURB	
BARITERLIMIT	
BARMEMORY	

BARORDER コレスキー因子分解で使うアルゴリズム
 BAROUTPUT ニュートンバリア法での 解出力のレベル
 BARTHREADS 実行するスレッドの最大数
 Xpress-Optimizer Reference Manual Console and Library Functions 176
XPR\$maxim / XPR\$minim

MAXIM / MINIM 6

Double

BIGMMETHOD Big M法か phaseI/phaseIIかの指定
 CACHESIZE ニュートンバリア法のキャッシュサイズ (単位はkバイト)
 CPUTIME 1 CPUタイム 0 経過時間
 CRASH クラッシュのタイプ
 CROSSOVER ニュートンバリア法のクロスオーバー制御
 DEFAULTALG 木探索で使うアルゴリズム
 DENSECOLLIMIT 多くの要素が密と考えられる列
 INVERTFREQ 反転頻度
 INVERTMIN 反転間の最小反復数
 KEEPBASIS 前もってロードされた基底の使用
 LPITERLIMIT シンプレックス法の繰返し限度
 LPLOG シンプレックス法の繰返しログの表示頻度
 MAXTIME 最大許容時間
 PRESOLVE 事前分析程度
 PRESOLVEOPS 事前分析中に実施する操作の指定
 PRICINGALG プライシングのタイプ
 REFACTOR 最適基底の再因子分解の指示
 SOLUTIONFILE 解が読み書きされる位置の制御
 TRACE 事前分析中の実行不能性診断の制御
 BARDUALSTOP 双対実行不能におけるニュートンバリア許容範囲
 BARGAPSTOP 相対双対ギャップにおけるニュートンバリア許容範囲
 BARPRIMALSTOP 主実行不能におけるニュートンバリア許容範囲
 BARSTEPSTOP ニュートンバリア法の最小ステップサイズ
 BIGM 実行不能ペナルティ
 CHOLESKYTOL コレスキー分解の零許容範囲
 ELIMTOL 事前分析の消去フェーズのマルコビット許容範囲
 ETATOL イータ要素の零許容範囲
 FEASTOL RHSの零許容範囲
 MARKOWITZTOL 因子化のマルコビット許容範囲
 MIPABSCUTOFF LP最適化コマンドの後のカットオフ集合 (双対のみ)
 OPTIMALITYTOL 被約費用の許容範囲
 PENALTY 最大絶対ペナルティ変数係数
 PERTURB 摂動値
 PIVOTTOL ピボット許容範囲
 Xpress-Optimizer Reference Manual Console and Library Functions 177
XPR\$maxim / XPR\$minim

MAXIM / MINIM 6

例 1 (ライブラリ)

```
XPRsmaxim(prob, "b");
```

ニュートンバリア法を使用して現在の問題を最大化します。

例 2 (コンソール)

```
MINIM -g
```

ニュートンバリア法を使用して現在の問題を最小化し、整数解探索を実施します。

補足

1. 最適化に使用されるアルゴリズムはDEFAULTALG制御で決まります。デフォルトでは、双対シンプレックス法がLPとMIP問題に使用され、ニュートンバリア法がQP問題に使用されます。
2. 双対シンプレックス法は双対実行不能を除去できる二段階アルゴリズムです。
3. 解が見つかった場合、デフォルトでは解ファイルに解が書きこまれます。これにはいくらか時間がかかるので、XPRsmaxim (MAXIM) やXPRsminim (MINIM)を呼ぶ前に SOLUTIONFILE制御を0に設定し、この機能ははずすよう設定することをお勧めします。
4. (コンソール) 最適化処理をCTRL+L入力で中断したときには、繰返しは最初の「安全な」時点まで実施して終了し、そのときの解が解ファイルに保存されます。

関連事項

XPRsglobal (GLOBAL), XPRsrecurse (RECURSE), XPRsreadbasis

(READBASIS), XPRsgoal (GOAL), 4章「アルゴリズムの性能」

PPFACTOR 部分プライシング候補リストサイズパラメータ

RELPIVOTTOL 相対ピボット許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 178

XPRSpivot 6**XPRSpivot****目的**

変数inを基底に入れ、outを取り除くことによって、シンプレックスピボットを実行します。

概要

```
int XPRSpivot(XPRsprob prob, int in, int out);
```

引数**エラー値****関係する制御**

Double

例

7番目の変数を基底に入れ、5番目の変数を除去します。

```
XPRSpivot(prob, 6, 4)
```

補足

行の添字は0からROWS-1の範囲で、列の添字はROWS+SPAREROWSからROWS+SPAREROWS+COLS-1の範囲です。

関連事項

XPRsgetpivotorder, XPRsgetpivots.

prob	現在の問題
in	基底に入れる行, 列の添字
out	基底から出す行, 列の添字
425	inは無効です(範囲外もしくは既に基底)

426

outは無効です（範囲外もしくは不適格。例えば、基底でない、零ピボット等）

PIVOTTOL

ピボット許容範囲

RELPIVOTTOL

相対ピボット許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 179

QUIT 6

QUIT

目的

OSに零の終了コードを返し、コンソールOptimizerを終了します。

概要

QUIT

引数

なし

関係する制御

なし

例 (コンソール)

コマンドは単純で次のように呼ばれます。

QUIT

補足

1. 致命的なエラーのときにはホストOSで使用する非零終了コードを返します。9章「エラーメッセージ&リターンコード」をお読みください。
2. 最終的な解の状態を表す終了コードを返したい場合は、STOPコマンドを代わりにお使い下さい。

関連事項

STOP, XPRSSave (SAVE).

Xpress-Optimizer Reference Manual Console and Library Functions 180

XPRStrange

RANGE 6

XPRStrange RANGE

目的

問題のレンジ情報を計算し、ファイル`problem_name.rng`に保存します。

概要

```
int XPRStrange(XPRSProb prob);
```

RANGE

引数

関係する制御

なし

例 1 (ライブラリ)

最適化に続いてレンジ情報を計算し、解をファイル`leonor.rrt`に書き出します。

```
XPRsreadprob(prob, "leonor", "");
```

```
XPRsmaxim(prob, "");
```

```
XPRStrange(prob);
```

```
XPRswriteprtrange(prob);
```

例 2 (コンソール)

出力をファイルの代わりに画面に送る以外は、コンソールと同じです。

```
READPROB leonor
MAXIM
RANGE
PRINTRANGE
```

補足

1. 問題の最適基底解が使える状態でなくてはなりません。つまり、XPRsmaxim(MAXIM) , XPRsminim (MINIM)を呼んだ後でなくてはなりません (またはニュートンバリア法でクロスオーバーを用いた後)
2. XPRsrange (RANGE)により計算して得られた情報によって、問題の細かい事後分析ができます。特に右辺の値を動かしても最適基底の変わらない範囲、シャドウプライスがそのままの範囲、変化を制限しているアクティビティを知ることができます。

prob 現在の問題
Xpress-Optimizer Reference Manual Console and Library Functions 181
XPRsrange

RANGE 6

計算された値を得る関数 XPRsgetcolrange , XPRsgetrowrange , XPRswriteprtrange (WRITEPRTRANGE) そして/または XPRswriterange (WRITERANGE) をご覧下さい。

3. MIP問題のレンジはできません。離散要素はXPRsfixglobal (FIXGLOBAL)を最初に使って固定され、残されたLPが解かれます。XPRsfixglobal (FIXGLOBAL)をご覧下さい。

関連事項

XPRsgetcolrange, XPRsgetrowrange, XPRswriteprtrange (WRITEPRTRANGE), XPRswriterange (WRITERANGE).
Xpress-Optimizer Reference Manual Console and Library Functions 182
XPRsreadbasis

READBASIS 6**XPRsreadbasis READBASIS****目的**

ファイルから、前に保存した基底を読みこむようにOptimizerに指示します。

概要

```
int XPRsreadbasis(XPRsprob prob, char *filename,
char *flags);
READBASIS [-flags] [filename]
```

引数**関係する制御**

なし

例 1 (ライブラリ)

現在の問題で更により基底が利用できるのであれば、Optimizerの入力は次のようになります。

```
XPRsreadprob(prob, "filename", "");
XPRsreadbasis(prob, "", "");
XPRsmaxim(prob, "g");
```

行列ファイルを読み、更により開始となる基底を入力してMIPを最大化します。

例 2 (コンソール)

コンソールユーザの場合、同じコマンド集合で次のようになります。

```

READPROB
READBASIS
MAXIM -g
prob          現在の問題
filename      基底が読まれるファイル名を含む200文字以内の文字列。省略時は
              problem_nameに拡張子.bssをつけた形がデフォルトです
flags        XPRSreadbasis (READBASIS) に引き渡すフラグ
              t 基底のコンパクト形式の入力
              基底の行と列の数が現在の行と列の数と一致するかどうかだけがチェ
                ックされます

```

Xpress-Optimizer Reference Manual Console and Library Functions 183

XPRSreadbasis

READBASIS 6

補足

行列が事前分析済みでも、XPRSreadbasis (READBASIS)は、原問題の基底を読みこみます。Optimizerは基底を読みこみ、有効であるかチェックをし、矛盾があればエラーメッセージを表示します。

関連事項

XPRSloadbasis, XPRSwritebasis (WRITEBASIS).

Xpress-Optimizer Reference Manual Console and Library Functions 184

XPRSreaddir

READDIRS 6

XPRSreaddir READDIRS

目的

整数解探索のためのディレクティブファイルを読みます。

概要

```

int XPRSreaddir(XPRSprob prob, char *filename);
READDIRS [filename]

```

引数

関係する制御

Double

例 1 (ライブラリ)

問題steveで使うためにファイルsue.dirから指示を読みます。

```

XPRSreadprob(prob, "steve", "");
XPRSreaddir(prob, "sue");
XPRSminim(prob, "g");

```

例 2 (コンソール)

```

READPROB
READDIRS
MINIM -g

```

これは、コンソールでの最も普通の形です。現在の問題名に拡張子.dirが付いたディレクティブファイルを読みます。

補足

1. モデルを事前分析した後では指示を読むことはできません。PRESOLVEが0に設定されていて事前分析ができないのでない限り、このコマンドはXPRSminim (MINIM) や

	XPRSmaxim (MAXIM)の前に実施されなくてはなりません。
prob	現在の問題
filename	指示が読まれるファイル名を含む200文字以内の文字列。省略(またはNULL)時、デフォルトでは <code>problem_name</code> に拡張子 <code>.dir</code> をつけたものになります
PSEUDOCOST	ノード劣化推定量のデフォルト擬コスト

Xpress-Optimizer Reference Manual Console and Library Functions 185
XPRSreaddir

READDIRS 6

- 優先順位, 分枝方向, 擬コスト, モデルカットに関する指示が得られます。各離散要素には優先順位が付いています。値が小さい方が、要素は分枝に使用される可能性が高く、値が大きければ低くなります。デフォルトでは、全離散要素の優先順位値は500で、この値はディレクティブファイルの優先順位入力で変更できます。一般に、要素の優先度をモデルでの重要性に合わせておくことが有効です。1000を超える順位値は無効で無視されません。ディレクティブファイルの詳細については、付録A.5「ディレクティブファイル(.dir)」をご覧ください。
- デフォルトでは、XPRSglobal (GLOBAL)は、各ノードから分枝要素を押し上げたものと押し下げたもののうち、よりよい整数解を出すと推定された方を探索します。ディレクティブファイルにUPまたはDNを指定すると、XPRSglobal (GLOBAL)は指定された要素を先に押し上げ/下げて探索します。
- 擬コストは要素を押し上げる、あるいは押し下げるときの単位費用の推定値です。デフォルトでは、XPRSglobal (GLOBAL)は双対変数値から推定値を計算し、PSEUDOCOST制御で設定されたデフォルト擬コストに加えます。デフォルト擬コストは、ディレクティブファイルのPUやPD要素で置きかえられます。
- モデルカットが使用される場合、指定された制約条件は行列から除去されOptimizerのカットプールに加えられます。整数解探索のあるノードのLP解により違反されるときにだけ、行列に返されます。
- ディレクティブファイルをマニュアルで作成する場合、ワイルドカードが複数のベクトルを同時に指定できません。例えば、PR x1* 2により、優先度が2で名前がx1 から始まる離散要素が与えられます。

関連事項

XPRSloaddirs, 付録 A.5「ディレクティブファイル(.dir)」
 Xpress-Optimizer Reference Manual Console and Library Functions 186
XPRSreadprob

READPROB 6

XPRSreadprob READPROB

目的

ファイルから(X)MPS形式あるいはLP形式の行列を読みます。

概要

```
int XPRSreadprob(XPRSprob prob, char *probname,
char *flags);
READPROB [-flags] [probname]
```

引数

関係する制御

Integer

prob

現在の問題

probname

probname ファイル名で問題が読み込まれる200文字以内の文字列。省略時は(コンソールユーザのみ)、デフォルトで *problem_name* に拡張子がついたものになります。以下を参照ください

flags

引き渡すフラグ

l probname.lp だけを探索します

b mp モデルバイナリインタフェースファイル形式の問題を読みこみます。mp モデルのBIFGENERATEコマンドで生成されたファイルprobname.bifとprobname.solが必要です

EXTRAELEMS

追加できる行列の要素の数

EXTRACOLS

追加できる列の数

EXTRAMIPENTS

追加できる離散要素の数

EXTRAPRESOLVE

事前分析において追加できる要素の数

EXTRAROWS

追加できる行の数

KEEPNROWS

非束縛行の状態

MPSECHO

MPSコメントのエコーの設定

MPSERRIGNORE

無視する小さなエラーの数

MPSFORMAT

MPSファイルの形式の指定

MPSNAMELENGTH

最大の名前文字長さ

SCALING

スケーリングのタイプ

Xpress-Optimizer Reference Manual Console and Library Functions 187

XPRsreadprob

READPROB 6

Double

String

例 1 (ライブラリ)

```
XPRsreadprob(prob, "myprob", "");
```

myprob.mat, myprob.mps, (LP形式の) myprob.lpで最初に見つかったファイルからMPS形式の行列をOptimizerに読みこむように指示をします。

例 2 (コンソール)

```
READPROB -l
```

OptimizerにLP形式の行列をファイル*problem_name.lp*から読みこむように指示をします。

補足

1. フラグがない場合、.mat, .mps, .lpの順番でファイルタイプを探索します。拡張子が.lp(つまりLPファイル)でない場合は、行列ファイルは、XMPS形式 あるいは MPS形式と推定されます。
2. probnameが指定されたときは、問題名は拡張子を無視してprobnameにされます。
3. 文字パラメータMPSOBJNAMEが設定されていない限り、XPRsreadprob (READPROB)は行列で最初のNタイプの行を目的関数として採用します。この場合、探された目的関数行はMPSOBJNAMEによって名付けられます。同様に、空白でない場合、文字列パラメータMPSRHSNAME, MPSBOUNDNAME, MPSRANGENAMEは、右辺, 上下界値, レンジ集合を指

定めます。例えば次の通りです。

```
MPSOBJNAME='Cost ' ! 8文字全部を書く必要があることに注意下さい。
MPSRHSNAME='RHS 1' !
READPROB
```

目的関数以外のNタイプの行の扱いは、KEEPNROWS制御の設定によります。KEEPNROWSが1の場合、行や要素はメモリに取っておきます。0の場合、行は保持されますが、要素は取り除かれます。-1の場合、行は完全に削除されます。最適化の前にPRESOLVEに0を設定して事前分析を無効にしておかない限り、このようなNタイプの行を保持することの影響は小さいと思われま

MATRIXTOL	行列要素の零許容範囲
SOSREFTOL	参照行要素の最小ギャップ
MPSBOUNDNAME	アクティブ上下界値名
MPSOBJNAME	目的関数行の名前
MPSRANGENAME	範囲の名前
MPSRHSNAME	右辺の名前

Xpress-Optimizer Reference Manual Console and Library Functions 188
XPRSreadprob

READPROB 6

- Optimizerは行列ファイルが形式的に問題がないかをチェックし、エラーが見つければエラーメッセージを出します。Optimizerは問題を読んでチェックが終わると、問題の概略を示します。
- デフォルトでは、MPSFORMAT制御に-1が設定されると、XPRSreadprob(READPROB)はMPSファイルがフリー形式か固定形式であるか自動的に決定します。MPSFORMATに0が設定された場合は、固定形式とされ、1に設定された場合は、フリー形式とされます。フリー形式のMPSファイルのフィールドは、1つ以上の空白で区切られます。キーワードNAME, ROWS, COLUMNS, QUADOBJ, SETS, RHS, RANGES, BOUNDS は、列で始まり、ベクトル名は空白を含みません。特殊順序集合が参照行と指定される場合、その名前は行名と同じでないかもしれません。MPSFORMATが0に設定されない限り、スペースを含む数値が(例えばマイナス符号の後)、正しく読めないことに注意してください。
- 問題を自動的にスケーリングしない場合は、XPRSreadprob (READPROB)コマンドの前にパラメータSCALINGを0に設定してください。
- 長いMPSベクトル名は、MPSファイル, LPファイル, ディレクティブファイル, 基底ファイルではサポートされています。MPSNAMELENGTH制御は、MPSベクトル名の最大長を指定し、ファイルが読みこみされる前に設定されなくてはなりません。内部では、最小の8の倍数で丸められ、64を超えません。

関連事項

XPRSloadglobal, XPRSloadlp, XPRSloadqglobal, XPRSloadqp.
Xpress-Optimizer Reference Manual Console and Library Functions 189
XPRSrecurse

RECURSE 6

XPRSrecurse RECURSE

目的

ある種の非線形計画問題 (NLP) を解くための、逐次線形計画法や反復を実行します。

概要

```
int XPRSrecurse(XPRSprob prob);
RECURSE
```

引数**関係する制御**

Integer

Double

例 1 (ライブラリ)

反復を実行する前に、最大ループ回数を10回に設定し、最後に収束しなかった項の数を返します。

```
int reconverge;
..
XPRSreadprob(prob, "myprob", "");
XPRSmaxim(prob, "");
XPRSsetintcontrol(prob, XPRS_RECMAXPASSES, 10);
XPRSrecurse(prob);
XPRSgetintattrib(prob, XPRS_RECCONVERGE, &reconverge);
printf("RECCONVERGE = %d\n", reconverge);
```

prob	現在の問題
CPUTIME	1 はCPU時間 0は経過時間
RECMAXPASSES	最大反復回数
RECEXPAND	BX係数とSB上下界値の増加割合
RECSHRINK	BX係数とSB上下界値の減少割合
RECSTEPLength	反復ステップの長さ
RECSTOP	反復収束パラメータ

Xpress-Optimizer Reference Manual Console and Library Functions 190

XPRSrecurse**RECURSE 6****例 2 (コンソール)**

```
KEEPNROWS = 1
READPROB
MINIM
RECSHRINK = 0.8
RECEXPAND = 1.6
RECURSE
QUIT
```

補足

事前分析機能が反復と使われると、行列は各反復の最初に事前分析され、最後に事後分析されます。事前分析が再最適化時間に比べて長ければ、事前分析をしないことをお勧めします。ニュートンバリア法を使う場合は、XPRSrecurse(RECURSE)を使う前にクロスオーバーされなくてはなりません。

関連事項

XPRSmaxim (MAXIM) / XPRSminim (MINIM).

Xpress-Optimizer Reference Manual Console and Library Functions 191

XPRSrestore

RESTORE 6

XPRSrestore RESTORE

目的

Optimizerのデータ構造をXPRSsave(SAVE)で作成したファイルから保存します。最適化は、そのファイルが作成された時点から再開します。

概要

```
int XPRSrestore(XPRSprob prob, char *probname);
RESTORE
```

引数

関係する制御

なし

例 1 (ライブラリ)

```
XPRSrestore(prob, "")
```

例 2 (コンソール)

```
RESTORE
```

補足

- このルーチンは、以前にXPRSsave (SAVE)で作成したファイル *problem_name.svf* からデータ構造を戻すのに使います。ファイル *problem_name.sol* も必要ですが、整数解探索の最適化を再開する場合、ファイル *problem_name.glb* と *problem_name.ctp* も必要となります。*.svf* ファイルは、それらを作成するのに使われるOptimizerのリリース版用に特化されています。それらを作る際に使われたのと同じリリース版Optimizerでのみ読みこみ可能です。
- (コンソール) XPRSsave (SAVE) と XPRSrestore (RESTORE) の主な使い道は、長い最適化の実行をCTRL+Cで中断した際、Optimizerの状態を保存しておいて、後でその時点から再開できるようにすることです。問題の最適状態を保存しておき、問題にXPRSalter (ALTER)を実行し、保存された状態から最適化を再開するという方法もあります。

```
prob                現在の問題
probname            問題名を含む200文字以内の文字列
Xpress-Optimizer Reference Manual Console and Library Functions 192
XPRSrestore
```

RESTORE 6

関連事項

XPRSalter (ALTER), XPRSsave (SAVE).
Xpress-Optimizer Reference Manual Console and Library Functions 193
XPRSsave

SAVE 6

XPRSsave SAVE

目的

現在のデータ構造、つまり行列や、ファイルに設定されている制御や問題特性を保存して、実行を終了します。最適化は後で再開できます。

概要

```
int XPRSsave(XPRSprob prob);
SAVE
```


引数**関係する制御**

なし

例 1 (ライブラリ)

```
XPRSSave(prob);
```

例 2 (コンソール)

```
SAVE
```

補足

データ構造はファイル `problem_name.svf` に書かれます。最適化は、`XPRRestore (RESTORE)`によりデータ構造が戻されたときと同じポイントから再開できます。このような状況では、ファイル `problem_name.sol`と、分枝限定法が進行中であれば、整数ファイル `problem_name.glb`と `problem_name.ctp`も必要とされます。これらのファイルは、`XPRSSave (SAVE)`の実行後には存在しますが、その後の最適化で変更されます。したがって、`XPRSSave (SAVE)`を呼び出した後は、最適化を呼びださない法がよいかもしれません。作成される `.svf`ファイルは、作成するのに使われたリリース版のOptimizerにより作成されます。それらを作る際に使われたのと同じリリース版のOptimizerを使ってのみ読み込みます。

関連事項

`XPRRestore (RESTORE)`.

`prob` 現在の問題
 Xpress-Optimizer Reference Manual Console and Library Functions 194
XPRSScale

SCALE 6**XPRSScale SCALE****目的**

現在の行列を再スケールします。

概要

```
int XPRSScale(XPRSProb prob, int *mrscal, int *mcscal);  
SCALE
```

引数**関係する制御**

Integer

例 1 (ライブラリ)

```
XPRSScale(prob, "jovial", "");  
XPRSSalter(prob, "serious");  
XPRSScale(prob, NULL, NULL);  
XPRSSminim(prob, "");
```

MPSファイル `jovial.mat`を読みこみ、ファイル `serious.alt`の指示によって変更し、行列を再スケールして最小目的関数値を求めます。

例 2 (コンソール)

コンソールで同様のことを実行するコマンドは次の通りです。

```
READPROB jovial  
ALTER serious  
SCALE  
MINIM
```

`prob` 現在の問題

<code>mrscal</code>	行をスケールする2の幅を含む長さROWSの整数型配列。必要なければNULLとなります
<code>mcscale</code>	列をスケールする2の幅を含む長さCOLSの整数型配列。必要なければNULLとなります
SCALING	スケールのタイプ

Xpress-Optimizer Reference Manual Console and Library Functions 195
XPRSScale

SCALE 6

補足

1. `mrscal`と`mcscale`が共にNULLでない場合、行列をスケールするのに使用されます。そうでなければ、行列はSCALING制御にしたがってスケールされます。現在の行列が `XPRSSalter` (ALTER), `XPRSSchgmcoef`, `XPRSSaddrows`の様なルーチン呼びだして変更済みの場合、このルーチンは有用です。
2. 現在の行列が事前分析済みの場合、`XPRSSscale` (SCALE)を呼びだせません。

関連事項

`XPRSSalter` (ALTER), `XPRSSreadprob` (READPROB).
 Xpress-Optimizer Reference Manual Console and Library Functions 196

XPRSSetcbbarlog 6

XPRSSetcbbarlog

目的

内点法における各反復において呼ばれるバリアーログコールバック関数を宣言します。

概要

```
int XPRSSetcbbarlog(XPRSSprob prob,
int (XPRSS_CC *fub1)(XPRSSprob my_prob, void *my_object),
void *object);
```

引数

関係する制御

なし

例

アルゴリズムの各反復ごと画面に1行出力する単純な例です。

```
XPRSSetcbbarlog(prob, barLog, NULL);
XPRSSmaxim(prob, "b");
```

このコールバック関数は以下に似ています。

```
int XPRSS_CC barLog(XPRSSprob prob, void *object)
{
printf("Next barrier iteration\n");
}
```

`prob`

現在の問題

`fub1`

コールバック関数です。2つの引数`my_prob`と`my_object`があり、整数の戻り値を持ちます。`fub1`により返される値が零でない場合、求解は中断されます。この関数は、バリアの繰返しの度に呼ばれます

`my_prob`

コールバック関数`fub1`に渡される問題

`my_object`

`XPRSSetcbbarlog`でコールバックを設定したとき、オブジェクトとして渡されるユーザ定義オブジェクト

object コールバック関数 `fucb1` に渡すユーザ定義オブジェクト
Xpress-Optimizer Reference Manual Console and Library Functions 197

XPRSsetcbbarlog 6

補足

コールバック関数が零でない値を返した場合、Optimizerは中断します。

関連事項

XPRSsetcbgloballog, XPRSsetcbpllog, XPRSsetcbmessage.
Xpress-Optimizer Reference Manual Console and Library Functions 198

XPRSsetcbchgbranch 6

XPRSsetcbchgbranch

目的

MIP探索中に新しく分枝変数が設定あるいは選択される度に呼ばれる分枝変数コールバック関数を宣言します。

概要

```
int XPRSsetcbchgbranch(XPRSprob prob,
void (XPRS_CC *fucb)(XPRSprob my_prob, void *my_object,
int *entity, int *up, double *estdeg), void *object);
```

引数

関係する制御

なし

prob	現在の問題
fucb	5つの引数 <code>my_prob</code> , <code>my_object</code> , <code>entity</code> , <code>up</code> , <code>estdeg</code> を持ち、戻り値がないコールバック関数です。この関数は新しく分枝変数や集合選択される度に呼ばれます
my_prob	コールバック関数 <code>fucb</code> に渡される問題
my_object	XPRSsetcbchgbranchでコールバックを設定したとき、 <code>object</code> として渡されるユーザ定義オブジェクト
entity	分枝方向を示す変数や集合を示すポインタ。通常、離散変数は列添字、例えば <code>0, 1, ..., (COLS-1)</code> や集合添字、例えば <code>0, 1, ..., (SEETS-1)</code> により区別されます
up	<code>entity</code> が変数の場合、最初に上方へ分枝をするのであれば1、そうでなければ0とします。 <code>entity</code> が設定されている場合、最初に上方へ分枝するのであれば3、そうでなければ2とします
estdeg	ノードでの推定劣化量
object	コールバック関数 <code>fucb</code> に渡されるユーザ定義オブジェクト

Xpress-Optimizer Reference Manual Console and Library Functions 199

XPRSsetcbchgbranch 6

例

整数探索において最も違反が大きなバイナリ整数に分枝ルールを適用します。

```
typedef struct {
double* soln;
char* type;
double tol;
int cols;
```

```

} solutionData;
...
solutionData nodeData;
...
XPRSsminim(prob, "");
XPRSssetintcontrol(prob, XPRS_MIPLOG, 3);
XPRSssetintcontrol(prob, XPRS_CUTSTRATEGY, 0);
/* setup data */
XPRSsgetintattrib(prob, XPRS_COLS, &(nodeData.cols));
XPRSsgetdblcontrol(prob, XPRS_MATRIXTOL, &(nodeData.tol));
nodeData.soln = (double*)
malloc(sizeof(double)*nodeData.cols);
nodeData.type = (char*)
malloc(sizeof(char)*nodeData.cols);
XPRSsgetcoltype(prob, nodeData.type, 0, nodeData.cols-1);
XPRSssetcbchbranch(prob, varSelection, &nodeData);
XPRSsglobal(prob);

```

コールバック関数は次と似ています。

```

void XPRS_CC varSelection(XPRSprob prob, void* vdata,
int *iColumn, int *iUp, double *dEstimate )
{
double dDist, dUpDist, dDownDist, dGreatestDist=0;
int iCol;
solutionData *nodeData = (solutionData*) vdata;
XPRSsgetpresolvesol(prob, (*nodeData).soln, NULL, NULL,
NULL);
for(iCol=0; iCol<(*nodeData).cols; iCol++)
if((*nodeData).type[iCol]=='I' ||
(*nodeData).type[iCol]=='B')
{
Xpress-Optimizer Reference Manual Console and Library Functions 200

```

XPRSssetcbchbranch 6

```

dUpDist=ceil((*nodeData).soln[iCol]) -
(*nodeData).soln[iCol];
dDownDist = (*nodeData).soln[iCol] -
floor((*nodeData).soln[iCol]);
dDist = (dUpDist>dDownDist)?dUpDist:dDownDist;
if(dDownDist > (*nodeData).tol &&
dUpDist > (*nodeData).tol)
if( dDist > dGreatestDist )
{
*iColumn = iCol;
dGreatestDist = dDist;
}
}
}
}

```

補足

引数は分枝変数, 分枝方向, 推定劣化量のデフォルト値を初期値として含みます。それらが変更されると、Optimizerは新しい値を使います。変更なければデフォルトのまま使用されます。

関連事項

XPRSssetcbchgnode, XPRSssetcboptnode, XPRSssetcbinfnode,

XPRSsetcbintsol, XPRSsetcbnodecutoff, XPRSsetcbprenode.
Xpress-Optimizer Reference Manual Console and Library Functions 201

XPRSsetcbchgnode 6

XPRSsetcbchgnode

目的

ノード選択コールバック関数を宣言します。この関数は、MIP探索中に、新しいノードを選択するためにコードを遡る度に呼ばれます。

概要

```
int XPRSsetcbchgnode(XPRSprob prob,
void (XPRS_CC *fusn)(XPRSprob my_prob, void *my_object,
int *nodnum), void *object);
```

引数

関係する制御

Integer

例 1

整数解探索中、新しいノードが選ばれる度にノード番号を出力します。

```
XPRsminim(prob, "");
XPRSsetintcontrol(prob, XPRS_MIPLOG, 3);
XPRSsetintcontrol(prob, XPRS_NODESELECTION, 2);
XPRSsetcbchgnode(prob, nodeSelection, NULL);
XPRsglobal(prob);
```

prob	現在の問題
fusn	3つの引数my_prob, my_object, nodnumを持ち、戻り値のないコールバック関数です。この関数は新しいノードが選択される度に呼ばれます
my_prob	コールバック関数fusnに渡される問題
my_object	XPRSsetcbchgnodeでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
nodnum	Optimizerにより選ばれたノードnodnumの数を示すポインタ。この引数により示される値を変えると、選ばれたノードがこの関数に基づいて変更されます
object	コールバック関数fusnに渡されるユーザ定義オブジェクト
NODESELECTION	ノード選択制御

Xpress-Optimizer Reference Manual Console and Library Functions 202

XPRSsetcbchgnode 6

このコールバック関数は以下と同様です。

```
XPRS_CC void nodeSelection(XPRSprob prob, void *object,
int *Node)
{
printf("Node number %d\n", *Node);
}
```

例 2

Xpress-MP CD-ROMの例 depthfirst.c をご覧下さい。

関連事項

XPRSsetcboptnode, XPRSsetcbinfnode, XPRSsetcbintsol,

XPRSsetcbnodecutoff, XPRSsetcbchgbranch, XPRSsetcbprenode.
Xpress-Optimizer Reference Manual Console and Library Functions 203

XPRSsetcbcutlog 6

XPRSsetcbcutlog

目的

カットログが出力される度に呼ばれるカットログコールバック関数を宣言します。

概要

```
int XPRSsetcbcutlog(XPRSprob prob,
int (XPRS_CC *fuc1)(XPRSprob my_prob, void *my_object),
void *object);
```

引数

関係する制御

なし

関連事項

XPRSsetcbcutmgr, XPRSsetcbfreecutmgr, XPRSsetcbinitcutmgr.

prob	現在の問題
fuc1	2つの引数my_probとmy_objectを持ち、整数の返り値があるコールバック関数
my_prob	コールバック関数fuc1に渡される問題
my_object	XPRSsetcbcutlogでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
object	コールバック関数fuc1に引き渡されるユーザ定義オブジェクト

Xpress-Optimizer Reference Manual Console and Library Functions 204

XPRSsetcbcutmgr 6

XPRSsetcbcutmgr

目的

分枝限定探索の各ノードで呼ばれるユーザ定義カットマネージャルーチンを宣言します。

概要

```
int XPRSsetcbcutmgr(XPRSprob prob,
int (XPRS_CC *fcme)(XPRSprob my_prob, void *my_object),
void *object);
```

引数

関係する制御

Integer

補足

1. 最も効果的に実行するために、値がわかっているなら、スペース割り当て制御EXTRAROWS, EXTRAELEMENTS, CPMAXELEMENTS, CPMAXCUTSをユーザにより指定すべきです。不明ならば、自動的にリサイズしますが、スペースはユーザ設定で必要な分以上に大きく割り当てられてしまいます。
2. カットマネージャルーチンは、戻り値が0になるまで、各ノードで繰り返し呼ばれます。カットが追加、削除されると、部分問題は自動的に最適化されます。

prob	現在の問題
fcme	2つの引数my_probとmy_objectがあり、整数返り値があるコールバック関数。この関数は分枝限定探索の各ノードで呼ばれます

my_prob	コールバック関数fcmelに渡される問題
my_object	XPRSsetcbcutmgrでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
object	コールバック関数fcmelに渡されるユーザ定義オブジェクト
CPKEEPALLCUTS	アクティブでないユーザ作成カットの除去
CPMAXCUTS	カットプール内のカット最大数
CPMAXELEMS	カットプール内の非零要素の最大数
EXTRAELEMS	追加できる行列要素数
EXTRAROWS	追加できる行数

Xpress-Optimizer Reference Manual Console and Library Functions 205

XPRSsetcbcutmgr 6

3. Xpress-Optimizerではノードに追加されたカットは、子ノードで必ず自動的に復元されます。このために、全てのカットはカットプールに保存され、Optimizerはノードごとにカットプールからカットを復元します

関連事項

XPRSsetcbcutlog, XPRSsetcbfreecutmgr, XPRSsetcbinitcutmgr.
Xpress-Optimizer Reference Manual Console and Library Functions 206

XPRSsetcbestimate 6

XPRSsetcbestimate

目的

推定コールバック関数を宣言します。定義されると、ユーザの離散要素の分枝から推定劣化量を決定するために、分枝限定木のノードごとで呼ばれます。

概要

```
void XPRSsetcbestimate(XPRSprob prob,
int (XPRS_CC *fbe)(XPRSprob my_prob, void *my_object,
int *iglsel, int *iprio, double *degbest,
double *degworst, double *curval, int *ifupx,
int *nglinf, double *degsum, int *nbr), void *object);
```

引数

prob	現在の問題
fbe	7つの引数my_prob, my_object, iglsel, iprio, degbest, degworst, curval, ifupx, nglinf, degsum, nbrを持ち、整数返り値があるコールバック関数。この関数は分枝限定探索の各ノードで呼ばれます
my_prob	コールバック関数fbelに渡される問題
my_object	XPRSsetcbestimateでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
iglsel	選択されたユーザ離散要素(非負であること)
iprio	選ばれたユーザ離散要素の優先度
degbest	望ましい方向で選ばれたユーザ要素の分枝から得られる推定劣化量
degworst	最悪の方向で選ばれたユーザ要素の分枝から得られる推定劣化量
curval	ユーザ離散要素の現在値
ifupx	ユーザ離散要素(0,...,nbr-1)の望ましい分枝
nglinf	実行不可能なユーザ離散要素の数

degsum 全てのユーザ要素を満足する推定劣化量の和
 nbr 分枝の数
 Xpress-Optimizer Reference Manual Console and Library Functions 207

XPRSsetcbestimate 6

関係する制御

なし

関連事項

XPRSbranchcut, XPRSsetcbsepnod, XPRSstorecuts.

object コールバック関数fbcに渡されるユーザ定義オブジェクト
 Xpress-Optimizer Reference Manual Console and Library Functions 208

XPRSsetcbfreecutmgr 6

XPRSsetcbfreecutmgr

目的

分枝限定探索の終わりに呼ばれるユーザ終了ルーチンを宣言します。

概要

```
int XPRSsetcbfreecutmgr(XPRSprob prob,
int (XPRS_CC *fcms)(XPRSprob my_prob, void *my_object),
void *object);
```

引数

関係する制御

なし

関連事項

XPRSsetcbcutlog, XPRSsetcbcutmgr, XPRSsetcbinitcutmgr.

prob	現在の問題
fcms	2つの引数my_probとmy_objectを持ち、整数返り値があるコールバック関数
my_prob	コールバック関数fcmsに渡される問題
my_object	XPRSsetcbfreecutmgrでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
object	コールバック関数fcmsに渡されるユーザ定義オブジェクト

Xpress-Optimizer Reference Manual Console and Library Functions 209

XPRSsetcbgloballog 6

XPRSsetcbgloballog

目的

離散ログが出力される度に呼ばれる離散ログコールバック関数を宣言します。

概要

```
int XPRSsetcbgloballog(XPRSprob prob,
int (XPRS_CC *fugl)(XPRSprob my_prob, void *my_object),
void *object);
```

引数

関係する制御

Integer

例 1

整数解探索の各ノードで、ノード番号とその深さを出力します。

```
XPRSsetintcontrol(prob, XPRS_MIPLOG, 3);
XPRSsetcbgloballog(prob, globalLog, NULL);
XPRSsminim(prob, "g");
```

このコールバック関数は以下と同様です。

```
XPRS_CC int globalLog(XPRSprob prob, void *data)
{
  int nodes, nodedepth;
  XPRSgetintattrib(prob, XPRS_NODEDEPTH, &nodedepth);
```

prob	現在の問題
fugl	2つの引数my_probとmy_objectを持ち、整数返り値があるコールバック関数。この関数はMIPLOG制御により決められた離散ログが出力されると呼びだされます
my_prob	コールバック関数fuglに渡される問題
my_object	XPRSsetcbgloballogでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
object	コールバック関数fuglに渡されるユーザ定義オブジェクト
MIPLOG	整数印刷フラグ

Xpress-Optimizer Reference Manual Console and Library Functions 210

XPRSsetcbgloballog 6

```
XPRSgetintattrib(prob, XPRS_NODES, &nodes);
printf("Node %d with depth %d has just been
processed\n", nodes, nodedepth);
return 0;
}
```

例 2

Xpress-MP CD-ROMの例 depthfirst.c をご覧下さい。

補足

コールバック関数が零でない値を返した場合、整数解探索は中断されます。コールバック関数が零でない値を返した場合、整数解探索は中断されます。

関連事項

XPRSsetcbbarlog, XPRSsetcbplplog, XPRSsetcbmessage.
Xpress-Optimizer Reference Manual Console and Library Functions 211

XPRSsetcbinfnode 6**XPRSsetcbinfnode****目的**

分枝限定探索中に現在のノードが実行不能と分かった後で呼ばれるユーザ最適ノードコールバック関数を宣言します。

概要

```
int XPRSsetcbinfnode(XPRSprob prob,
void (XPRS_CC *fuin)(XPRSprob my_prob, void *my_object),
void *object);
```

引数**関係する制御**

Integer

例 1

整数探索中に実行不能ノードが見つかったらユーザに通知します。

```
XPRSsetintcontrol(prob,XPRS_NODESELECTION,2);
XPRSsetcbinfnode(prob,nodeInfeasible,NULL);
XPRSsmaxim(prob,"g");
```

このコールバック関数は以下と同様です。

```
void XPRS_CC nodeInfeasible(XPRSprob prob, void *obj)
{
  int node;
  XPRSgetintattrib(prob,XPRS_NODES,&node);
  printf("Node %d infeasible\n",node);
```

prob 現在の問題

fuin 2つの引数my_prob とmy_objectを持ち、戻り値がないコールバック関数。この関数は現在のノードが実行不能と分かった後に呼ばれます

my_prob コールバック関数fuinに渡される問題

my_object XPRSsetcbinfnodeでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト

object コールバック関数fuinに渡されるユーザ定義オブジェクト

NODESELECTION ノード選択制御

Xpress-Optimizer Reference Manual Console and Library Functions 212

XPRSsetcbinfnode 6

```
}
```

例 2

Xpress-MP CD-ROM の例 depthfirst.cをご覧ください。

関連事項

```
XPRSsetcbchgnode, XPRSsetcboptnode, XPRSsetcbintsol,
XPRSsetcbnodecutoff, XPRSsetcbchgbranch, XPRSsetcbprenode.
Xpress-Optimizer Reference Manual Console and Library Functions 213
```

XPRSsetcbinitcutmgr 6**XPRSsetcbinitcutmgr****目的**

カットマネージャーを初期化するために呼ばれるユーザコールバックルーチンを宣言します。

概要

```
int XPRSsetcbinitcutmgr(XPRSprob prob,
int (XPRS_CC *fcmi)(XPRSprob my_prob, void *my_object),
void *object);
```

引数**関係する制御**

なし

関連事項

```
XPRSsetcbcutlog, XPRSsetcbcutmgr, XPRSsetcbfreecutmgr.
```

prob 現在の問題

fcmi 2つの引数my_probとmy_objectを持ち、整数戻り値があるコールバック関数

my_prob	コールバック関数fcmiに引き渡される問題
my_object	XPRSsetcbinitcutmgrでコールバックを設定したとき、objectとして引き渡されるユーザ定義オブジェクト
object	コールバック関数fcmiに引き渡されるユーザ定義オブジェクト

Xpress-Optimizer Reference Manual Console and Library Functions 214

XPRSsetcbintsol 6

XPRSsetcbintsol

目的

分枝限定探索中に整数解が見つかる度に呼ばれるユーザ整数解コールバック関数を宣言します。

概要

```
int XPRSsetcbintsol(XPRSProb prob,
void (XPRS_CC *fuis)(XPRSProb my_prob, void *my_object),
void *object);
```

引数

関係する制御

なし

例 1

整数解探索で見つめられた整数解を解ファイルを使わずに出力します。

```
XPRSsetcbintsol(prob, printsol, NULL);
XPRSsetintcontrol(prob, XPRS_SOLUTIONFILE, 0);
XPRSmxim(prob, "g");
```

このコールバック関数は以下と同様です。

```
void XPRS_CC printsol(XPRSProb my_prob, void *my_object)
{
int i, cols, *x;
double objval;
XPRSgetintattrib(my_prob, XPRS_COLS, &cols);
```

prob	現在の問題
fuis	2つの引数my_probとmy_objectを持ち、戻り値はないコールバック関数。現在のノードが整数実行可能解を持つと分かったとき（つまり整数実行可能解が見つかる度に）呼ばれます

my_prob	コールバック関数fuisに渡される問題
my_object	XPRSsetcbintsolでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
object	コールバック関数fuisに渡されるユーザ定義オブジェクト

Xpress-Optimizer Reference Manual Console and Library Functions 215

XPRSsetcbintsol 6

```
XPRSgetdblattrib(my_prob, XPRS_LPOBJVAL, &objval)
x = malloc(cols * sizeof(int));
XPRSgetsol(my_prob, x, NULL, NULL, NULL);
printf("\nInteger solution found: %f\n", objval);
for(i=0; i<cols; i++) printf(" x[%d] = %d\n", i, x[i])
}
```

例 2

Xpress-MP CD-ROM の例depthfirst.cをご覧ください。

補足

メモリから整数解の値を戻したいときに、この関数は有用です。整数解が見つかった後、他の方法でこの情報を得るには、解ファイルから得るしかありません。

関連事項

XPRSsetcbchgnode, XPRSsetcboptnode, XPRSsetcbinfnode,
XPRSsetcbnodecutoff, XPRSsetcbchgbranch, XPRSsetcbprenode.
Xpress-Optimizer Reference Manual Console and Library Functions 216

XPRSsetcbplog 6**XPRSsetcbplog****目的**

シンプレックス法の各LPLOG繰返しの後に呼ばれるシンプレックスログコールバック関数を宣言します。

概要

```
int XPRSsetcbplog(XPRSprob prob,
int (XPRS_CC *fuil)(XPRSprob my_prob, void *my_object),
void *object);
```

引数**関係する制御**

Integer

例

次のコードはコールバック関数lpLog を最適化における10回の繰返しごとに呼ばれるように設定しています。

```
XPRSsetintcontrol(prob, XPRS_LPLOG, 10);
XPRSsetcbplog(prob, lpLog, NULL);
XPRSreadprob(prob, "problem", "");
XPRSminim(prob, "");
```

prob	現在の問題
fuil	2つの引数my_probとmy_objectを持ち、整数返り値があるコールバック関数。この関数は、繰返し0と最後の繰返しを含む各LPLOGシンプレックス繰返しの度に呼ばれます
my_prob	コールバック関数fuilに渡される問題
my_object	XPRSsetcbplogでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
object	コールバック関数fuilに渡されるユーザ定義オブジェクト
LPLOG	シンプレックスアルゴリズムの繰返しログを出力する頻度

Xpress-Optimizer Reference Manual Console and Library Functions 217

XPRSsetcbplog 6

このコールバックは以下と同様です。

```
int XPRS_CC lpLog(XPRSprob my_prob, void *my_object)
{
int iter; double obj;
XPRSgetintattrib(my_prob, XPRS_SIMPLEXITER, &iter);
XPRSgetdblattrib(my_prob, XPRS_LPOBJVAL, &obj);
printf("At iteration %d objval is %g\n", iter, obj);
return 0;
}
```

補足

コールバック関数が零でない値を返す場合、解の探索は中断されます。

関連事項

XPRSsetcbbarlog, XPRSsetcbgloballog, XPRSsetcbmessage.
Xpress-Optimizer Reference Manual Console and Library Functions 218

XPRSsetcbmessage 6**XPRSsetcbmessage****目的**

Optimizerによりテキストラインが出力される度に呼ばれる出力コールバック関数を宣言します。

概要

```
int XPRSsetcbmessage(XPRSprob prob,
void (XPRS_CC *fop)(XPRSprob my_prob, void *my_object,
const char *msg, int len, int msgtype), void *object);
```

引数**関係する制御***Integer*

prob	現在の問題
fop	5つの引数my_prob, my_object, msg, len, msgtypeを持ち、返り値のないコールバック関数
my_prob	コールバック関数fopに渡される問題
my_object	XPRSsetcbmessageでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
msg	新しいラインのメッセージを含む文字列
len	メッセージの長さ
msgtype	出力メッセージタイプを示します 1 正常メッセージ (情報) 2 正常メッセージ (デバッグ) 3 警告メッセージ 4 エラーメッセージ
object	の値の場合、出力がファイルにリダイレクトされると、部分ルーチンライブラリの呼び出しは終了しようとし、バッファは消去されるかもしれません コールバック関数fopに渡されるユーザ定義オブジェクト
OUTPUTLOG	メッセージの出力

Xpress-Optimizer Reference Manual Console and Library Functions 219

XPRSsetcbmessage 6**例**

全ての出力を単純に画面(stdout)に送ります。

```
XPRSsetcbmessage(prob, Message, NULL);
```

コールバック関数は以下と同様です。

```
void XPRS_CC Message(XPRSprob my_prob, void* my_object,
const char *msg, int len, int msgtype)
{
```

```

switch(msgtype)
{
case 4: /* error */
case 3: /* warning */
case 2: /* dialogue */
case 1: /* information */
printf("%s\n", msg);
break;
default: /* exiting - buffers need flushing */
fflush(stdout);
break;
}
}

```

補足

1. ユーザ出力コールバックを定義することで、全てのスクリーン出力はオフにされます。全ての出力メッセージはOUTPUTLOG整数制御を0にすることで消去されます。
2. この関数はプログラム実行中に発生する警告やエラーを詳細に画面に表示する際に有用です。そうしない場合、関数の戻り値をチェックしたり、XPRSgetlasterrorあるいはログファイルの出力内容を調査することで（そのような設定になっているとします）、同じ情報が得られます。
3. このルーチンは、引数として文字列を返すのはこのコールバックだけであるため、Visual Basic では特別な扱いが必要です。Visual Basic で使用するためには、XPRSetcbmessageVBを使用する必要があります。CD-Romにこの使用例がいくつかあります。

関連事項

XPRSetcbbarlog, XPRSetcbgloballog, XPRSetcbplplog,
XPRSetlogfile.

Xpress-Optimizer Reference Manual Console and Library Functions 220

XPRSetcbnodecutoff 6

XPRSetcbnodecutoff

目的

分枝限定探索中に見つかった改善された整数解により、ノードがカットオフされる度に呼ばれるユーザノードカットオフコールバック関数を宣言します。

概要

```

int XPRSetcbnodecutoff(XPRSProb prob,
void (XPRS_CC *fucn)(XPRSProb my_prob, void *my_object,
int nodnum), void *object);

```

引数

関係する制御

なし

例 1

整数解探索中にノードがカットオフされるとユーザに通知します。

```

XPRSetcbnodecutoff(prob, Cutoff, NULL);
XPRSmxim(prob, "g");

```

このコールバック関数は以下と同様です。

```

void XPRS_CC Cutoff(XPRSProb prob, void *obj, int node);
{

```

```
printf("Node %d cutoff\n",node);
prob          現在の問題
fucn          3つの引数my_prob, my_object, nodnumを持ち、戻り値がない
              コールバック関数。改善された整数解が見つかり、ノードがカット
              オフされる度に呼ばれます
my_prob       コールバック関数fucnに渡される問題
my_object     XPRSsetcbnodecutoffでコールバックを設定したとき、objectと
              して渡すユーザ定義オブジェクト
nodnum        カットオフされるノードの数
object        コールバック関数fucnに渡されるユーザ定義オブジェクト
Xpress-Optimizer Reference Manual Console and Library Functions 221
```

XPRSsetcbnodecutoff 6

```
}
```

例 2

Xpress-MP CD-ROM の例depthfirst.cをご覧ください。

補足

この関数によりユーザは適切なノードの経緯を保存できます。

関連事項

XPRSsetcbchgnode, XPRSsetcboptnode, XPRSsetcbinfnode,
XPRSsetcbintsol, XPRSsetcbchgbranch, XPRSsetcbprenode.
Xpress-Optimizer Reference Manual Console and Library Functions 222

XPRSsetcboptnode 6

XPRSsetcboptnode

目的

分枝限定探索中に現在のノードの最適解を発見した後で呼ばれる最適ノードコールバック関数を宣言します。

概要

```
int XPRSsetcboptnode(XPRSprob prob,
void (XPRS_CC *fuon)(XPRSprob my_prob, void *my_object,
int *feas), void *object);
```

引数

関係する制御

なし

例 1

先に見つかった最適解を出力します。

```
XPRSsetcboptnode(prob,nodeOptimal,NULL);
XPRSmaxim(prob,"g");
```

コールバック関数は以下と同様です。

```
void XPRS_CC nodeOptimal(XPRSprob prob, void *obj,
int *feas)
{
int node;
double objval;
XPRSgetintattrib(prob,XPRS_NODES,&node);
prob          現在の問題
```

fuon	3つの引数、my_prob, my_object, feasを持ち、戻り値がないコールバック関数
my_prob	コールバック関数fuonに引き渡す問題
my_object	XPRSsetcboptnodeでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
feas	実行可能性の状態。零でない値を設定すると、現在のノードは実行不能と宣言されます
object	コールバック関数fuonに渡されるユーザ定義オブジェクト

Xpress-Optimizer Reference Manual Console and Library Functions 223

XPRSsetcboptnode 6

```
printf("NodeOptimal: node number %d\n", node);
XPRSgetdblattrib(prob, XPRS_LPOBJVAL, &objval);
printf("\tObjective function value = %f\n", objval);
}
```

例 2

Xpress-MP CD-ROM の例depthfirst.cをご覧ください。

補足

ノードがこのコールバック関数から実行不能と宣言されると、ノードを最適化するコストは無効になります。

関連事項

XPRSsetcbchgnode, XPRSsetcbinfnode, XPRSsetcbintsol,
XPRSsetcbnodecutoff, XPRSsetcbchgbranch, XPRSsetcbprenode.
Xpress-Optimizer Reference Manual Console and Library Functions 224

XPRSsetcbprenode 6

XPRSsetcbprenode

目的

事前分析ノードコールバック関数を宣言します。ノードが最適化される前に呼ばれるため、ノードの解は使えません。

概要

```
int XPRSsetcbprenode(XPRSprob prob,
void (XPRS_CC *fupn)(XPRSprob my_prob, void *my_object,
int *nodinfeas), void *object);
```

引数

関係する制御

なし

例

各ノードが事前分析される前に、ユーザに通知します。

```
XPRSsetcbprenode(prob, preNode, NULL);
XPRSminim(prob, "g");
```

このコールバック関数は以下と同様です。

```
void XPRS_CC preNode(XPRSprob prob, void* data, int
*Nodinfeas)
{
prob                現在の問題
```


fupn	3つの引数my_prob,my_object , nodnumを持ち、戻り値がないコールバック関数。この関数は、新しいノードが選択される度に呼ばれ、選択されたノードはこれにより変更されます
my_prob	コールバック関数fupnに渡される問題
my_object	XPRSsetcbprenodeでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
nodinfeas	ノードが実行可能かどうかを表します。事前分析中にノードが実行不能と判明した場合、1が設定され、そのノードは再最適化されないことを示します
object	コールバック関数fupnに渡されるユーザ定義オブジェクト

Xpress-Optimizer Reference Manual Console and Library Functions 225

XPRSsetcbprenode 6

```
*Nodinfeas = 0; /* set to 1 if node is infeasible */
}
```

関連事項

XPRSsetcbchgnode, XPRSsetcbinfnode, XPRSsetcbintsol,
XPRSsetcbnodecutoff, XPRSsetcboptnode.
Xpress-Optimizer Reference Manual Console and Library Functions 226

XPRSsetcbsepnode 6

XPRSsetcbsepnode

目的

整数目的関数を使う分枝限定木のノードの分割法を指定するための分割コールバック関数を宣言します。カットまたは上下界値を各ノードに適用してノードは分割されます。これらはカットプールに保存されます。

概要

```
void XPRSsetcbsepnode(XPRSprob prob,
int (XPRS_CC *fse)(XPRSprob my_prob, void *my_object,
int ibr, int iglsel, int ifup, double curval),
void *object);
```

引数

関係する制御

なし

例

整数解探索のためのユーザ分割コールバック関数を定義する前に問題を最小化します。

```
XPRSminim(prob, "");
XPRSsetcbsepnode(prob, nodeSep, NULL);
```

prob	現在の問題
fse	6つの引数my_prob , my_object , ibr , iglsel , ifup , curvalを持ち、整数戻り値があるコールバック関数
my_prob	コールバック関数fseに渡される問題
my_object	XPRSsetcbsepnodeでコールバックを設定したとき、objectとして渡されるユーザ定義オブジェクト
ibr	分枝番号
iglsel	離散要素番号

ifup 離散要素の分枝方向 (ibrと同様)
 curval 離散要素の現在値
 object コールバック関数fselに渡されるユーザ定義オブジェクト
 Xpress-Optimizer Reference Manual Console and Library Functions 227

XPRSsetcbsepnode 6

XPRSGlobal(prob);

関数 nodeSep は次のように定義されます。

```
int nodeSep(XPRSProb my_prob, void *my_object, int ibr,
int iglsel, int ifup, double curval)
{
int index;
double dbd;
if( ifup )
{
dbd = floor(xval);
XPRSstorebounds(my_prob,1,&iglsel,"U",&dbd,&index);
}
else
{
dbd = ceil(xval);
XPRSstorebounds(my_prob,1,&iglsel,"L",&dbd,&index);
}
XPRSbranchcut(my_prob,1,&index);
return 0;
}
```

補足

1. ユーザ分割ルーチンは推定コールバック関数XPRSsetcbestimate により返されるnbr回だけ呼ばれます。これにより、複数の分枝が可能になります
2. XPRSbranchcutを呼ぶことで、ユーザ分割ルーチンにあるノードで指定された上下界値とノあるいはカットに適用されます

関連事項

XPRSbranchcut, XPRSsetcbestimate, XPRSstorecuts.
 Xpress-Optimizer Reference Manual Console and Library Functions 228

XPRSsetdblcontrol 6

XPRSsetdblcontrol

目的

倍精度型制御パラメータの値を設定します。

概要

```
int XPRSsetdblcontrol(XPRSProb prob, int ipar,
double dsval);
```

引数

関係する制御

7章「制御パラメータ」参照

例

倍精度型の制御DEGRADEFACTOR に1.0を設定します。

```
XPRSsetdblcontrol(prob,XPRS_DEGRADEFACTOR,1.0);
```

関連事項

XPRSgetdblcontrol, XPRSsetintcontrol, XPRSsetstrcontrol.

prob 現在の問題
 ipar 設定される値を持つ制御パラメータ。全ての制御パラメータのリストについては7章「制御パラメータ」かヘッダファイルxprs.h のリストをご覧ください
 dsval 制御パラメータに設定される値
 Xpress-Optimizer Reference Manual Console and Library Functions 229

XPRSsetdefaults 6**XPRSsetdefaults****目的**

全ての制御にデフォルト値を設定します。

概要

```
int XPRSsetdefaults(XPRSprob prob);
```

引数**関係する制御**

なし

例

制御にデフォルトを再設定し、再び解く前に事前分析をオフにします。

```
XPRSsetintcontrol(prob, XPRS_PRESOLVE, 0);
XPRSm Maxim(prob, "g");
XPRSwriteprtsol(prob);
XPRSsetdefaults(prob);
XPRSm Maxim(prob, "g");
```

関連事項

7章「制御パラメータ」, XPRSsetintcontrol, XPRSsetdblcontrol, XPRSsetstrcontrol.

prob 現在の問題
 Xpress-Optimizer Reference Manual Console and Library Functions 230

XPRSsetintcontrol 6**XPRSsetintcontrol****目的**

与えられた整数制御パラメータの値を設定します。

概要

```
int XPRSsetintcontrol(XPRSprob prob, int ipar, int isval);
```

引数**関係する制御**

7章「制御パラメータ」参照

例

制御PRESOLVEを0に設定し、最適化の前の事前分析機能をオフにします。

```
XPRSsetintcontrol(prob, XPRS_PRESOLVE, 0);
XPRSm Maxim(prob, "");
```

補足

SCALING のような整数制御パラメータのいくつかはビットマップで、各ビットは異なる動作

を意味します。ビット0は値1、ビット1は値2、ビット2は値4を意味します。

関連事項

XPRSgetintcontrol, XPRSsetdblcontrol, XPRSsetstrcontrol.

prob	現在の問題
ipar	設定される値の制御パラメータ。全制御のリストは7章「制御パラメータ」かヘッダファイルxprs.h内のリストをご覧ください
isval	設定される制御パラメータの値

Xpress-Optimizer Reference Manual Console and Library Functions 231

XPRSsetlogfile 6

XPRSsetlogfile

目的

全てのOptimizer出力をログファイルに書き出すように指示します。

概要

```
int XPRSsetlogfile(XPRSprob prob, char *filename);
```

引数

関係する制御

なし

例

ファイルlogfile.logに出力するよう指示します。

```
XPRSinit(NULL);
XPRScreateprob(&prob);
XPRSsetlogfile(prob, "logfile.log");
```

補足

1. 求解過程でOptimizerにより出力されるエラーや警告を得る手段なので、実行された問題ごとにログファイルを設定することをお勧めします。
2. 出力がXPRSsetlogfileによってリダイレクトされる場合、全ての画面出力はオフになります。
3. 出力コールバックは、XPRSsetcbmessageによって定義され、テキストラインが出力される度に呼ばれます。ユーザ出力コールバックを定義すると、全ての画面出力がオフになります。全ての出力メッセージを破棄するためにはOUTPUTLOG整数制御を0に設定してください。

関連事項

XPRSsetcbmessage.

prob	現在の問題
filename	全ての出力書かれるファイルの名前。NULLを設定すると、出力のリダイレクトが止まり、全てのスクリーン出力がオンに戻ります（常に画面出力をオフにしているDLLユーザを除く）

Xpress-Optimizer Reference Manual Console and Library Functions 232

XPRSsetprobname

SETPROBNAME 6

XPRSsetprobname SETPROBNAME

目的

現在のデフォルト問題名を設定します。あまり使うことはありません。

概要

```
int XPRSssetprobname(XPRSprob prob, char *probname);
SETPROBNAME probname
```

引数**関係する制御**

なし

例 1 (ライブラリ)

現在の問題名をjoに設定します。

```
char sProblem[]="jo";
...
XPRSssetprobname(prob,sProblem);          .....
```

例 2 (コンソール)

```
READPROB bob
MINIM
SETPROBNAME jim
PRINTSOL
SETPROBNAME bob
PRINTSOL
```

問題jimの解を表示し(解ファイルjim.solが存在するとして)、その後で問題bobの解を表示します。

警告

XPRSsreadprob (READPROB) の後、最適化ルーチンを呼ぶ前にXPRSssetprobname (SETPROBNAME)で問題名を変更してはいけません。問題名が変更されると、メモリの行列が保存されません。例えば

```
READPROB number1
SETPROBNAME number2
prob                現在の問題
probname            問題名を含む200文字以内の文字列
Xpress-Optimizer Reference Manual Console and Library Functions 233
```

XPRSssetprobname**SETPROBNAME 6**

```
PRINTSOL
SETPROBNAME number1
MINIM
```

これは、XPRSsreadprob (READPROB)により設定されたメモリ内の行列をPRINTSOLコマンドが上書きするため、不自然な結果になります。

補足

次のようなコマンド列が続くと、XPRSsreadprob (READPROB) はデフォルト問題名を変更します。

```
SETPROBNAME bob
READPROB
```

と

```
READPROB bob
は同じ結果になります。
```

関連事項

XPRSsreadprob (READPROB).

XPRSsetstrcontrol 6**XPRSsetstrcontrol****目的**

与えられた文字列制御パラメータの値を設定します。

概要

```
int XPRSsetstrcontrol(XPRSprob prob, int ipar,
char *csval);
```

引数**関係する制御**

7章「制御パラメータ」参照

例

制御MPSOBJNAME を「Profit」に設定します。

```
XPRSsetstrcontrol(prob, XPRS_MPSOBJNAME, "Profit");
```

関連事項

XPRSgetstrcontrol, XPRSsetdblcontrol, XPRSsetintcontrol.

prob	現在の問題
ipar	値が設定される制御パラメータ。全制御のリストは7章「制御パラメータ」かヘッダファイルxprs.h 内のリストをご覧ください
csval	制御が設定される値を含む文字列(NULL終了分を加える)

STOP 6**STOP****目的**

コンソールOptimizerを終了し、終了コードをOSに返します。バッチ操作でも使用できます。

概要

STOP

引数

なし

関係する制御

なし

例 1 (コンソール)

行列ファイル lama.mat を入力し、整数最適化を実行した後、終了します。

```
READPROB lama
```

```
MAXIM -g
```

```
STOP
```

補足

QUITコマンドと同様にOptimizerを終了します。ホストOSまたは呼び出したプログラムで調べることができる終了値を設定します。

関連事項

QUIT.

XPRSstorebounds 6

XPRSstorebounds

目的

カットプールにある上下界値を保存します。

概要

```
int XPRSstorebounds(XPRSprob prob, int nbnds, int *mcols,
char *qbtype, double *dbds, int *mindex);
```

引数

関係する制御

なし

例 1

整数解探索のためのユーザ分割コールバック関数を定義します。

```
XPRSsetcbsepnod(prob,nodeSep,void);
```

関数nodeSepは次のように定義されます。

```
int nodeSep(XPRSprob prob, void *obj int ibr,
int iglssel, int ifup, double curval)
```

```
{
int index;
double dbd;
if( ifup )
{
dbd = floor(xval);
XPRSstorebounds(prob,1,&iglssel,"U",&dbd,&index);
```

prob

現在の問題

nbnds

保存する上下界値の数

mcols

列添字を含む配列

qbtype

上下界値タイプを含む配列

U 上限を示します

L 下限を示します

dbds

上下界値値を含む配列

mindex

カットプール内の上下界値の添字が返される領域のポインタ

Xpress-Optimizer Reference Manual Console and Library Functions 237

XPRSstorebounds 6

```
}
else
{
dbd = ceil(xval);
XPRSstorebounds(prob,1,&iglssel,"L",&dbd,&index);
}
XPRSbranchcut(prob,1, &index);
return 0;
}
```

関連事項

XPRSbranchcut, XPRSsetcbestimate, XPRSsetcbsepnod.

Xpress-Optimizer Reference Manual Console and Library Functions 238

XPRSstorecuts 6

XPRSstorecuts

目的

カットをカットプールに保存しますが、それは現在のノードには適用されません。これらのカットはアクティブになる前にXPRsloadcutsを使って行列に明示的にロードされなければいけません。

概要

```
int XPRsstorecuts(XPRsprob prob, int ncuts, int nodupl,
int *mtype, char *qrtype, double *drhs, int *mstart,
int *mindex, int *mcols, double *dmatval);
```

引数

prob	現在の問題
ncuts	追加されるカットの数.
nodupl	0 カットプールから重複したものを除きません 1 重複したものはカットプールから除かれます 2 カットタイプを無視して、重複したものはカットプールから除かれます
mtype	カットタイプを含む長さncutsの整数型配列。カットタイプは正の整数でカットを規定するのに使われます
qrtype	行タイプを含む長さncutsの文字型配列 L 行を示します E 行を示します G 行を示します
drhs	カットの右辺の要素を含む長さncuts の倍精度型配列
mstart	各カットの開始を示すmcolsとdmatval配列のオフセットを含む整数型配列。この配列長さはncuts+1で最終要素mstart[ncuts]でncuts+1が始まります
Mindex	カットの添字番号が返される長さncutsの整数型配列。mstartと同じメモリスペースを使用します
mcols	カットの列添字を含む長さmstart[ncuts]-1 の整数型配列
.	.
=	.
.	.

Xpress-Optimizer Reference Manual Console and Library Functions 239

XPRsstorecuts 6

関係する制御

Double

補足

1. XPRsstorecutsは重複カットを減らすのに使われます。パラメータnoduplが1に設定されると、カットプールは追加されるカットと同じカットタイプのカットについて重複を調べます。重複カットが見つかったら、右辺値がカットを強くする新しいカットであれば追加します。プール内のカットが追加されるカットより弱ければ、木のアクティブノードに適用されていない限り、取り除かれます。noduplが2に設定されると、同じテストが、カットタイプを無視して全てのカットに対して実施されます。
2. XPRsstorecutsは、mindex配列のカットプールに保存されたカットの添字番号のリストを返します。より強いカットが存在するために、カットプールにカットが追加されない場合は、-1の添字番号が返されます。mindex配列は、一番新しく行列に保存されたカッ

トをロードするために、XPRsloadcutsに直接渡されます。

3. カットの列と要素は、XPRsstorecutsに渡されるmcolsとdmtval配列内に連続的に保存されなくてはなりません。各カットの開始点はmstart配列に保存されます。最終カットの長さを決めるために、配列mstartは長さncuts+1であり、カットncuts+1この配列の最終要素でスタートします。

関連事項

XPRsbranchcut, XPRssetcbestimate, XPRssetcbsepnod, 章5.4「カットマネージャのはたらき」

dmatval	カットの行列値を含む長さmstart[ncuts]-1の倍精度型配列
MATRIXTOL	行列要素の零許容範囲
Xpress-Optimizer Reference Manual Console and Library Functions 240	
XPRswritebasis	

WRITEBASIS 6

XPRswritebasis WRITEBASIS

目的

後でOptimizerに入力するために、現在の基底をファイルに書きこみます。

概要

```
int XPRswritebasis(XPRsprob prob, char *filename,
char *flags);
WRITEBASIS [-flags] [filename]
```

引数

関係する制御

なし

例 1 (ライブラリ)

LPを解いた後で、将来、他のよく似た問題のより進んだ出発点とするために基底を保存して使うことは有効です。LPが複雑で解くのに時間がかかる場合に特に有効です。このためのコマンドは以下の通りです。

```
XPRsreadprob(prob, "myprob", "");
XPRsmaxim(prob, "");
XPRswritebasis(prob, "", "");
XPRsglobal(prob);
```

行列ファイルを読み、LP問題を最大化し、基底を保存して整数解を求めます。IP基底を保存することはあまり役に立たないので、上の例でもLP基底を保存しています。

prob	在の問題
filename	基底が読みこまれるファイル名を含む200文字以内の文字列。省略時はデフォルトで <code>problem_name</code> に拡張子 <code>.bss</code> が付いたものになります
flags	XPRswritebasis (WRITEBASIS)に渡されるフラグ
t	コンパクトな基底の出力。後で似た問題の入力として使うには XPRsreadbasis (READBASIS)にtフラグをつけて使うことが有効です

Xpress-Optimizer Reference Manual Console and Library Functions 241

XPRswritebasis

WRITEBASIS 6

例 2 (コンソール)

コンソールユーザの場合、上記と同様のコマンドは次の通りです。

```
READPROB
MAXIM
WRITEBASIS
GLOBAL
```

補足

1. bフラグを使ってXPRSreadprob (READPROB)コマンドで問題が入力された場合や長い行や列の名前を持つ問題のとき、基底を保存するにはtフラグオプションを使うしかありません。
2. 最適化にニュートンバリア法を使用した場合には、有効な基底を作る前にクロスオーバーを実行しておく必要があります。この基底はシンプレックス法(主, 双対)で再スタートする場合にのみ使えます。
3. 行列が事前分析済みでも、XPRSwritebasis (WRITEBASIS) は原問題の基底を書き出します。

関連事項

XPRSgetbasis, XPRSreadbasis (READBASIS).
Xpress-Optimizer Reference Manual Console and Library Functions 242
XPRSwriteomni

WRITEOMNI 6**XPRSwriteomni WRITEOMNI****目的**

現在の解をバイナリのOMNI形式ファイルSOLFILEに解ファイル`problem_name.sol`として書き込みます。また、現在の行列も書き込まれます。全ての情報がこのファイルに付け加えられます。

概要

```
int XPRSwriteomni(XPRSProb prob);
WRITEOMNI
```

引数**関係する制御**

Integer

String

例 1 (ライブラリ)

```
XPRSreadprob(prob, "bob", "");
XPRSSminim(prob, "");
XPRSsetintcontrol(prob, XPRS_OMNIFORMAT, 1);
XPRSwriteomni(prob);
XPRSreadprob(prob, "jo", "");
XPRSsetstrcontrol(prob, XPRS_OMNIDATANAME, "Jo ");
XPRSSminim(prob, "");
XPRSwriteomni(prob);
```

2つの解をSOLFILEに書き込みます。最初の解はbob.matにある問題名をデータ名フィールドに、2番目の解はデータ名前フィールドJoに入ります。

例 2 (コンソール)

コンソールでは次のコマンドが同意です。

```
READPROB bob
```

```
MINIM
OMNIFORMAT = 1
prob                現在の問題
OMNIFORMAT          OMNI出力に行列係数を含むかを表します
OMNIDATANAME        OMNIデータ名フィールドのためのデータ
Xpress-Optimizer Reference Manual Console and Library Functions 243
XPRswriteomni
```

WRITEOMNI 6

```
WRITEOMNI
READPROB jo
OMNIDATANAME = 'Jo '
MINIM
WRITEOMNI
```

補足

1. OMNI仕様に合わせるために、出力はSOLFILEというファイルになされます。このファイルが既にある場合は、XPRswriteomni(WRITEOMNI)の出力はSOLFILEに追加されます。ファイルがないときは、現在の作業ディレクトリにSOLFILEが作られます。このファイルは、Haverly Systems社のドキュメント「各種OptimizerへのOMNI標準インターフェース」に従ったフォーマットで書かれます。状態コードは、'OPTM', 'INFE', 'FEAS', 'UNBD'です。初期の仕様とはコードが変わっています。
2. MHDR, MROW, MCOLの各項目は、XPRswriteomni (WRITEOMNI)を呼ぶ前に制御パラメータOMNIFORMATを零でない値に設定すると作られます。例えば次のような場合です。

```
READPROB
MINIM
WRITEBASIS
OMNIFORMAT = 1
WRITEOMNI
QUIT
```

3. OMNI形式のデータ名フィールド（最初のバイトを第0バイトとして第8から15バイト）は、制御パラメータOMNIDATANAMEの内容が入ります。そのパラメータはXPRswriteomni (WRITEOMNI)を呼ぶ前に設定しておかなければなりません。

関連事項

なし

Xpress-Optimizer Reference Manual Console and Library Functions 244

XPRswriteprob

WRITEPROB 6

XPRswriteprob WRITEPROB

目的

現在の問題をMPSあるいはLPファイルに書きこみます。

概要

```
int XPRswriteprob(XPRsprob prob, char *filename,
char *flags);
```

```
WRITEPROB [-flags] [filename]
```

引数

関係する制御

なし

例 1 (ライブラリ)

現在の問題をフル精度で出力し、ファイル *problem_name.lp* にベクトル名が含まれる LP 形式で出力します。

```
XPRswriteprob(prob, "", "lps");
```

例 2 (コンソール)

```
WRITEPROB -p C:myprob
```

Optimizer に MPS 行列を *C* ドライブのファイル *myprob.mat* に倍精度で書くように指示します。

prob	現在の問題
filename	filename 問題が書かれるファイル名を含む 200 文字以内の文字列。拡張子 <i>.lp</i> が使われるフラグが 1 のときでない限り、省略時はデフォルト <i>problem_name</i> に拡張子 <i>.mat</i> がついたものになります
flags	フラグ。次のいずれかです
	p フル精度数値
	o 1 行に 1 要素
	n スケーリング済み
	s ベクトル名を含む
	l LP 形式の出力

Xpress-Optimizer Reference Manual Console and Library Functions 245

XPRswriteprob**WRITEPROB 6****警告**

XPRsreadprob (READPROB) が問題の入力に使われる場合、新しいファイル名を指定しないと、XPRswriteprob (WRITEPROB) によって上書きされます。

補足

XPRsloadlp, XPRsloadglobal, XPRsloadqglobal, XPRsloadqp が行列を得るために使用される場合、目的関数と行列の *N* 行との間に関連はなく、(*__OBJ__* と呼ばれる) 分離 *N* 行は、XPRswriteprob (WRITEPROB) を実行するときに生成されます。また XPRsreadprob (READPROB) を実行して、目的行あるいは目的行に相当する行列の *N* 行のどちらかが変更されると、二つの関連はなくなり、*__OBJ__* 行は XPRswriteprob (WRITEPROB) を実行するときに生成されます。XPRsreadprob (READPROB) を実行するとき、行列から目的関数行を取り除くためには、XPRsreadprob (READPROB) の前に KEEP_NROWS に -1 を設定します。

関連事項

XPRsreadprob (READPROB).

Xpress-Optimizer Reference Manual Console and Library Functions 246

XPRswriteprtrange**WRITEPRTRANGE 6****XPRswriteprtrange WRITEPRTRANGE****目的**

ルーチン XPRsrange (RANGE) により保存されたレンジ情報を ASCII 形式のファイル *problem_name.rtrt* に書きこみます。

概要

```
int XPRswriteprtrange(XPRsprob prob);
```

```
WRITEPRTRANGE
```

```
PRINTRANGE
```

引数**関係する制御***Integer**Double***例 1 (ライブラリ)**

LP問題を解いてプリントファイルに結果を出力する前にXPRStrange(RANGE)を呼びだします。

```
XPRsreadprob(prob, "myprob", "");
```

```
XPRsmaxim(prob, "");
```

```
XPRsrange(prob);
```

```
XPRswriteprtrange(prob);
```

例 2 (コンソール)

コンソールユーザの場合、次のコマンドで同様に実行できます。

```
READPROB
```

```
MAXIM
```

```
RANGE
```

```
WRITEPRTRANGE
```

```
prob
```

現在の問題

```
MAXPAGELINES
```

改ページ行数

```
OUTPUTLOG
```

メッセージの印刷を実施するかについて

```
OUTPUTTOL
```

印刷値の零許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 247

XPRswriteprtrange**WRITEPRTRANGE 6****補足**

1. (コンソール) 画面に同じ情報を出力する同様のコマンドPRINTRANGEがあります。形式はXPRswriteprtrange (WRITEPRTRANGE)によるファイル出力と同じですが、更に出力が必要なときに各画面の後に応答が入力可能をなる場合を除きます。
2. このコマンドで生成されるASCII形式のファイルは、XPRsriterange (WRITERANGE)で生成されるファイルほど容易に使えるものではありません。フィールドが固定幅で出力されるためです。
XPRswriteprtrange (WRITEPRTRANGE)は、印刷できるファイルを作るのが主たる目的です。このレンジファイルの形式については、付録A「ログとファイル形式」に説明があります。
3. 制御OUTPUTLOG は出力を作成するためには零以外でないといけません。

関連事項

XPRsgetcolrange, XPRsgetrowrange, XPRsrange (RANGE),

XPRswriteprtsol, XPRsriterange, 付録A.4「ASCIIレンジファイル」

Xpress-Optimizer Reference Manual Console and Library Functions 248

XPRswriteprtsol**WRITEPRTSOL 6****XPRswriteprtsol WRITEPRTSOL****目的**

SOLUTIONFILEを1に設定したとき、現在の解をASCIIファイル`problem_name.prt`に出力します。

概要

```
int XPRswriteprtsol(XPRsprob prob);
WRITEPRTSOL
PRINTSOL
```

引数**関係する制御***Integer**Double***例 1 (ライブラリ)**

次の例は、この関数の標準的な使い方、最適化に続いて解を書き出します。

```
XPRsreadprob(prob, "myprob", "");
XPRsmaxim(prob, "");
XPRswriteprtsol(prob);
```

例 2 (コンソール)

```
READPROB
MAXIM
PRINTSOL
```

コンソールユーザが画面に出力を直接表示したい場合の例です。

prob	現在の問題
MAXPAGELINES	改ページ行数
OUTPUTLOG	メッセージの印刷の実施について
REL10STYLE	双対変数の旧型表記の使用について
OUTPUTTOL	出力値の零許容範囲

Xpress-Optimizer Reference Manual Console and Library Functions 249

XPRswriteprtsol**WRITEPRTSOL 6****補足**

1. (コンソール) 画面に同じ情報を表示する同様のコマンド PRINTSOL があります。形式は XPRswriteprtsol (WRITEPRTSOL) で得られるファイルと同じですが、更に出力が必要な場合に各画面の後に応答を入力できる場合を除きます。
2. このコマンドで生成される ASCII 形式のファイルは、XPRswritesol (WRITESOL) で生成されるファイルほど容易に使えるものではありません。フィールドが固定幅で出力されるためです。
XPRswriteprtsol (WRITEPRTSOL) は、直接プリンタへ送ることのできるファイルを作るのが目的です。この解印刷ファイルの形式については、付録 A 「ログとファイル形式」に説明があります
3. 制御 OUTPUTLOG は出力作成のためには零以外でないといけません

関連事項

XPRsgetsol, XPRswriteprtrange, XPRswritesol, 付録 A.3 「ASCII 解ファイル」

Xpress-Optimizer Reference Manual Console and Library Functions 250

XPRswriterange**WRITERANGE 6****XPRswriterange WRITERANGE****目的**

ルーチン XPRsrange (RANGE) により保存されたレンジ情報を ASCII ファイルとそれに関係す

るヘッダファイルに書きこみます。

概要

```
int XPRewriterange(XPRsprob prob, char *filename,
char *flags);
WRITERANGE [filename] [-flags]
```

引数

関係する制御

Double

String

例 1 (ライブラリ)

基本的には、XPRewriterange (WRITERANGE)の使い方はXPRwriteprtrange (WRITEPRTRANGE)に似ていますが、出力は他のプログラムの入力に向いています。次の例にその使い方を示します。

prob	現在の問題
filename	解が書かれるファイル名を含む200文字以内の文字列。ファイル名に拡張子 .hdr と .rscが付けられますが、省略時はデフォルトの <i>problem_name</i> が使用されます
flags	XPRewriterange (WRITERANGE)に渡されるフラグ
	s 連続番号
	n 名前
	t タイプ
	b 基底状態
	a アクティビティ
	c コスト (列), スラック (行)
	フラグの指定がない場合、デフォルトとして、全てのフィールドを出力します
OUTPUTLOG	メッセージの出力について
OUTPUTTOL	出力値の零許容範囲
OUTPUTMASK	ファイルへ出力する行, 列名を制限するためのマスク

Xpress-Optimizer Reference Manual Console and Library Functions 251

XPRewriterange

WRITERANGE 6

```
XPRsreadprob(prob, "myprob", "");
XPRsminim(prob, "");
XPRsrange(prob);
XPRewriterange(prob, "", "");
```

例 2 (コンソール)

RANGE

WRITERANGE -nbac

この例ではベクトルごとに、名前、基底状態、アクティビティ、(列)コスト、(行)スラックの値をファイル *problem_name.rsc*に出力します。この他にユーザの選択により他のレンジ情報の他のフィールド番号も出力されます / ません。

補足

- 以下のフィールドは .rsc ファイルの中に以下の順序で必ず現れます。レンジファイルの詳細は付録A「ログとファイル形式」に説明があります。

下限アクティビティ
 単価コスト下げ
 コスト上限値 (最大化の場合は、利益下限値)
 極限プロセス下げ
 極限プロセス下げの状態
 上限アクティビティ
 単価コスト上げ
 コスト下限値 (最大化の場合は、利益上限値)
 極限プロセス上げ
 極限プロセス上げの状態

行については、コスト下限値とコスト上限値は零です。制限過程, アクティビティがなければ、そのフィールドは空白のまま、倍精度クオーテーションで範囲が定められます。

2. 制御パラメータOUTPUTMASKはASCIIファイルに報告されるベクトルを制御するために使われます。OUTPUTMASKに一致する名前のベクトルだけが出力されます。デフォルトは '????????' で、このときは全てのベクトルが出力されます。

関連事項

XPRsgetsol, XPRswriteprtrange (WRITEPRTRANGE), XPRsrange (RANGE),
 XPRswritesol (WRITESOL), A.4 「ASCII レンジファイル」
 Xpress-Optimizer Reference Manual Console and Library Functions 252
XPRswritesol

WRITESOL 6

XPRswritesol WRITESOL

目的

現在の解をASCIIファイルに書きこみます。

概要

```
int XPRswritesol(XPRsprob prob, char *filename,
char *flags);
```

```
WRITESOL [filename] [-flags]
```

引数

関係する制御

Integer

Double

prob

現在の問題

filename

解が書かれるファイルの名前を含む200文字以内の文字列。ファイル名に拡張子 .hdr と .asc が付けられますが、省略時は *problem_name* が使われます

flags

XPRswritesol (WRITESOL) に渡されるフラグ

s 連続番号

n 名前

t タイプ

b 基底状態

a アクティビティ

c コスト(列), スラック (行)

l 下限値

u 上限値
 d dj (列; 被約費用), 双対値 (行; シャドウプライス)
 r 右辺 (行)
 e 各MIP, 目標計画の解の出力
 q 零でない最適値をもつベクトルだけを出力

REL10STYLE 旧型の双対値の使用について
 OUTPUTTOL 出力値の零許容範囲
 Xpress-Optimizer Reference Manual Console and Library Functions 253
XPRSwritesol

WRITESOL 6

String

例 1 (ライブラリ)

最適化に続いて基底状態を(連続番号で)出力します。

```
XPRsreadprob(prob, "richard", "");
XPRsminim(prob, "");
XPRSwritesol(prob, "", "sb");
```

例 2 (コンソール)

次のようなファイルを作成します。

- 名前がxから始まり、値が零でない変数の名前と値、及び
- CO2で始まる制約条件の名前と値と右辺

Optimizerに必要なコマンドは次の通りです。

```
OUTPUTMASK = 'X????????'
WRITESOL XVALS -naq
OUTPUTMASK = 'CO2???????'
WRITESOL CO2 -nar
```

補足

1. このコマンドはfilename.hdr (ヘッダファイル) とfilename.asc (ASCII solution ファイル) の2つの読み込み可能なファイルを作ります。ヘッダファイルには1行に要約した情報が入っています。ASCIIファイルには、問題の各行、各列の情報が1行ずつ入っています。.asc ファイルには、フィールドが上記のフラグ順に現れます。フラグのリストに示された順番は関係ありません。
2. また、マスク制御OUTPUTMASK はASCIIファイルに書き込まれる名前を制御するために使われます。名前がOUTPUTMASKに一致するベクトルだけが出力されます。OUTPUTMASKはデフォルトでは「????????」が設定され、全てのベクトルが出力されます。
3. KEEPMIPSOL が、多くのMIP解、目標計画解を保存するために使われる場合、eフラグは保存してある解ごとに解情報を出力するために使われます。最もよい解は、*problem_name.hdr*, *problem_name.asc*に出力されます。他の解はヘッダファイル *problem_name.hd0*, *problem_name.hd1*, ... そして、ASCII形式の解ファイル *problem_name.as0*, *problem_name.as1*, ...に出力されます。

OUTPUTMASK ファイルへ出力する行, 列名を制限するためのマスク
 Xpress-Optimizer Reference Manual Console and Library Functions 254
XPRSwritesol

WRITESOL 6

関連事項

XPRSwriterange (WRITERANGE), XPRSwriteprtsol (WRITEPRTSOL), .

Control Parameters 7

7 制御パラメータ

Optimizerには解法や出力様式を制御するために様々な制御があります。これらの多くは整数値をとり、種々の動作のスイッチとして機能します。値の許容範囲は倍型精度で、一部については構造に名前を設定するために文字列となっています。これらについては、ユーザがOptimizerの能力を高めるために変更できます。しかし、設定されているデフォルト値は、これまでに様々な問題に適用して良好な動作を得られた実績をもとに設定されていますので、変更の場合にはよく検討してください。

7.1 制御値の扱い方

コンソールXpressユーザは、Optimizerのプロンプト、>,で制御名とリターンキーを入力して制御値を得ることができます。制御は指定された文法を使って設定できます。

```
control_name = new_value
```

ここでnew_valueは適切な整数値、倍精度型値または文字列です。文字列については' 'で囲まれ全8文字が与えられます。

Xpress-MPのライブラリユーザは、制御値を設定したり得るために次のような関数の組が与えられます。本章にリストしたような制御はXpress-MP ライブラリに接頭辞XPRS_を前に付ける必要があり、そうしないとエラーになってしまうことは極めて重要です。次に例を示します。

```
XPRSgetintcontrol(prob, XPRS_PRESOLVE, &presolve);
printf("The value of PRESOLVE is %d\n", presolve);
XPRSsetintcontrol(prob, XPRS_PRESOLVE, 1-presolve);
printf("The value of PRESOLVE is now %d\n", 1-presolve);
XPRSgetintcontrol XPRSgetdblcontrol XPRSgetstrcontrol
XPRSsetintcontrol XPRSsetdblcontrol XPRSsetstrcontrol
Xpress-Optimizer Reference Manual Control Parameters 256
```

AUTOPERTURB

- BACKTRACK 7

AUTOPERTURB

BACKTRACK

解説 シンプレックス法:自動的に摂動を実施するかを決めます。1に設定されると、シンプレックス法が極端な数の退化ピボットステップに遭遇した場合、PERTURBにより問題が摂動され、Optimizerは退化により機能低下するのを防ぎます。

タイプ 整数

値	0	摂動しません
	1	自動的に摂動します

デフォルト値 1

作用ルーチン XPRSmaxim (MAXIM)/XPRSminim (MINIM).

解説 分枝限定法: 木探索において次のノードの選び方を決めます。

タイプ 整数

値	1	MIPTARGETが設定されなければ、最もよい推定量のノードを選びます。MIPTARGETが(ユーザが整数解を見つける前に整数探索により)設定されれば、Forrest-Hirst-Tomlin基準により選ばれます。これは現在、最もよい整数解を考慮して、改善される可能性が高い新しいノードを探します
	2	最もよい推定解のノードを常に選びます
	3	その解において最もよい上下界値のノードを常に選びます

デフォルト値 3

作用ルーチン XPRSGlobal (GLOBAL).

Xpress-Optimizer Reference Manual Control Parameters 257

BAR DUALSTOP

- BARGAPSTOP 7

BAR DUALSTOP

BARGAPSTOP

解説 ニュートンバリア法: 双対実行不能の許容範囲を表す収束パラメータ。双対問題の制約と上下界値の差が絶対値許容範囲を下回ると、最適化は停止し、そのときの解が返されます。

タイプ 倍精度

デフォルト値 1.0E-08

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 ニュートンバリア法: 相対双対ギャップの許容範囲を表す収束パラメータ。主目的関数と双対目的関数値の差がこの許容範囲を下回ると、Optimizerは最適解が見つかったと判断します。

タイプ 倍精度

デフォルト値 1.0E-08

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 258

BAR ITERLIMIT

- BARMEMORY 7

BAR ITERLIMIT

BARMEMORY

解説 ニュートンバリア法: 繰返し回数の最大値。シンプレックス法は、通常、問題の制約(行)の数に応じた繰返し計算を実施しますが、バリア法は問題の大きさとは独立な繰返し計算の後、与えられた精度の最適解を見つけます。ペナルティは問題サイズに伴う各繰返し時間の増加です。

BARITERLIMITはバリアにより実行される繰返しの最大回数を指定します。

タイプ 整数

デフォルト値 200

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 ニュートンバリア法: 最適解の探索におけるバリアアルゴリズムによって使われるメモリをメガバイト単位で指定します。0に設定すると、Optimizerにより自動的に決められます。

タイプ 整数

デフォルト値 0 —メモリは自動的に決められます

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 259

BAR ORDER

- BAR OUTPUT 7

BAR ORDER

BAR OUTPUT

解説 ニュートンバリア法: コレスキー因子分解の順序アルゴリズムを指定し、因子分解された行列の疎密を保存するのに使われます。

タイプ 整数

値 0 自動的に選択します

1 最小度法。行, 列における最小非零数の対角要素を選びます

2 最小ローカルフィル法。行列内の非零要素の近接グラフを考え、新しい辺の作

- 成を最小化するノードを減らすように探します
 3 ネスト解体法。近接グラフを考え、反復的にそれを非近接部分に分割するように探します

デフォルト値 0

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 ニュートンバリア法: 解の出力レベルを指定します。出力はアルゴリズムの各繰返し後に得られるか、このパラメータにより完全にオフにされます。

タイプ 整数

値 0 出力なし
 1 各繰返しにおいて出力

デフォルト値 1

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 260

BARPRIMALSTOP

– **BARSTEPSTOP** 7

BARPRIMALSTOP

BARSTEPSTOP

解説 ニュートンバリア法: 主実行不能の許容範囲を示す収束パラメータ。主問題の制約と上下界値との間の差がこの絶対値許容範囲を下回ればOptimizerは終了し、そのときの答を返します。

タイプ 倍精度

デフォルト値 1.0E-08

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 ニュートンバリア法: 最小ステップサイズを表す収束パラメータ。バリア法の各繰返しでは、計算された探索方向に沿ってステップは実施されます。ステップサイズがBARSTEPSTOPより小さい場合、Optimizerは終了し、その時の解を返します。

タイプ 倍精度

デフォルト値 1.0E-10

注意 バリア法が後の繰返しにおいてBARGAPSTOPで改善を少しする場合、この値を大きくした方が、発見済みの最適解に近い解が得られるため有効です。

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 261

BARTHREADS

– **BIGM** 7

BARTHREADS

BIGM

解説 ニュートンバリア法: アルゴリズムを実行するためになされたスレッドの数。通常、Parallel Xpress-MP を 1 台のマルチプロセッサ機上で動作させるときにプロセッサの数を設定します。

タイプ 整数

デフォルト値 1

注意 BARTHREADS の値はユーザのライセンスによって決まります。ライセンスで認められている以上の数に設定すると、Optimizerは最適化に先立って修正します。最適化後にその値を得ることで、実際にプロセッサがいくつ使われたかが分かります。

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 Big M法が実施される場合、実行不能ペナルティが使われます。

タイプ 倍精度

デフォルト値 行列によります

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 262

BIGMMETHOD

- BREADTHFIRST **7**

BIGMMETHOD

BREADTHFIRST

解説 シンプレックス法: Big M法を使うか、標準フェーズI(実行可能到達)とフェーズII(最適化到達)を使うかを指定します。Big M法では、目的関数の変数の係数は実行可能フェーズにおいて考慮され、できるだけ最適解に近い最初の実行可能基底に到達します。問題内のBig M要素の存在によりround-offエラーの可能性がある副作用があります。

タイプ 整数

値	0	フェーズI/フェーズII
	1	Big M法が使われます

デフォルト値 1

注意 XPRsreadprob (READPROB), XPRsloadglobal, XPRsloadlp, XPRsloadqglobal, XPRsloadqpによりリセットされます

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 NODESELECTION = 4のとき、深さ優先探索において含まれるノードの数を決定します。

タイプ 整数

デフォルト値 10

作用ルーチン XPRsglobal (GLOBAL).

Xpress-Optimizer Reference Manual Control Parameters 263

CACHESIZE

- CHOLESKYTOL **7**

CACHESIZE

CHOLESKYALG

CHOLESKYTOL

解説 ニュートンバリア法: ユーザのコンピュータのキャッシュサイズ (KB)。インテルプラットフォームでは-1にすると自動的にキャッシュサイズを決めます。

タイプ 整数

デフォルト値 ハードウェア/プラットフォームに依存します

注意 サイズが分からない場合、小さめに設定することをお勧めします。Windowsにおいて自動的にサイズを決められない場合には、デフォルトの512kBが使われます。

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 コレスキー因子分解のタイプ。

タイプ 整数

値	0	Cholesky不採用
	1	Cholesky採用

デフォルト値 1

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 ニュートンバリア法: 普通の等式係数行列のコレスキー分解におけるピボット要素の零許容範囲で、バリア法の各繰返しにおいて計算されます。ピボット要素の絶対値がCHOLESKYTOL以下であれば、コレスキー分解過程において特別に取り扱われる利点があります。

タイプ 倍精度

デフォルト値 1.0E-15

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 264

COVERCUTS

- CPKEEPALLCUTS 7

COVERCUTS

CPKEEPALLCUTS

解説 分枝限定法: トップノードにおいて採用された被覆不等式の使用回数。採用された被覆不等式は可能性のある整数解を除外せずに、実行可能領域の大きさを減らす際に特に有効な追加制約です。これらを生成する過程は、例え時間ペナルティを負っても何度も実行され、実行可能領域が更に縮小されます。トップノードでこれらを生成することで、通常、良い解が得られます。と言うのは、これらの不等式は木探索においてあらゆるノードに対して継続的に適用されるためです。

タイプ 整数

デフォルト値 20

作用ルーチン XPRsglobal (GLOBAL).

解説 カットプール: ユーザが作成したアクティブでないカットをカットプールから取り除くかを示します。これにより、カットを再度続いて作成しなくてはならないときに、計算時間が増えてもメモリを節約できます。

タイプ 整数

値	0	アクティブでないカットを取り除かない
	1	アクティブでないカットを取り除く

デフォルト値 1

注意 自動生成のカットはカットプールから取り除かれません。

作用ルーチン XPRssetcbcutmgr.

Xpress-Optimizer Reference Manual Control Parameters 265

CPMAXCUTS

- CPUTIME 7

CPMAXCUTS

CPMAXELEMS

CPUTIME

解説 カットプール: カットプールに保存されるカットの初期最大数。最適化中にカットプールは自動的にリサイズされます。

タイプ 整数

デフォルト値 100

作用ルーチン XPRssetcbcutmgr.

解説 カットプール: カットプールに保持される非零係数の初期最大数。最適化中、カットプールは自動的にリサイズされます。

タイプ 整数

デフォルト値 200

作用ルーチン XPRssetcbcutmgr.

解説 求解時間を報告するのに使われる時間基準

タイプ 整数

値	0	経過時間
	1	CPU時間

デフォルト値 1

注意 Windows では経過時間が常に使われます。

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM), XPRsglobal (GLOBAL), XPRsrecurse (RECURSE).

Xpress-Optimizer Reference Manual Control Parameters 266

CRASH

- CROSSOVER 7

CRASH

CROSSOVER

解説 シンプレックス法: アルゴリズムの開始時に使われたクラッシュのタイプを決めます。クラッシュ処理中、初期基底は実行可能性と三角可能性に可能な限り近づくように決定されます。ここでのよい選択は最適解を見つけるのに必要な反復回数を減らすことです。その値は時間の消費に応じて増加します。

タイプ 整数

値	0	全クラッシュ処理を行いません
	1	シングルトンのみ (ワンパス)
	2	シングルトンのみ (マルチパス)
	3	スラックを考慮した、行列のマルチパス
	4	行列全体のマルチ(10)パスですが、最後にはスラックのみを対象とします
	$n > 10$	4のとき、最大で $n - 10$ パスを実施します

デフォルト値 2

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 ニュートンバリア法: 最適解が見つかった場合、バリア法からシンプレックス法にクロスオーバーを実施するかを制御し、最終基底と感度分析情報を示す。

タイプ 整数

値	0	クロスオーバーなし
	1	基底解にクロスオーバーする

デフォルト値 1

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 267

CSTYLE

- CUTFREQ 7

CSTYLE

CUTDEPTH

CUTFREQ

解説 配列の番号付けの際に用いられる表記法

タイプ 整数

値	0	配列にFORTRANのルールが用いられることを示します(1から配列番号開始)
	1	配列にCのルールが用いられることを示します(0から配列番号開始)

デフォルト値 1

作用ルーチン 引数として配列を採用する全ライブラリルーチン

解説 分枝限定法: カットが作成される箇所における木探索で最大深さを設定する。カットの作成には長時間を要し、木の深いレベルでは余り重要でないことがしばしばである。と言うのは、変数についての強い上下界値は既に行実行可能領域を縮小化済みであるためです。値0はカットが作られないことを示します。

タイプ 整数

デフォルト値 0 — デフォルトで木にカットが生成されません。

作用ルーチン XPRSglobal (GLOBAL).

解説 分枝限定法: 木探索でカットが作成される頻度を決めます。ノード深さCUTFREQが零であれば、カットが作成されます。

タイプ 整数

デフォルト値 8

作用ルーチン XPRSglobal (GLOBAL).

Xpress-Optimizer Reference Manual Control Parameters 268

CUTSTRATEGY

- DEFAULTALG 7

CUTSTRATEGY

DEFAULTALG

解説 分枝限定法: カット戦略を規定します。カットを多く作成する攻撃的なカット戦略は、探索ノードを減らしますが、カット作成に関係する時間費用がかかります。カットの生成を減らすと、時間が減りますが、たくさんのノードを探すことになります。

タイプ 整数

値	-1	カット戦略を保守的か攻撃的か自動的に選択します
	0	カットなし
	1	保守的なカット戦略
	2	攻撃的なカット戦略

デフォルト値 -1

作用ルーチン XPRSglobal (GLOBAL).

解説 最適化ルーチンに渡されるアルゴリズムフラグがない場合、LPを解くのに使うアルゴリズムを選びます。

タイプ 整数

値	1	自動的に決まります
	2	双対シンプレックス
	3	主シンプレックス
	4	ニュートンバリア

デフォルト値 1

作用ルーチン XPRSm Maxim (MAXIM)/ XPRSm minim (MINIM), XPRSglobal (GLOBAL).

Xpress-Optimizer Reference Manual Control Parameters 269

DEGRADEFACTOR

- ELIMTOL 7

DEGRADEFACTOR

DENSECOLLIMIT

ELIMTOL

解説 分枝限定法: 木において未探索ノードに関する推定劣化量を増やす要素です。推定劣化量は与えられたノードを探索することで得られる整数解において目的関数の見込まれる悪化量のことです。

タイプ 倍精度

デフォルト値 1.0

作用ルーチン XPRSglobal (GLOBAL).

解説 ニュートンバリア法: DENSECOLLIMIT要素以上の列は、密であるとして考慮されます。このような列は行列のコレスキー因子分解において特別に扱われます。

タイプ 整数

デフォルト値 0 — 自動的に決められます

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM)

解説 事前分析の減少フェーズのためのマルコビット許容範囲

タイプ 倍精度

デフォルト値 1.0E-03

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 270

ETATOL

- EXTRACOLS 7

ETATOL

EXTRACOLS

解説 イータ要素の零許容範囲それぞれの反復に対して、基底逆行列が基本行列に乘じられる。基本行列は、1つの列のイータベクトルを除いて同一とする。絶対値がETATOLより小さなイータベクトルの要素は、ここでは零となります。

タイプ 倍精度

デフォルト値 1.0E-12

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM), XPRsbtran,

XPRsftran.

解説 行列内に追加できる列の初期数。列が行列に追加されると、最大効果をあげるためにEXTRACOLS制御の設定により、行列が入力される前に、列のためのスペースが準備されます。これがなされない場合には、自動的にリサイズされますが、実際にユーザが必要とするスペース以上に割り当てられます。

タイプ 整数

デフォルト値 0

作用ルーチン XPRsreadprob (READPROB), XPRsloadglobal, XPRsloadlp,

XPRsloadqglobal, XPRsloadqp.

EXTRAROWS, EXTRAELEMENTS, EXTRAMIPENTSもご覧ください。

Xpress-Optimizer Reference Manual Control Parameters 271

EXTRAELEMENTS

- EXTRAMIPENTS 7

EXTRAELEMENTS

EXTRAMIPENTS

解説 カットの係数を含む行列に追加可能な要素の初期数。行や列が行列に追加されると、最大効果をあげるために、EXTRAELEMENTS制御の設定により、行列が入力される前に行列要素のためのスペースが準備されます。これがなされない場合には、自動的にリサイズされますが、実際にユーザが必要とするスペース以上に割り当てられます。カット係数のためのスペースは、行や列が追加された後で、整数探索の開始前の追加行列の要素数に同じです。EXTRAELEMENTSは、行列が最初にカットのためのスペースに入力されたとき、Optimizerによって自動的に設定されますが、行や列を追加しても、この自動的な設定はアップデートされません。自動カットかユーザカットを行列に追加したい/追加する場合、EXTRAELEMENTSは行列が最初に入力される前に設定されなくてはなりません。そして、追加したいカットと行や列の追加可能なスペースができます。

タイプ 整数

デフォルト値 ハードウェアやプラットフォームによります

作用ルーチン XPRSreadprob (READPROB), XPRSloadglobal, XPRSloadlp, XPRSloadqglobal, XPRSloadqp, XPRSsetcbcutmgr.

EXTRACOLS , EXTARROWSもご覧ください。

解説 追加できる離散要素数の初期値

タイプ 整数

デフォルト値 0

作用ルーチン XPRSreadprob (READPROB), XPRSloadglobal, XPRSloadqglobal.

Xpress-Optimizer Reference Manual Control Parameters 272

EXTRAPRESOLVE

- EXTRAROWS 7

EXTRAPRESOLVE

EXTRAROWS

解説 事前分析における追加できる要素数の初期値

タイプ 整数

デフォルト値 ハードウェアやプラットフォームによります

注意 追加事前分析要素を保存するために必要なスペースはダイナミックに割り当てられます。そのため、この制御を設定する必要はありません。

作用ルーチン XPRSreadprob (READPROB), XPRSloadglobal, XPRSloadlp, XPRSloadqglobal, XPRSloadqp.

解説 行列に追加可能な行の数の初期値でカットを含みます。行列に行が追加されると、最大効果をあげるために、EXTRAROWS制御の設定により、行列の入力前行のためのスペースが準備されます。これがなされない場合には、自動的にリサイズされますが、実際にユーザが必要とするスペース以上に割り当てられます。カットのためのスペースは、行が追加された後で、整数探索の開始前の追加行の数と同じです。EXTRAROWSは、最初に行列がカットのためのスペースに入力されたとき、Optimizerによって自動的に設定されますが、行を追加しても、この自動的な設定はアップデートされません。自動カットかユーザカットを行列に追加したい/する場合、EXTRAROWSは行列が最初に入力される前に設定されなくてはなりません。そして、行列に追加したいカットや行の追加可能なスペースができます。

タイプ 整数

デフォルト値 行列の性質に依存します。

作用ルーチン XPRSreadprob (READPROB), XPRSloadglobal, XPRSloadlp, XPRSloadqglobal, XPRSloadqp, XPRSsetcbcutmgr.

EXTRACOLSもご覧ください。

Xpress-Optimizer Reference Manual Control Parameters 273

FEASTOL

- GOMCUTS 7

FEASTOL

GOMCUTS

解説 右辺値, 上下界値, レンジ値, つまり基底変数の上下界値の零許容範囲。この中の1つが絶対値FEASTOL以下であれば、零として扱います。

タイプ 倍精度

デフォルト値 1.0E-06

作用ルーチン XPRSmaxim (MAXIM)/XPRSminim (MINIM), XPRSgetinfeas.

解説 分枝限定法: トップノードでのGomoryカットの使用回数です。現在のノードが整数解を作らないのであれば、常に生成されます。しかし、Gomoryカットは常に実行可能領域の大きさを減らすために採用された被覆不等式と同じように有効とは限りません。

タイプ 整数

デフォルト値 2

作用ルーチン XPRSGlobal (GLOBAL).

Xpress-Optimizer Reference Manual Control Parameters 274

INVERTFREQ

– INVERTMIN **7**

INVERTFREQ

INVERTMIN

解説 シンプレックス法: 基底の逆行列計算の頻度。基底は因子分解された形で維持され、ほとんどのシンプレックス反復において、採用されたステップを反映して次々とアップデートされます。何度もの反復の後、基底が余り良い状態でなくなり、十分な逆行列計算の必要があっても、これは各反復において逆行列計算をするより相当速くなります。

タイプ 整数

デフォルト値 -1 — 自動的に頻度が決められます

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 シンプレックス法: 基底行列の逆行列を計算する際数の最小値。詳細はINVERTFREQの解説をご覧ください。

タイプ 整数

デフォルト値 3

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 275

KEEPBASIS

– KEEPMIPSOL **7**

KEEPBASIS

KEEPMIPSOL

解説 シンプレックス法: 次の反復に使う基底を決めます。最初の反復でのクラッシュ処理により決定されたものか、最後の反復の基底を使うかを選択します。

タイプ 整数

値 0 最初の反復から最適化を開始します。つまり、以前の基底は無視されます。

1 先にロードされた基底（メモリにあるもの）が使われます。

デフォルト値 1

注意 最適化が実施された後ではデフォルト値に戻ります。

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 分枝限定法: 保持する整数解の数。木探索中、一般に、最適解であるかないかは別にして、いくつかの整数解が見つかります。メモリに保持する整数解の数はKEEPMIPSOLで表されます。

タイプ 整数

デフォルト値 1 — デフォルトでは最適解のみ保持されます

作用ルーチン XPRsgoal (GOAL).

参照 XPRswritesol (WRITESOL) にeフラグを付ける場合

Xpress-Optimizer Reference Manual Control Parameters 276

KEEPNROWS

- LPLOG 7**KEEPNROWS****LPITERLIMIT****LPLOG**

解説 非束縛行の状態

タイプ 整数

値 -1 Nタイプの行を消去し、予備行のスペースを作成します
 0 Nタイプの行を消去します
 1 Nタイプの行を保持します

デフォルト値 1

作用ルーチン XPRSreadprob (READPROB), XPRSloadglobal, XPRSloadlp,
 XPRSloadqglobal, XPRSloadqp.

解説 シンプレックス法: 最適化過程が終了する前に、実施される反復回数の最大値。MIP問題の場合、これは分枝限定法で探される各ノードでの反復回数の最大値です。

タイプ 整数

デフォルト値 2147483645

作用ルーチン XPRSmaxim (MAXIM)/XPRSminim (MINIM).

解説 シンプレックス法: 反復ログが出力される頻度。

タイプ 整数

値 $n < 0$ n 回の反復ごとに詳細に出力
 0 解探索が終了した際のみログを出力
 $n > 0$ n 回の反復ごとに概要を出力

デフォルト値 100

作用ルーチン XPRSmaxim (MAXIM)/XPRSminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 277

MARKOWITZTOL**- MAXCUTTIME 7****MARKOWITZTOL****MATRIXTOL****MAXCUTTIME**

解説 基底行列の因子分解で使われるマルコビッツ許容範囲。

タイプ 倍精度

デフォルト値 0.01

作用ルーチン XPRSmaxim (MAXIM)/XPRSminim (MINIM).

解説 行列要素の零許容範囲。行列要素の値が絶対値MATRIXTOL以下であれば零として扱います。

タイプ 倍精度

デフォルト値 1.0E-09

作用ルーチン XPRSreadprob (READPROB), XPRSloadglobal, XPRSloadlp,
 XPRSloadqglobal, XPRSloadqp, XPRSalter (ALTER),
 XPRSaddcols, XPRSaddcuts, XPRSaddrows, XPRSchgcoef,
 XPRSchgmcoef, XPRSstorecuts.

解説 カット面の作成と再最適化のための許容時間の最大値。この限度は作成中にチェックされ、一旦限度を超えるとカットをそれ以上追加しません。

タイプ 整数

値 0 時間制限なし
 $n > 0$ n 秒後からはカットを生成しない

デフォルト値 0

作用ルーチン XPRsmaxim (MAXIM) / XPRsminim (MINIM), XPRsglobal (GLOBAL).

Xpress-Optimizer Reference Manual Control Parameters 278

MAXIIS

– MAXNODE **7**

MAXIIS

MAXMIPSOL

MAXNODE

解説 探索する既約実行不可能集合の数を制御します。

タイプ 整数

値 -1 IISの各々について探します
 0 何も探しません
 $n > 0$ 最初の n 個のIISについて探します

デフォルト値 -1

作用ルーチン XPRSiis (IIS), XPRsgetiis.

解説 分枝限定法: 一時停止して続行するかをチェックする前に、Optimizerで見つけられた整数解の数の限度を指定します。最適化の間、Optimizerは別のノードから同じ目的関数値の解を見つけてます。しかし、MAXMIPSOLは見つかる整数解の総数を決め、必ずしも値が違う解の数ではありません。

タイプ 整数

デフォルト値 0

作用ルーチン XPRsglobal (GLOBAL).

解説 分枝限定法: Optimizerが一時停止して続行するかをチェックする前に探索されるノードの最大値。

タイプ 整数

デフォルト値 100000000

作用ルーチン XPRsglobal (GLOBAL).

Xpress-Optimizer Reference Manual Control Parameters 279

MAXPAGELINES

– MAXTIME **7**

MAXPAGELINES

MAXSLAVE

MAXTIME

解説 印刷可能な出力における改ページ間の行数。

タイプ 整数

デフォルト値 23

作用ルーチン XPRswriteprtsol (WRITEPRTSOL), XPRswriteprtrange (WRITEPRTRANGE).

解説 並行MIP探索で使用するスレーブプロセッサの数。

タイプ 整数

デフォルト値 0

注意 MIP問題を解くのに有効なプロセッサの数を設定します。

作用ルーチン XPRSglobal (GLOBAL).

解説 Optimizerが終了するまでの動作時間(秒)の最大値。問題のセットアップ時間や解く時間を含みます。MIP問題の場合、全てのノードを解くのに要する総時間です。

タイプ 整数

値 0 制限時間なし
 $n > 0$ 整数解が見つかった場合、 n 秒後にMIP探索を終了するか、最後の解が見つかるとまで継続します
 $n < 0$ LPやMIP探索を n 秒後に停止します

デフォルト値 0

作用ルーチン XPRSglobal (GLOBAL), XPRSmaxim (MAXIM)/XPRSminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 280

MIPABSCUTOFF

- MIPABSSTOP 7

MIPABSCUTOFF

MIPABSSTOP

解説 分枝限定法: 目的関数がある値よりよい値を持つ場合にのみ興味がある場合、MIPABSCUTOFFにより、その値を設定します。Optimizerは探索時間を節約するために、これより悪い目的関数値になるノードについて解きません。これについては、ユーザに前もって設定されない限り、LP Optimizerコマンドの後に自動的に設定されます。カットオフはMIPRELCUTOFFやMIPADDCUTOFF制御によりMIP解が見つかる度に自動的に更新されます。

タイプ 倍精度

デフォルト値 1.0E+40 (最小化問題)
 -1.0E+40 (最大化問題)

作用ルーチン XPRSglobal (GLOBAL), XPRSmaxim (MAXIM)/XPRSminim (MINIM).

MIPRELCUTOFF, MIPADDCUTOFFもご覧ください

解説 分枝限定法: 整数探索を継続するかしないかを定める絶対値許容範囲。MIPOBJVALが最適解の目的関数値で、BESTBOUNDがその時点の最良解であった場合に | MIPOBJVAL - BESTBOUND | MIPABSSTOPであれば探索を終了します。

タイプ 倍精度

デフォルト値 0.0

作用ルーチン XPRSglobal (GLOBAL).

参照 MIPRELSTOP, MIPADDCUTOFF

Xpress-Optimizer Reference Manual Control Parameters 281

MIPADDCUTOFF

- MIPLOG 7

MIPADDCUTOFF

MIPLOG

解説 分枝限定法: 新しいカットオフを与えるために、見つかった最良整数解の目的関数に加える量。一旦、整数解が見つかり、MIPABSCUTOFF以上の目的関数値である場合、少なくともある量より良くない限り、この値の改善には興味がありません。MIPADDCUTOFFが零でない場合、この新しい値よりよい整数解が見つかる度にMIPABSCUTOFFが追加されます。これにより、目的関数において大きな改善がない答を持つ木の部分を切り離して計算時間を節約します。制御MIPABSSTOPは

似たような関数ですが、働きが違います。

タイプ 倍精度

デフォルト値 -1.0E-05

作用ルーチン XPRSglobal (GLOBAL).

MIPRELCUTOFF , MIPABSSTOP , MIPABSSTOPもご覧ください

解説 整数出力制御 .

タイプ 整数

値	0	整数探索の出力なし
	1	終了時に概要のみを出力する
	2	全ての解が見つかったときに詳細ログを出力する
	3	各ノードにおいて詳細ログを出力する
	<i>n</i>	<i>n</i> 番目のノードごとに概要ログを出力する

デフォルト値 -100

作用ルーチン XPRSglobal (GLOBAL).

参照 付録A.7「グローバルログ」

Xpress-Optimizer Reference Manual Control Parameters 282

MIPPRESOLVE

- MIPPRESOLVE 7

MIPPRESOLVE

解説 分枝限定法: 実施される整数処理のタイプ。0に設定されると、何も処理されません。

タイプ 整数

値 ビットに意味があります。

- | | |
|---|-----------------------------------------------------------------------------------------------------------------|
| 0 | 各ノードでリデュースコストの固定化が実行されます。これはノードが解かれる前に単純化するもので、ある変数の値が、このノードで他の変数に設定された追加上下界値に基づいて固定されることを推測して実施されます。 |
| 1 | 各ノードで論理事前処理を行います。これはバイナリ変数について実施され、しばしば制約に基づいて変数を固定します。これは非常に問題を単純化し、シンプレックス法を始める前に、ノードの最適性や実行不能性を決定することさえできます。 |
| 2 | トップノードでバイナリ変数の深さを調査します。これはあるバイナリ変数を設定し、同じ制約内の他のバイナリ変数への影響を推定します。 |

デフォルト値 行列特性によります

作用ルーチン XPRSglobal (GLOBAL).

参照 章 5.2「事前分析のはたらき」, PRESOLVE , PRESOLVEOPS.

Xpress-Optimizer Reference Manual Control Parameters 283

MIPRELCUTOFF

- MIPRELSTOP 7

MIPRELCUTOFF

MIPRELSTOP

解説 分枝限定法: MIPABSCUTOFF の新しい値を設定するために、整数解が見つかったときに目的関数の値に加えるLP解の値の割合。この効果としては、現在の解よりもそれ程良好でない目的関数を持つ可能性が最も高い木の一部分を探索から切り離すことです。制御 MIPRELSTOPは似た機能を持っていますが、働きが違います。

タイプ 倍精度

デフォルト値 1.0E-04

作用ルーチン XPRSglobal (GLOBAL).

参照 MIPABSCUTOFF, MIPADDCUTOFF, MIPRELSTOP.

解説 分枝限定法: これは整数探索を終了するかしないかを決めます。MIPOBJVAL が最適解の目的関数値でBESTBOUND が現在の最良解の上下界値であるとして $|MIPOBJVAL - BESTBOUND|$ MIPRELSTOP BESTBOUND のときには探索を終了します。

タイプ 倍精度

デフォルト値 0.0

作用ルーチン XPRSglobal (GLOBAL).

参照 MIPABSSTOP, MIPRELCUTOFF.

Xpress-Optimizer Reference Manual Control Parameters 284

MIPTARGET

– MPSBOUNDNAME 7

MIPTARGET

MIPTOL

MPSBOUNDNAME

解説 分枝限定法: 整数探索における目的関数の目標 (ノード選択基準により使われる)。これはユーザに事前に設定されない限り、LP最適化ルーチンの後で自動的に設定されます。

タイプ 倍精度

デフォルト値 1.0E+40

作用ルーチン XPRSglobal (GLOBAL).

参照 BACKTRACK.

解説 分枝限定法: 整数条件を考慮した決定変数の値の許容範囲。

タイプ 倍精度

デフォルト値 5.0E-06

作用ルーチン XPRSglobal (GLOBAL).

解説 MPSファイル内で探索された上下界値名。全ての文字列制御を考慮して64文字とNULL終了\0を加えた長さです。

タイプ 文字列

デフォルト値 64ブランク

作用ルーチン XPRSreadprob (READPROB).

Xpress-Optimizer Reference Manual Control Parameters 285

MPSECHO

– MPSFORMAT 7

MPSECHO

MPSERRIGNORE

MPSFORMAT

解説 入力中にMPSコメントをエコーするかの決定。

タイプ 整数

値 0 MPSコメントがエコーされない

1 MPSコメントがエコーされる

デフォルト値 1

作用ルーチン XPRSreadprob (READPROB).

解説 MPSファイルの読み込み中に無視されるエラーの数。

タイプ 整数

デフォルト値 0

作用ルーチン XPRSreadprob (READPROB).

解説 MPS ファイル形式を指定します。

タイプ 整数

値 -1 自動的にファイルタイプを決定します

0 固定形式です

1 入力時にMPSファイルが前提である場合です

デフォルト値 -1

作用ルーチン XPRSalter (ALTER), XPRSreadbasis (READBASIS),

XPRSreadprob (READPROB).

Xpress-Optimizer Reference Manual Control Parameters 286

MPSNAMELENGTH

- MPSRANGENAME **7**

MPSNAMELENGTH

MPSOBJNAME

MPSRANGENAME

解説 MPS名の最大文字列長さ。リセットは問題の入力前に実施されなくてはなりません。内部的には最低8文字の組合せであり、MPS名は右に空白が付けられます。

タイプ 整数

デフォルト値 8

最大値 64

作用ルーチン XPRSaddnames, XPRSreadprob (READPROB)

解説 MPSファイル内で探索される目的関数の名前。全ての文字列制御を考慮して64文字にNULL終了\0を加えた長さです。

タイプ 文字列

デフォルト値 64空白

作用ルーチン XPRSreadprob (READPROB).

解説 MPSファイル内で探索されるレンジ名。全ての文字列制御を考慮して64文字にNULL終了\0を加えた長さです。

タイプ 文字列

デフォルト値 64空白

作用ルーチン XPRSreadprob (READPROB).

Xpress-Optimizer Reference Manual Control Parameters 287

MPSRHSNAME

- OMNIDATANAME **7**

MPSRHSNAME

NODESELECTION

OMNIDATANAME

解説 MPSファイル内を探索される右辺名。全ての文字列制御を考慮して64文字とNULL終了\0を加えた長さです。

タイプ 文字列

デフォルト値 64空白

作用ルーチン XPRSreadprob (READPROB).

解説 分枝限定法: 現在のノードが一度解かれているときに、どのノードを考慮すべきかを決定します。

タイプ 整数

- 値 1 ローカル優先: 使用可能であれば、2つの子ノードから選択します。そうでなければ全てのアクティブなノードから選択します
- 2 最良値優先: 常に全てのアクティブノードが考慮されます
- 3 深さ優先: 使用可能であれば2つの子ノードから選択します。そうでなければ最も深いノードから選択します
- 4 最良値優先とローカル優先: 最初にBREADTHFIRSTノードについて全てのノードを考慮します。その後、ローカル優先探索が使われます

デフォルト値 行列の特性によります

作用ルーチン XPRsglobal (GLOBAL).

解説 OMNIデータ名フィールドのためのデータ。全ての文字列制御を考慮して64文字とNULL終了\0を加えた長さです。

タイプ 文字列

デフォルト値 64ブランク

作用ルーチン XPRswriteomni (WRITEOMNI).

Xpress-Optimizer Reference Manual Control Parameters 288

OMNIFORMAT**- OUTPUTLOG 7****OMNIFORMAT****OPTIMALITYTOL****OUTPUTLOG**

解説 XPRswriteomni (WRITEOMNI) 出力の中に行列係数を含むかを決めます。

タイプ 整数

- 値 0 行列係数は出力に含まれません
- 1 出力に係数を含みます (新形式)
- 2 出力に係数を含みます (旧形式 - 1999年以前)

デフォルト値 0

作用ルーチン XPRswriteomni (WRITEOMNI).

解説 シンプレックス法: 被約費用の零許容範囲。各反復において、シンプレックス法は負のリデュースコストを持つ基底に入れる変数を探します。負のOPTIMALITYTOLの値以下のリデュースコストを持つ変数だけが候補となります。

タイプ 倍精度

デフォルト値 1.0E-06

作用ルーチン XPRsgetinfeas, XPRsmaxim (MAXIM) / XPRsminim (MINIM).

解説 最適化の間、Optimizerによって作られる出力のレベルを制御します。可能なオプションとして、全メッセージを出力するか、全く出力をしないか、が決められます。

タイプ 整数

- 値 0 全ての出力をしません
- 1 メッセージを出力します

デフォルト値 1

作用ルーチン XPRssetcbmessage, XPRswriteprtsol (WRITEPRTSOL).

Xpress-Optimizer Reference Manual Control Parameters 289

OUTPUTMASK**- PENALTY 7****OUTPUTMASK**

OUTPUTTOL PENALTY

解説 ファイルに書かれる行や列の名前を制限するためのマスクです。全ての文字列制御を考慮して64文字とNULL終了\0を加えた長さです。

タイプ 文字列

デフォルト値 64の?

作用ルーチン XPRewriterange (WRITERANGE), XPRwritesol (WRITESOL).

解説 出力値の零許容範囲。

タイプ 倍精度

デフォルト値 1.0E-05

作用ルーチン XPRwriteprtrange (WRITEPRTRANGE), XPRwriteprtsol (WRITEPRTSOL), XPRwriterange (WRITERANGE), XPRwritesol (WRITESOL).

解説 ペナルティ変数係数の絶対値最小値。BIGMとPENALTYは入力ルーチン (XPRsreadprob (READPROB))により設定されますが、XPRsmaxim (MAXIM) / XPRsminim (MINIM) に先立ってユーザにリセットできます。

タイプ 倍精度

デフォルト値 行列の特性によります。

作用ルーチン XPRsmaxim (MAXIM) / XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 290

PERTURB

- PPFACOR 7

PERTURB

PIVOTTOL

PPFACTOR

解説 制御AUTOPERTURBが1に設定されているとき、最適化に先立って実行される摂動問題の係数 (摂動値) です。0.0に設定されているときは摂動値は自動的に決められます。

タイプ 倍精度

デフォルト値 0.0 — デフォルトでは自動的に摂動値が決められます

作用ルーチン XPRsmaxim (MAXIM) / XPRsminim (MINIM).

解説 シンプレックス法: 行列要素の零許容範囲。各繰返しにおいて、シンプレックス法はピボットする非零行列要素を探します。PIVOTTOLより小さい絶対値の要素は零として扱われます。

タイプ 倍精度

デフォルト値 1.0E-09

作用ルーチン XPRsmaxim (MAXIM) / XPRsminim (MINIM), XPRSpivot.

解説 パラメータをサイジングする部分プライシング候補リスト。

タイプ 倍精度

デフォルト値 1.0

作用ルーチン XPRsmaxim (MAXIM) / XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 291

PRESOLVE

- PRESOLVE 7

PRESOLVE

解説 メインアルゴリズムの開始に先立って事前分析を実施するかを決定する制御。事前分析は退化制約を検出して取り除いたり、変数上下界値を強化する等して問題を単純化します。場合によっ

では、この段階で実行不能性さえも決定できたり最適解が見つかります。

タイプ 整数

値	-1	事前分析が実施されませんが、主実行不能性が検出される際、問題が実行不能と宣言されません。問題はLP最適化アルゴリズムで解かれ、実行不能解が返されますが、このことは時に有用です
	0	事前分析は実施されません
	1	事前分析は実施されます
	2	事前分析は実施されませんが、退化上下界値は取り除かれませんがこれはバリア法の有効性を時々向上させます

デフォルト値 1

注意 事前分析のためのメモリはダイナミックにリサイズされます。Optimizerが事前分析するのにメモリ不足の場合、エラーメッセージ245が作成されます。

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

参照 章5.2「事前分析のはたらき」、PRESOLVEOPS, MIPPRESOLVE.

Xpress-Optimizer Reference Manual Control Parameters 292

PRESOLVEOPS

- PRICINGALG 7

PRESOLVEOPS

PRICINGALG

解説 事前分析中に実施されるオペレーションを指定します。

タイプ 整数

値		ビットの意味は以下の通りです。
	0	シングルトン列が除かれます
	1	シングルトン行が除かれます
	2	行の強制削除
	3	双対縮小
	4	退化行削除
	5	二重列削除
	6	二重行削除
	7	強制双対縮小
	8	変数消去
	9	IP縮小なし

デフォルト値 511 (0 - 8のビットに設定)

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

参照 章5.2「事前分析のはたらき」、PRESOLVE, MIPPRESOLVE.

解説 シンプレックス法: 各繰返しにおいて使うプライシング法を決定し、基底に入れる変数を選びます。一般に、Devexプライシングは各繰返しで多くの時間を要しますが、繰返し回数は減ります。一方、部分プライシングは各繰返しの時間は節約できますが、繰返し回数が増える可能性があります。

タイプ 整数

値	-1	部分プライシングが使われます
	0	自動的にプライシング方法が決まります
	1	Devexプライシングが使われます

デフォルト値 0

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 293
PSEUDOCOST

– RECMAXPASSES **7**

PSEUDOCOST

RECEXPAND

RECMAXPASSES

解説 分枝限定法: 木探索における未発見ノードに関する推定劣化量の算定に使われる擬コストのデフォルト値。擬コストは各整数決定変数に関係し、その値を整数変数にした場合の目的関数の値の悪化量の推定値です。

タイプ 倍精度

デフォルト値 0.01

作用ルーチン XPRSglobal (GLOBAL), XPRSreaddir (READDIRS).

解説 反復におけるBX係数とSB上下界値の改善倍率。

タイプ 倍精度

デフォルト値 1.1

作用ルーチン XPRSrecurse (RECURSE).

解説 反復ループの実施最大数。

タイプ 整数

デフォルト値 20

作用ルーチン XPRSrecurse (RECURSE).

Xpress-Optimizer Reference Manual Control Parameters 294

RECSHRINK

– RECSTOP **7**

RECSHRINK

RECSTEPLength

RECSTOP

解説 反復におけるBX係数とSB上下界値の減少倍率。

タイプ 倍精度

デフォルト値 0.9

作用ルーチン XPRSrecurse (RECURSE).

解説 反復ステップの長さ。ステップ長さ1.0は時々大きすぎて収束しないことがあります。その場合、パラメータRECSTEPLengthを減らす必要があります。

タイプ 倍精度

デフォルト値 1.0

作用ルーチン XPRSrecurse (RECURSE).

解説 反復収束パラメータ。

タイプ 倍精度

デフォルト値 1.0E-05

作用ルーチン XPRSrecurse (RECURSE).

Xpress-Optimizer Reference Manual Control Parameters 295

REFACTOR

– REL10STYLE **7**

REFACTOR

REL10STYLE

解説 メモリ内の現在の因子分解の再表現を使って最適化を再スタートするかを指定します。

タイプ 整数

値 0 再最適化において再因子分解しない

1 再最適化において再因子分解する

デフォルト値 0 整数探索

1 再最適化

注意 木探索において、ノードにおける最適基底は、デフォルトでは再因子分解されませんが、LP問題の最適基底は再因子分解されます。LP問題を少しの変更だけで繰り返し解く場合、REFACTORを0に設定することがより効果的です。

作用ルーチン XPRsglobal (GLOBAL), XPRsmaxim (MAXIM)/XPRsminim (MINIM).

解説 双対変数やスラック、被約費用に旧型表記を使用するかを決めます。

タイプ 整数

値 0 解の値に標準表記を使います

1 リリース10より前の表記を解の値に使います

デフォルト値 0

作用ルーチン XPRsgetsol, XPRswriteprtsol (WRITEPRTSOL), XPRswritesol (WRITESOL).

Xpress-Optimizer Reference Manual Control Parameters 296

RELPIVOTTOL

- SBITERLIMIT 7**RELPIVOTTOL****SBBEST****SBITERLIMIT**

解説 シンプレックス法: 各反復において、ピボット要素は行列の与えられた列の中で選ばれます。相対ピボット許容範囲RELPIVOTTOLは、同じ列で最大の可能なピボット要素に関して選ばれます。

タイプ 倍精度

デフォルト値 1.0E-06

作用ルーチン XPRsmaxim (MAXIM)/XPRsminim (MINIM), XPRSpivot.

解説 強力分枝を実施する実行不能離散要素の数。

タイプ 整数

デフォルト値 -1 — 自動的に決められます。

作用ルーチン XPRsglobal (GLOBAL).

解説 強力分枝を実施するための双対繰り返し数。

タイプ 整数

デフォルト値 0

作用ルーチン XPRsglobal (GLOBAL).

Xpress-Optimizer Reference Manual Control Parameters 297

SCALING

- SOLUTIONFILE 7**SCALING****SOLUTIONFILE**

解説 最適化の前にOptimizerが内部的にモデルをリスケールする方法を決定します。0に設定されるとスケールリングは実行されません。

タイプ 整数

値 ビットの意味は以下の通りです。

0	行スケーリング
1	列スケーリング
2	再度の行スケーリング
3	最大
4	Curtis-Reid
5	0: 幾何平均によるスケーリング 1: 最大要素によるスケーリング

デフォルト値 35

作用ルーチン XPRSloadglobal, XPRSloadlp, XPRSloadqglobal,
XPRSloadqp, XPRSreadprob (READPROB), XPRSscale (SCALE).

参照 章3.4「スケーリング」

解説 解ファイルから(へ)の解の読み(書き)を実施するかを制御します。

タイプ 整数

値	0	メモリから(へ)の解の読み(書き)
	1	ファイルから(へ)の解の読み(書き)

デフォルト値 1

作用ルーチン XPRSgetbasis, XPRSgetinfeas, XPRSgetsol, XPRSmaxim
(MAXIM)/XPRSminim (MINIM).

Xpress-Optimizer Reference Manual Control Parameters 298

SOSREFTOL

– TREECOVERCUTS **7**

SOSREFTOL

TRACE

TREECOVERCUTS

解説 特殊順序集合における要素の順序値間の最小ギャップ。

タイプ 倍精度

デフォルト値 1.0E-03

作用ルーチン XPRSloadglobal, XPRSloadqglobal, XPRSreadprob
(READPROB).

解説 事前分析での実行不能診断の制御— 零でなければ実行不能性が説明されます。

タイプ 整数

デフォルト値 0

作用ルーチン XPRSmaxim (MAXIM)/XPRSminim (MINIM).

解説 分枝限定法: 木のトップノードでないノードで生成された採用被覆不等式の使用回数。

COVERCUTSの解説と比較してください。

タイプ 整数

デフォルト値 2

作用ルーチン XPRSglobal (GLOBAL).

Xpress-Optimizer Reference Manual Control Parameters 299

TREGOMCUTS

– VERSION **7**

TREGOMCUTS

VARSELECTION

VERSION

解説 分枝限定法: 木の最初のノードでないノードで生成された Gomory カットの使用回数。GOMCUTS の解説と比べてください。

タイプ 整数

デフォルト値 0

作用ルーチン XPRSglobal (GLOBAL).

解説 分枝限定法: 与えられたノードの探索からの整数解において期待される目的関数の全推定劣化量を得るための整数変数に関する擬コストの組合せ方を決めます。BACKTRACKが1に設定されているときにのみ有効です。

タイプ 整数

値	1	上下側擬コストの最小値の和
	2	上下側擬コストの総和
	3	上下側擬コストの最大値と最小値の2倍の和
	4	上下側擬コストの最大値の和
	5	下側擬コストの和
	6	上側擬コストの和

デフォルト値 1

作用ルーチン XPRSglobal (GLOBAL).

解説 Optimizerのバージョン番号e.g.1301 はリリース13.01を表します。

タイプ 整数

Xpress-Optimizer Reference Manual Control Parameters 300

VERSION

- VERSION 7

Xpress-Optimizer Reference Manual Retrieving Problem Attributes 301

Problem Attributes 8

8 問題特性

最適化においては、解かれた問題の特性が保存され、Xpress-MP ライブラリユーザは *problem attributes* (問題特性) の形で利用できます。これらについては、制御と同様の方法でアクセスできます。問題特性の例は、配列の大きさを含みます。よって、ライブラリユーザは、配列が検索される前にスペースを割り当てる必要があるかもしれません。利用可能な特性やタイプについて、本章に全リストを示します。

8.1 問題特性の検索

ライブラリユーザは、特性値を得るために次のような3つの関数を使えます。

先の制御と同じように、Xpress-MP ライブラリユーザの場合には、ここに示された特性にプレフィクスXPRS_を付ける必要があり、そうしないとエラーになることに注意してください。これらの使い方の例は、次のようにLP問題を解いた後に目的関数の最適値を返して出力します。

```
XPRSgetdblattrib(prob, XPRS_LPOBJVAL, &lpobjval);
printf("The objective value is %2.1f\n", lpobjval);
XPRSgetintattrib XPRSgetdblattrib XPRSgetstrattrib
```

Xpress-Optimizer Reference Manual Problem Attributes 302

ACTIVENODES

- BARDENSECOL 8

ACTIVENODES

BARAASIZE

BARCROSSOVER

BARDENSECOL

解説 未解決ノード数。

タイプ 整数

ルーチンによる設定 XPRSdelnode, XPRSglobal, XPRSinitglobal.

解説 AAT内の非零要素数

タイプ 整数

ルーチンによる設定 XPRSmaxim / XPRSminim.

解説 基底クロスオーバーフェーズが入力済みかを示します。

タイプ 整数

値	0	クロスオーバーフェーズ入力済みではありません
	1	クロスオーバーフェーズ入力済みです

ルーチンによる設定 XPRSmaxim / XPRSminim.

解説 行列内で見つかった密な列の数

タイプ 整数

ルーチンによる設定 XPRSmaxim / XPRSminim.

Xpress-Optimizer Reference Manual Problem Attributes 303

BARDUALINF

– BARPRIMALINF 8

BARDUALINF**BARDUALOBJ****BARITER****BARLSIZE****BARPRIMALINF**

解説 ニュートンバリア法での双対実行不能の合計。

タイプ 倍精度

ルーチンによる設定 XPRSmaxim / XPRSminim.

解説 ニュートンバリア法により計算された双対目的関数値。

タイプ 倍精度

ルーチンによる設定 XPRSmaxim / XPRSminim.

解説 ニュートンバリア法の繰り返し数。

タイプ 整数

ルーチンによる設定 XPRSmaxim / XPRSminim.

解説 コレスキー因子分解の結果についてのL内の非零数。

タイプ 整数

ルーチンによる設定 XPRSmaxim / XPRSminim.

解説 ニュートンバリア法での主実行不能の和。

タイプ 倍精度

ルーチンによる設定 XPRSmaxim / XPRSminim.

Xpress-Optimizer Reference Manual Problem Attributes 304

BARPRIMALOBJ

– BOUNDNAME 8

BARPRIMALOBJ**BARSTOP****BESTBOUND****BOUNDNAME**

解説 ニュートンバリア法により計算された主目的関数値。

タイプ 倍精度

ルーチンによる設定 XPRSmaxim / XPRSminim.

解説 ニュートンバリア法の収束基準。

タイプ 倍精度

ルーチンによる設定 XPRSmaxim / XPRSminim.

解説 整数探索により決定された最良上下界値の値。

タイプ 倍精度

ルーチンによる設定 XPRSglobal.

解説 アクティブな上下界値名。

タイプ 文字列

ルーチンによる設定 XPRSreadprob.

Xpress-Optimizer Reference Manual Problem Attributes 305

COLS

– ELEMS **8**

COLS

CUTS

DUALINFEAS

ELEMS

解説 行列内の列の数。

タイプ 整数

ルーチンによる設定 XPRSloadglobal, XPRSloadlp, XPRSloadqglobal, XPRSloadqp, XPRSreadprob.

解説 行列への追加カットの数。

タイプ 整数

ルーチンによる設定 XPRSaddcuts, XPRSdelcpcuts, XPRSdelcuts, XPRSloadcuts, XPRSloadmodelcuts.

解説 双対実行不能の数。

タイプ 整数

ルーチンによる設定 XPRSmaxim / XPRSminim.

参照 PRIMALINFEAS

解説 行列の非零要素の数。

タイプ 整数

ルーチンによる設定 XPRSloadglobal, XPRSloadlp, XPRSloadqglobal, XPRSloadqp, XPRSmaxim / XPRSminim, XPRSreadprob.

Xpress-Optimizer Reference Manual Problem Attributes 306

ERRORCODE

– LPOBJVAL **8**

ERRORCODE

IIS

LPOBJVAL

解説 一番最近に発生したOptimizer エラー番号。

タイプ 整数

ルーチンによる設定 どれでも

解説 発見されたIISの数。

タイプ 整数

ルーチンによる設定 XPRSsiis.

解説 最後に解かれたLPの目的関数値。

タイプ 倍精度

ルーチンによる設定 XPRSmaxim / XPRSminim, XPRSglobal.

参照 MIPOBJVAL, OBJRHS, OBJFIXED

Xpress-Optimizer Reference Manual Problem Attributes 307

LPSTATUS

– MATRIXNAME 8

LPSTATUS

MATRIXNAME

解説 LP解の状態。

タイプ 整数

値	XPRS_LP_OPTIMAL	最適
	XPRS_LP_INFEAS	実行不能
	XPRS_LP_CUTOFF	目的関数がカットオフより悪い
	XPRS_LP_UNFINISHED	未完了
	XPRS_LP_UNBOUNDED	非有界
	XPRS_LP_CUTOFF_IN_DUAL	双対でのカットオフ

注意 Optimizer CヘッダファイルとVB.bas ファイル中で定数として定義される可能な戻り値。上記に示したりターンコードについてのファイルの1つを参照ください。

ルーチンによる設定 XPRSmaxim / XPRSminim.

参照 MIPSTATUS

解説 行列名。

タイプ 文字列

注意 MPS 行列内のMATRIX フィールドから読まれる名前であり、Optimizer 内で使われる問題名とは関係ありません。問題名を得るにはXPRSgetprobnameを使ってください。

ルーチンによる設定 XPRSreadprob, XPRSetprobname.

Xpress-Optimizer Reference Manual Problem Attributes 308

MIPENTS

– MIPSOLNODE 8

MIPENTS

MIPINFEAS

MIPOBJVAL

MIPSOLNODE

解説 離散要素の数(つまり、バイナリ, 整数, 半連続, 整数, 半連続整数変数)。但し、集合の数を除きます。

タイプ 整数

ルーチンによる設定 XPRSaddcols, XPRSchgcoltype, XPRSdelcols, XPRSloadglobal, XPRSloadqglobal, XPRSreadprob.

参照 SETS

解説 現在のノードでの整数実行不能の数。

タイプ 整数

ルーチンによる設定 XPRSglobal.

参照 PRIMALINFEAS

解説 見つかった最良整数解の目的関数値。

タイプ 倍精度

ルーチンによる設定 XPRSglobal.

参照 LPOBJVAL

解説 最後に見つかった整数実行可能解のノード。

タイプ 整数

ルーチンによる設定 XPRSglobal.

Xpress-Optimizer Reference Manual Problem Attributes 309

MIPSOLS

– MIPSTATUS 8

MIPSOLS

MIPSTATUS

解説 見つかった整数解の数。

タイプ 整数

ルーチンによる設定 XPRSglobal.

解説 整数(混合型整数)問題の解の状態。

タイプ 整数

値	XPRS_MIP_LP_NOT_LOADED	問題がロードされていません
	XPRS_MIP_LP_NOT_OPTIMAL	LP最適化がなされていません
	XPRS_MIP_LP_OPTIMAL	LP最適化済みです。MIP最適化が実施されれば、次の4つのどれかの状態コードが返されます
	XPRS_MIP_NO_SOL_FOUND	整数探索未終了— 整数解が見つかりません
	XPRS_MIP_SOLUTION	整数探索未終了— 整数解は見つかっています
	XPRS_MIP_INFEAS	整数探索終了— 整数解は見つかりませんでした
	XPRS_MIP_OPTIMAL	整数探索終了— 整数解が見つかりました

注意

1. XPRS_MIP_LP_OPTIMAL 状態コードが返されると、LP最適化の後か最中に、最適化が停止したことを示します。例えば、LP緩和が実行不能や非有界の場合です。この場合、LP解の状態の値をLPSTATUSを使って確認してください。
2. Optimizer CヘッダファイルとVB.bas ファイル中の定数として可能な戻り値が定義されます。上に示したリターンコードの値についてのファイルの1つを参照ください。

ルーチンによる設定 XPRSglobal, XPRSloadglobal, XPRSloadqglobal, XPRSmaxim / XPRSminim, XPRSreadprob.

参照 LPSTATUS

Xpress-Optimizer Reference Manual Problem Attributes 310

NAMELENGTH

– OBJFIXED 8

NAMELENGTH

NODEDEPTH

NODES

OBJFIXED

解説 行列内の行や列の名前の長さ(8文字の組)。名前を保存する文字配列を割り当てるためには、1つの名前(+1は文字列終了文字のため)あたり $8 * \text{NAMELENGTH} + 1$ 文字を許容しなくてはなりません。

タイプ 整数

ルーチンによる設定 XPRSloadglobal, XPRSloadlp, XPRSloadqglobal, XPRSloadqp, XPRSreadprob.

解説 現在のノードの深さ。

タイプ 整数

ルーチンによる設定 XPRSglobal, XPRSinitglobal.

解説 整数探索において解かれたノードの数。ノード番号は分枝限定木で最初（トップ）のノードが1で始まります。ノードは連続的に番号が付けられます。

タイプ 整数

ルーチンによる設定 XPRSglobal, XPRSinitglobal.

解説 人為（固定された）変数の目的関数への寄与。

タイプ 倍精度

ルーチンによる設定 XPRSmaxim / XPRSminim.

参照 LPOBJVAL, OBJRHS

Xpress-Optimizer Reference Manual Problem Attributes 311

OBJNAME

- PARENTNODE 8

OBJNAME

OBJRHS

OBJSENSE

PARENTNODE

解説 アクティブな目的関数行の名前。

タイプ 文字列

ルーチンによる設定 XPRSreadprob.

解説 目的関数の定数項部分。目的関数の定数項部分。

タイプ 倍精度

ルーチンによる設定 XPRSchgobj

参照 LPOBJVAL, OBJFIXED

解説 実施された最適化の方向。

タイプ 倍精度

値 1.0 最小化問題

-1.0 最大化問題

ルーチンによる設定 XPRSmaxim / XPRSminim.

解説 木探索における現在のノードの親ノード。

タイプ 整数

ルーチンによる設定 XPRSglobal, XPRSinitglobal.

Xpress-Optimizer Reference Manual Problem Attributes 312

PRESOLVSTATE

- RANGENAME 8

PRESOLVSTATE

PRIMALINFEAS

QLEMS

RANGENAME

解説 ビットマップで表される問題の状態。

タイプ 整数

値 ビットの意味

- 0 問題はロードされました
- 1 問題はLP事前分析されました
- 2 問題はMIP事前分析されました
- 7 メモリ内の解は有効です

注意 他のビットは予備です。

ルーチンによる設定 XPRsmaxim / XPRsminim.

解説 主実行不能の数。

タイプ 整数

ルーチンによる設定 XPRsmaxim / XPRsminim.

参照 SUMPRIMALINF, DUALINF, MIPINF

解説 行列内の 2 次要素の数。

タイプ 整数

ルーチンによる設定 XPRSchgmqobj, XPRSchgqobj, XPRsloadqglobal, XPRsloadqp.

解説 アクティブレンジ名。

タイプ 文字列

ルーチンによる設定 XPRsreadprob.

Xpress-Optimizer Reference Manual Problem Attributes 313

RECCONVERGE

– **SIMPLEXITER** 8

RECCONVERGE

RHSNAME

ROWS

SIMPLEXITER

解説 XPRsrecurse が収束したかを示します。

タイプ 整数

値	0	XPRsrecurse は収束しました
	>0	XPRsrecurse は収束していません。収束していない項の番号を返します
	-1	XPRsrecurse は目的関数収束試験によると収束していません

ルーチンによる設定 XPRsrecurse.

解説 アクティブ右辺の名前。

タイプ 文字列

ルーチンによる設定 XPRsreadprob.

解説 行列内の行の数。

タイプ 整数

ルーチンによる設定 XPRsaddrows, XPRsdelrows, XPRsloadglobal, XPRsloadlp,
XPRsloadqglobal, XPRsloadlp, XPRsreadprob.

解説 実施されたシンプレックス繰返しの数。

タイプ 整数

ルーチンによる設定 XPRsmaxim / XPRsminim.

Xpress-Optimizer Reference Manual Problem Attributes 314

SETMEMBERS

– **SPAREELEMS** 8

SETMEMBERS

SETS

SPARECOLS

SPAREELEMS

解説 行列内の集合要素の数。

タイプ 整数

ルーチンによる設定 XPRsloadglobal, XPRsloadqglobal, XPRsreadprob.

参照 SETS

解説 行列内の集合の数。

タイプ 整数

ルーチンによる設定 XPRsloadglobal, XPRsloadqglobal, XPRsreadprob.

参照 SETMEMBERS, MIPENTS

解説 行列内の予備列の数。

タイプ 整数

ルーチンによる設定 XPRsloadglobal, XPRsloadlp, XPRsloadqglobal, XPRsloadqp, XPRsreadprob.

解説 行列内の予備行列要素の数。

タイプ 整数

ルーチンによる設定 XPRsloadglobal, XPRsloadlp, XPRsloadqglobal, XPRsloadqp, XPRsreadprob.

Xpress-Optimizer Reference Manual Problem Attributes 315

SPAREMIPENTS**-SUMPRIMALINF 8****SPAREMIPENTS****SPAREROWS****SUMPRIMALINF**

解説 行列内の予備離散要素の数。

タイプ 整数

ルーチンによる設定 XPRsloadglobal, XPRsloadlp, XPRsloadqglobal, XPRsloadqp, XPRsreadprob.

解説 行列内の予備行の数。

タイプ 整数

ルーチンによる設定 XPRsloadglobal, XPRsloadlp, XPRsloadqglobal, XPRsloadqp, XPRsreadprob.

解説 主実行不能のスケーリングされた和。

タイプ 倍精度

ルーチンによる設定 XPRsmaxim / XPRsminim.

参照 PRIMALINFEAS

Xpress-Optimizer Reference Manual Problem Attributes 316

SUMPRIMALINF**-SUMPRIMALINF 8**

Xpress-Optimizer Reference Manual Return Codes for the Xpress-MP Optimizer 317

Error Messages &**Return Codes 9****9 エラーメッセージ & リターンコード****9.1 Xpress-MP Optimizerのリターンコード**

Optimizerが終了するとき、OSや呼びだしたプログラムによってチェック可能な終了コードが出力

されます。終了コードは次のように設定されます。

終了コード63~99は、STOPコマンドでOptimizerを終了した際に出力され、コンソールユーザにだけ重要です。

リターンコード	内容
0	プログラムは正常に終了(QUITを使用)
1a	
a.	異常終了を引き起こすような致命的エラー。入力が間違っています。
2a	不良あるいは壊れたファイル - 回復不可能
4a	メモリエラー
8a	使い方が間違っています
16a	プログラムエラー
32	呼びだし引数が無効です
63	LP最適化が終了していません
64	LP実行可能で最適解が得られました
65	LP実行不能です
66	LP非有界です
67	IP最適解が得られました
68	IP最適化が未終了ですがIP解は見つかっています
69	IP最適化が未終了でIP解は見つかりません
70	IP実行不能です
99	LP最適化がスタートしていません
128a	コプロセッサが見つかりません

Xpress-Optimizer Reference Manual Error Messages & Return Codes 318

Error Codes 3-99

9.2 Optimizer エラーコード

終了に続いて、発生したエラーについての必要な情報を得るためにOptimizerに問い合わせることができます。ライブラリユーザは、関数XPRSgetlasterrorによりエラーの解説を得られます。この関数は、エラーコードに関連する情報、問題属性ERRORCODEにある情報を返します。コンソールユーザは、遭遇したエラーを示す特性の値が画面に表示されます。ライブラリユーザは次のように検索します。

```
XPRgetintattrib(prob, XPRS_ERRORCODE, &errorcode);
```

ERRORCODEの値や解決法については以下のリストをご覧ください。

3 Extension not allowed — ignored.

指定した拡張子は許容外です。Optimizerはその拡張子を無視し、ファイル名を縮めます。

4 Column <col> has no upper bound.

列 <col> には基底における上限値が示されていません。残りの行と列は基底 / 非基底状態を維持し、列 <col> が下限値に基づいて内部的に新たな基底が作られます。

5 Error on .<ext> file.

エラーが.<ext>ファイルで発生しました。ディスクにスペースが十分あり、壊れていないことを確認して下さい。

6 No match for column <col> in matrix.

列 <col> が、行列のCOLUMNSセクションに定義されていないため、次のセクションで使えません。<col>のスペルが正しいことと列名フィールドの外に書かれていないかをチェックして下さい。

7 Empty matrix. Please increase EXTRAROWS.

列や行がほとんどありません。入力の前にEXTRAROWSを増やすか行列に少なくとも1行あることを確認して再度読みなおして下さい。

9 Error on read of basis file.

基底ファイル .BSS が壊れています。ファイルに十分なスペースがディスクにあり、壊れていないことを確認して下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 319

Error Codes 11 – 36 9**11 Not allowed — solution not optimal.**

解が最適値でないと、実行しようとしている操作ができません。XPRsminim (MINIM)あるいはXPRsmaxim (MAXIM)を呼びだして問題を最適化し、最適化が終了するのを確認して下さい。制御パラメータLPITERLIMITが設定されているときは、認められている反復の最大回数内で最適解を見つけられることを確認してください。

16 Null column <col>.

列<col>においてCOLUMNセクション内の全ての行が零係数です。警告メッセージをなくするために空の列を取り除いてください。

17 More than one RHS not permitted — subsequent ones ignored.

1つの制約には1つのRHSのみ認められています。読みこまれた最初のRHS値は保存されますが、その後のものは無視されます。

18 Bound conflict for column <col>.

列<col>に指定した上限が指定した下限よりも小さいです。矛盾のないよう、どちらか、あるいは両方を変更してもう一度やり直してください。

19 Eta overflow straight after invert — unrecoverable.

イータ配列のためのメモリが十分にありません。仮想ページングスペースか物理メモリのどちらかを増やしてください。

20 Insufficient memory for array <array>.

初期データ構造のためのメモリが十分にありません。仮想ページングスペースか物理メモリのどちらかを増やしてください。

21 Unidentified section

そのコマンドはOptimizerに認識されません。スペルを確認してもう一度やり直して下さい。リファレンスマニュアルの有効なコマンドのリストを参照ください。

29 Input aborted.

行列を読みこむ際にあまりにも多くの問題があったため中止されました。このメッセージは問題の性質についての情報を与えるエラー番号の他のエラーメッセージより先に出ます。全てのエラーを直してもう一度やり直してください。

36 Linear Optimizer only: buy IP Optimizer from Dash Associates.

線形Optimizerだけが使用できます。このコマンドを使用したい場合は、IPOptimizerへのアップグレードについてお近くの営業所にお問い合わせ下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 320

Error Codes 38 – 60 9**38 Invalid option.**

指定したオプションが正しくありません。入力したオプションをチェックし、コマンドを再入力して下さい。各コマンドの有効なオプションのリストは、6章「コンソールとライブラリ関数」をご覧ください。

41 Global error — contact Dash.

内部エラー。Dashに連絡して下さい。

45 Failure to open global file — aborting. (Perhaps disk is full).

Xpress-MPは.GLBファイルを開くことができません。通常ディスクが一杯のときに発生します。違う原因であれば.GLBファイルが壊れています。

50 Inconsistent basis.

メモリ内の内部基底が壊れています。Dash に連絡下さい。

52 Too many nonzero elements.

行列要素数が許容最大数を超過しています。Hyperバージョンをお持ちの場合は、仮想ページスペースか、物理メモリを増やしてください。他のバージョンのソフトウェアをお持ちの場合は、お近くの営業所に、この要素数を使用できるアップグレードについてお問い合わせ下さい。

54 XPRSrecurse (RECURSE) did not converge in <num> loops.

反復が収束しませんでした。収束するとは保証できませんが、より多くの反復要素を使用できるようにRECMAXPASSESを増やして下さい。

56 Reference row entries too close for set <set> member <col>.

集合<set>の参照行として使われる制約条件における列<col>の係数は、参照行の他の列の係数と非常に似ています。参照行の中の係数がお互い十分に異なっていることを確認して下さい。これを行う方法の1つは、集合<set>の全ての変数要素を含み、係数の間隔が少なくとも1単位であるよう指定した計算制約 (N型) でない制約を作成することです。

58 Duplicate element for column <col> row <row>.

列<col> の係数は、行<row>で2回以上現れます。その要素は一緒に加えられますが、この警告メッセージを出さないために列<col>は行<row>の中で1つの係数だけとして下さい。

60 Out of memory — program aborted.

Optimizerはこれ以上のメモリを割り当てられません。仮想ページスペースか物理メモリを増やして下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 321

Error Codes 61 – 76 **9**

61 Unexpected EOF on workfile.

内部ワークファイルが壊れています。ディスクに十分なスペースがあることを確認してもう一度やってみてください。解決しない場合は、Dash にご連絡下さい。

64 Error closing file <file>.

Xpress-MPはファイル<file>を閉じることができません。そのファイルが存在することと、他のアプリケーションで使用していないことを確認して下さい。

65 Fatal error on read from workfile <file> — program aborted.

内部ワークファイルが壊れています。ディスクに十分なスペースがあることを確認してもう一度やってみてください。解決しない場合は、Dash にご連絡下さい。

66 Unable to open file <file>.

Xpress-MPはファイル<file>を開くことができませんでした。そのファイルが存在することと、ディスクに十分なスペースがあることを確認して下さい。

67 Error on read of file <file>.

Xpress-MPはファイル<file>を読むのに失敗しました。ファイルが存在することと、壊れていないことを確認して下さい。

68 <num> errors in sizing parameter <par> — fatal.

初期化中にXpress-MP が<num> エラーに遭遇しました。すぐにお近くの営業所にお問い合わせ下さい。

71 Not a basic vector: <vector>.

行あるいは列<vector>の双対値が分析できません。そのベクトルが基底でないためです。

72 Not a non-basic vector: <vector>.

行あるいは列<vector>のアクティビティが分析できません。そのベクトルが基底でないためです。

73 Problem has too many rows. The maximum is <num>.

行の数が可能最大数<num>を超過しているため、Xpress-MPは問題を入力できません。Hyperバージョン以上を購入するなら、アップグレードによってこの行数の行列を読めるかをお近くの営業所にお問い合わせ下さい。

76 Illegal priority: entity <ent> value <num>.

要素<ent>がディレクティブファイル中の<num>の値の無効な優先度を指定しているため、この優先度は無視されます。優先度の値が0 ~ 1000の範囲内であることと、.DIRファイルの正しいフィールド内に書き込まれていることを確認して下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 322

Error Codes 77 – 90**77 Illegal set card <line>.**

.MAT ファイルあるいは.MPSファイルの<line>行目に定義された集合が矛盾しています。集合が正しいタイプで既に定義されていないことを確認して下さい。有効な集合タイプのリストについてはリファレンスマニュアルをご覧ください。

79 File error.

Optimizerはファイルエラーに遭遇しました。ディスクに十分なスペースがあることと、そのボリュームが壊れていないことを確認して下さい。

80 File creation error.

Optimizerはファイルを作成できません。ディスクに十分なスペースがあることと、そのボリュームが壊れていないことを確認して下さい。

81 Fatal error on write to workfile <file> — program aborted.

Optimizerはファイル<file>に書きこむことができません。ディスクに十分なスペースがあることと、そのボリュームが壊れていないことを確認して下さい。

83 Fatal error on write to file — program aborted.

Optimizerは内部ファイルに書きこむことができません。ディスクに十分なスペースがあることと、そのボリュームが壊れていないことを確認して下さい。

84 Input line too long. Maximum line length is <num>.

.MATファイル あるいは .MPSファイルの1行が長すぎます。 <num> 以下の長さに減らして再入力して下さい。

85 File not found: <file>.

Optimizerはファイル<file>を見つけれません。スペルとそのファイルが存在するかをチェックして下さい。このファイルをXpress-MPで作成した場合は、その作成過程が実行されたことを確認して下さい。

89 No optimization has been attempted.

解が最適値でない限り、実行しようとしているオペレーションは認められません。最適化を行うためにXPRsminim (MINIM)あるいはXPRsmaxim (MAXIM)を呼びだし、プロセスが終了することを確認して下さい。制御LPITERLIMITを設定した場合は、最大反復数内で最適解を見つけられることを確認してください。

90 Not enough memory for devex pricing: PRICINGALG has been set to -1.

Optimizerは、Devex プライシングを実行するために更にメモリが必要です。仮想ページングスペースあるいは物理メモリを増やすか、Devex プライシングアルゴリズムを使用しないようにして下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 323

Error Codes 91 – 114**91 No problem has been input.**

入力済みの問題を要求しました。再度実行する前に、XPRsreadprob (READPROB)を呼びだして問題が正しくロードされていることを確認してください。

95 SB recursion entries cannot be global entities <ent>.

離散要素は反復では使用できません。もう一度やり直す前に要素<ent>をXPRsrecurse (RECURSE)セクションから取り除くか、上下境界値タイプを連続に変更してください。

97 Split vector <vector>.

.MATファイルや.MPSファイルのCOLUMNセクションで列<vector>は、連続した行の中で宣言されなければなりません。別のベクトルに相当するラインで列の宣言を中断することはできません。

98 At line <num> no match for row <row>.

.MATファイルあるいは.MPSファイルの<num>行目で使用される行<row>は存在しません。スペルのチェックと、ROWSセクションに<row>が定義されていることを確認して下さい。

102 Eta file space exceeded — optimization aborted.

Optimizerは更に多くのメモリを必要としています。仮想ページングスペースか物理メモリを増やして最適化を再実行して下さい。

107 Too many global entities at column <col>.

離散要素の数が許容最大数を超過しているため、Xpress-MPIは問題を入力できません。Hyperバージョンなら、仮想ページングスペースか物理メモリを増やしてください。他のバージョンをお持ちの場合は、この離散要素数の行列の読みこみについてのアップグレードを、お近くの営業所にお問い合わせ下さい。

111 Duplicate row <row> — ignored.

行<row>は、同一セクションで2度以上使用されています。最初のだけが使われ2度目以降については無視されます。

112 Postoptimal analysis not permitted on presolved problems.

PRESOLVE = 0 で再最適化してください。事前分析された問題上で操作しようとした。フラグにを設定したXPRsminim (MINIM)/XPRsmaxim (MAXIM)を呼び出すか、PRESOLVEに0を設定して事前分析をしないようにして再度最適化を行ってください。

114 Fatal error — pool hash table full at vector <vector>.

内部エラー。Dash に連絡して下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 324

Error Codes 120 – 142 **9**

120 Problem has too many rows and columns. The maximum is <num>.

行と列の和が許容最大数を超過しているため、Xpress-MPIは問題を入力できません。Hyperバージョン以外を購入した場合には、このサイズの行列の読みこみに関するアップグレードについてお近くの営業所にお問い合わせ下さい。

122 Corrupt solution file.

解ファイル.SOL にアクセスできませんでした。ディスクに十分なスペースがあることとファイルが他のプロセスで使用されていないことを確認して下さい。

127 Not found: <vector>.

行あるいは列<vector>を使用しようとしたが、その問題上で見つけることができませんでした。スペルを確認して再度実行して下さい。

130 Bound type illegal <type>.

不正な上下界値タイプ<type>が基底ファイル.BSSで使われています。不正な上下界値タイプと列の残りが下限に存在し、新たな基底が内部的に作られます。そして、行は基底/非基底状態を維持します。XPRsreadbasis(READBASIS)にtフラグを付けて使って、コンパクト形式基底を読むことをチェックしてください。

131 No column: <col>.

基底ファイル.BSS内で使用される列<col>がその問題に存在しません。列<col>が取り除かれるところに新たな基底が内部的に作られ、残りの列と行は基底/非基底状態を維持します。

132 No row: <row>.

基底ファイル.BSSに使用される行<row>がその問題に存在しません。行<row>が取り除かれるところに新たな基底が内部的に作られ、残りの列と行は基底/非基底状態を維持します。

140 Basis lost — recovering.

Optimizerはメモリ内で問題と基底の間でいくつかの疑問を見つけました。修正された基底が内部

に作成されました。

142 Type illegal <type>.

間違った優先度タイプ <type> がディレクティブファイル .DIR で見つかったので無視されます。

有効な優先度タイプについては付録 A 「ログとファイル形式」を参照して下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 325

Error Codes 143 – 180 **9**

143 No entity <ent>.

ディレクティブファイル .DIR で使用される要素 <ent> は、その問題で見つかりません。それに相当する優先度は無視されます。スペルをチェックし、列 <ent> が、BOUNDS セクションで要素として実際に宣言されているかを確認するか、要素を設定してください。

151 Illegal MARKER.

整数列の集合のスタートや .MPS ファイル内の特別順序集合に属する列の集合をマークするラインが正しくありません。

152 Unexpected EOF.

Optimizer は想定外の EOF マークを見つけました。入力ファイルが正しいことを確認して再入力して下さい。

153 Illegal card at line <line>.

.MPS ファイルの <line> 行を解釈できませんでした。有効な MPS 形式については、リファレンスマニュアルを参照して下さい。

155 Too many files open for reading: <file>.

開いているファイルが多すぎるため、Optimizer はファイル <file> から読み込めませんでした。いくつかのファイルを閉じて、もう一度実行して下さい。

170 Corrupt global file.

グローバルファイル .GLB にアクセスできません。ディスクに十分なスペースがあることと、ファイルが他のプロセスで使用されていないことを確認して下さい。

171 Invalid row type for row <row>.

新しいタイプが無効であるため、XPRSalter (ALTER) は <row> の行タイプを変更できません。

訂正してもう一度実行して下さい。

172 Scaling too bad to continue.

スケールリングは反復要素に不都合です。行列をスケールリングしてみてください。

178 Not enough spare rows to remove all violations.

十分なスペースがないため、Optimizer はこれ以上のカットを行列に追加できませんでした。

EXTRAROWS を増やしてから入力して下さい。

180 No change to this SSV allowed.

Optimizer はこの制御を変更できません。学生バージョンの場合は、制御値を変更するためのアップグレードについてお近くの営業所にお問い合わせ下さい。そうでなければ、Optimizer は正常に初期化されたことと、セキュリティーの問題で学生用モードになっていないかチェックして下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 326

Error Codes 181 – 259 **9**

181 Cannot alter bound on BV, SC,UI, PI, or set member.

変数タイプがバイナリ、半連続、整数、部分整数、半連続整数である場合、あるいは、行列要素である場合、XPRSalter (ALTER) は変数の上下限值を変更できません。

186 Inconsistent number of variables in problem.

コンパクト形式の基底を作成した基底と違う変数の数で問題に読もうとしています。

199 SB recursion entries cannot be global entities <num>.

反復変数番号 <num> は離散要素です。離散要素を XPRSrecurse (RECURSE) セクションから取り

除いて再度入力して下さい。

245 Not enough memory to presolve matrix.

Optimizerは行列を事前分析するために、更にメモリが必要です。仮想ページングスペースか物理メモリを増やしてください。これができない場合、最適化の前にPRESOLVEに0を設定すれば、事前分析は実行されません。

246 Wrong release of binary files.

リリース<rel1>を検出しました。リリース<rel2>が必要です。リリース <rel1>で生成されたバイナリファイル*.BIFを読みこみできません。リリース<rel2>あるいは、1つ前の<rel2>-1 を使って、バイナリファイルを作成して、もう一度実行して下さい。

247 Directive on non-global entity not allowed: <col>.

ディレクティブファイル.DIRで使用された列<col>は離散要素でなく、それに相当する優先度は無視されます。変数は「離散要素」で、連続変数ではないか集合要素です。有効な要素と集合のタイプについては付録Aの「ログとファイル形式」をご参照下さい。

255 Not enough space to presolve matrix. Increase <par> before XPRSreadprob (READPROB).

行列の事前分析のための十分なスペースがありません。XPRSreadprob (READPROB)の前にパラメータ<par>を増やすか、PRESOLVEに0を設定して事前分析を行わないようにして下さい。

256 Simplex Optimizer only: buy barrier Optimizer from Dash Associates.

Optimizerはシンプレックスアルゴリズムのみ使用できます。このコマンドを使用したい場合は、ライセンスのアップグレードについてお近くの営業所にお問い合わせ下さい。

259 Cannot use XPRSrecurse (RECURSE) if presolve applied.

XPRSrecurse (RECURSE)は事前分析された問題では実行することができません。最適化を実行する前かフラグを1にしたXPRSminim (MINIM)/XPRSmaxim (MAXIM)を用いて最適化する前にPRESOLVEに0を設定して下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 327

Error Codes 261 – 278 **9**

261 <ent> already declared as a global entity — old declaration ignored.

要素<ent>は既に離散要素として宣言されています。新しい宣言の方が有効で古い宣言は無視されます。

262 Unable to remove shift infeasibilities of &.

問題を解くのに適用された制約の右辺の摂動は取り除かれません。入力データにおけるラウンドオフエラーか問題のスケールが悪いためと思われます。

263 The problem has been presolved.

メモリにある問題は事前分析されています。事前分析された問題で操作しようとした。フラグを1としたXPRSminim (MINIM)/XPRSmaxim (MAXIM)を呼び出すか、PRESOLVEに0を設定して事前分析をしないようにして再度最適化を行ってください。最適化された問題で実行する必要がない場合は、もう一度問題をロードするだけで構いません。

264 Not enough spare matrix elements to remove all violations.

Optimizerはスペース不足のため行列にこれ以上のカットを追加できません。EXTRAELEMENTSを増やした後、入力して動作を改善して下さい。

266 Cannot read basis for presolved problem. Re-input matrix.

メモリにある問題が事前分析済みのため、基底を読みません。XPRSreadprob (READPROB)を使用して問題を再度ロードし、再度基底を読みこんでみてください。

268 Cannot perform operation on presolved matrix. Re-input matrix.

メモリ上の問題が事前分析済みです。問題を再度ロードしてもう一度操作を行ってください。

277 This version is not authorized to run as a DLL.

Xpress-MPのサプライヤーに連絡してください。Optimizer部分ルーチンライブラリはDLLとして

実行する権限がありません。DLLとしてOptimizerを実行したい場合は、権限に関するアップグレードについてお近くの営業所にお問い合わせ下さい。

278 No purchase authorization found.

xpress.pwdファイルの存在を確認して下さい。OptimizerはOptimizerのディレクトリ内のxpress.pwdを使用しようとし、見つからない場合は、環境変数XPRESSに指定されたディレクトリで探します。Optimizerを実行するのにドングルが必要な場合は、それをチェックして下さい。それでも問題が解決されない場合は、お近くの営業所にお問い合わせ下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 328

Error Codes 279 – 309 **9**

279 Xpress-MP has not been initialized.

Optimizerは正常に初期化できませんでした。まず最初に初期化を行ってから他の操作を行って、再度実行してみてください。

285 Cut pool is full. Increase CPMAXELEMS before input.

Optimizerはカットを保存するためのスペースを使い切りました。CPMAXELEMSを増やして再度問題を読みこんで下さい。

286 Cut pool is full. Increase CPMAXCUTS before input.

Optimizerはカットを保存するためのスペースを使い切りました。CPMAXCUTSを増やして再度問題を読みこんで下さい。

287 Cannot read in directives after the problem has been presolved.

メモリにある問題が事前分析されているため、指示を読みこみません。最適化の前に、問題を再ロードしてディレクティブファイル.DIRを読みこんで下さい。-1 フラグを使って再度最適化を行うかPRESOLVEに0を設定するかして実行してみてください。

302 Option must be C/c or O/o.

目標のタイプの有効なオプションは、C, c, O, oのみです。他のものは無視されます。

305 Row <row> (number <num>) is an N row.

制限付きの行、つまりG、L、R、Eタイプだけが、目標計画のタイプで使われます。N行を目標として使う場合、目的関数となる目標計画を選択して下さい。

306 Option must be MAX/max or MIN/min.

最適化方向の有効なオプションは MAX, max, MIN, minのみです。それ以外は無視されます。

307 Option must be P/p or D/d.

目標の緩和タイプのオプションはP, p, D, d のみ有効です。それ以外は無視されます。

308 Row <row> (number <num>) is an unbounded goal.

目標計画は目標<row>が非有界であることを発見したので、この位置で止まります。<row>よりも優先度が低い目標は無視されます。

309 Row <row> (number <num>) is not an N row.

この目標計画タイプではNタイプの行だけが目標として選択できます。N以外のタイプの行を使用するときは、制約として目標計画を使用して下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 329

Error Codes 310 – 352 **9**

310 Option must be A/a or P/p.

目標計画のタイプの有効なオプションは A, a, P, p のみ有効です。それ以外は無視されます。

314 Invalid number.

入力が数字ではありません。スペルをチェックして再度実行して下さい。

316 Not enough space to add deviational variables.

入力の前にEXTRACOLSを増やしてください。Optimizerは予備偏差変数のための予備列を見つけれられません。EXTRACOLSを入力する前に最低でも目標制約数の2倍に増やし再度実行してください。

318 Maximum number of allowed goals is 100.

目標計画は100以上の目標をサポートしていないため中断します。

320 This version is not authorized to run under Windows NT.

Xpress-MPのサプライヤーに連絡してください。Optimizerは Windows NTでの実行が認められていません。このプラットフォームで実行したい場合は、ライセンスのアップグレードについてお近くの営業所にお問い合わせ下さい。

324 Not enough extra matrix elements to complete elimination phase.

入力の前にEXTRAPRESOLVE を増やして下さい。追加行列要素を作成した事前分析により消去フェーズが実施されました。この要素の数がEXTRAPRESOLVE パラメータで許容されている数より大きければ、消去フェーズは止まります。問題をロードする前にEXTRAPRESOLVE を増やして実行を改善して下さい。

326 Linear Optimizer only: buy QP Optimizer from Dash.

二次計画Optimizerを使用する権限がありません。このコマンドを使用したい場合は、QP Optimizerへのアップグレードについてお近くの営業所へお問い合わせ下さい。

349 Release <rel1> of binary files used with version <rel2>.

リリース<rel1>で作成されたバイナリファイルがリリース<rel2>で使用されています。

352 Command not authorized in this version.

使用権限のないコマンドを使おうとしました。このコマンドを使用したい場合は、ライセンスのアップグレードについてお近くの営業所へお問い合わせ下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 330

Error Codes 361 – 389**361 QMATRIX or QUADOBJ section must be after COLUMN section.**

行列ファイルでエラーがありました。QMATRIXあるいはQUADOBJセクションがCOLUMNセクションの後にあることを確認して再度実行して下さい。

362 Duplicate elements not allowed in QUADOBJ section.

列の係数が、QUADOBJセクションで2度以上現れました。このセクションにおいて、全ての列が1つの係数だけをもつことを確認して下さい。

363 Quadratic matrix must be symmetric in QMATRIX section.

対称行列だけが.MATファイルあるいは.MPSファイルのQMATRIXセクションに入力できます。修正して再度実行して下さい。

364 Problem has too many QP matrix elements. Please increase M_Q.

二次要素が非常に多いため、問題を読みこめません。M_Qを増やして、再度実行して下さい。

366 Problems with Quadratic terms can only be solved with the barrier.

バリア法以外のアルゴリズムを使って2次計画問題を解こうとしました。bフラグと共にXPRsminim (MINIM)/XPRsmaxim (MAXIM)を使ってバリアソルバにより解いてください。

368 QSECTION second element in line ignored: <line>.

<line> 行の第2要素は無視されます。

381 Bug in lifting of cover inequalities.

内部エラーです。直ちにお近くの営業所へお問い合わせ下さい。

386 This version is not authorized to run Goal Programming.

Xpress-MPのサプライヤーにご連絡ください。使用しているOptimizerには目標計画を実行する権限がありません。このコマンドを使用したい場合は、ライセンスのアップグレードについてお近くの営業所へお問い合わせ下さい。

387 Parallel code not initialized — continuing in serial mode.

パラレルモードが使用できません。スレーブの数が0より大きいことと、HPPVMが正しくインストールされていることをチェックして下さい。

388 Slave number <num> has failed.

スレーブ番号<num>は使えませんでした。有効なスレーブに作業が割り当て直されます。

389 Incorrect type of dongle, or security violation on slave <num>.

Xpress-MPはスレーブ<num>上で初期化できませんでした。残ったスレーブに割り当て直します。詳細は1章の初期化に関するセクションをご覧ください。それでも問題が解決しない場合はお近くの営業所へお問い合わせ下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 331

Error Codes 390 – 402 9**390 Slave number <num> has failed — insufficient memory.**

メモリが不足しているため、スレーブ<num>上のプロセスは中止されました。仮想ページングスペースか物理メモリを増やして再度実行して下さい。使えないスレーブの作業は残ったスレーブに割り当て直されます。

391 All slaves have failed — continuing in serial mode.

全てのスレーブが使えず、最適化はシリアルモードで続けられます。

392 This version is not authorized to be called from BCL.

Xpress-MPのサプライヤーにご連絡ください。部分ルーチンライブラリBCLからこのバージョンのOptimizerを呼ぶことはできません。BCLからOptimizerを実行したい場合は、ライセンスのアップグレードについてお近くの営業所へお問い合わせ下さい。

394 Fatal communications error.

マスターとスレーブプロセスの間でコミュニケーションエラーがありました。ネットワークをチェックし、再度実行して下さい。

395 This version is not authorized to be called from the Optimizer library.

Xpress-MPのサプライヤーにご連絡ください。OptimizerライブラリからこのバージョンのOptimizerを呼ぶことはできません。このライブラリを使ってOptimizerを実行したい場合は、ライセンスのアップグレードについてお近くの営業所へお問い合わせ下さい。

396 Insufficient memory on slave <num>.

スレーブ上のメモリが不足しているため、スレーブ<num>上でプロセスを開始できません。仮想ページングスペースか物理メモリを増やして再度実行して下さい。使えないスレーブの作業は残ったスレーブに割り当て直されます。

401 Invalid row type passed to <function>.

配列要素<num>に無効な行タイプ<type>があります。関数<function>の引数にエラーがありました。配列要素<num>に相当する行タイプが無効です。行タイプの詳細な情報やタイプについては6章「コンソールとライブラリ関数」の関数<function>をご参照下さい。

402 Invalid row number passed to <function>.

行番号<num>は無効です。関数<function>の引数にエラーがありました。配列の要素<num>に相当する行番号は無効です。行番号が0より小さい値でないことと、問題の行の総数を超えていないことを確認して下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 332

Error Codes 403 – 409 9**403 Invalid global entity passed to <function>.**

配列の要素<num>に無効な要素タイプ<type>があります。関数<function>の引数にエラーがありました。配列の要素<num>に相当する列タイプ<type>は離散要素には無効です。

404 Invalid set type passed to <function>.

配列の要素<num>に無効な集合タイプ<type>があります。関数<function>の引数にエラーがありました。要素<num>に相当する集合タイプ<type>は集合要素には無効です。

405 Invalid column number passed to <function>.

列番号<num>は無効です。関数<function>の引数にエラーがありました。配列の要素<num>に相当する列番号が無効です。列番号が0より小さくないことと問題の列の総数COLS-1を超えていないことを確認して下さい。XPRSgetobj, XPRSchgobjを呼ぶ場合、-1の列番号は有効で目的関

数の定数を参照します。

406 Invalid row range passed to <function>.

Limit <lim>は範囲外です。関数<function>の引数にエラーがありました。行番号は0から問題の行の総和の間です。Limit <lim>はこの範囲外にあり、有効ではありません。

407 Invalid column range passed to <function>.

Limit <lim>は範囲外です。関数<function>の引数にエラーがありました。列番号は0から問題の列の総和の間です。Limit <lim>はこの範囲外にあり、有効ではありません。

408 Too long a row or column name passed to <function>.

名前の長さ<num>が長すぎます。関数<function>の引数にエラーがありました。配列の要素<num>に相当する列名あるいは行名が長すぎます。

409 Invalid directive passed to <function>.

配列の要素<num>に無効な指示<type>があります。関数<function>の引数にエラーがありました。配列の要素<num>に相当する指示タイプ<type>は無効です。有効な指示タイプのリストについてはリファレンスマニュアルをご参照下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 333

Error Codes 410 – 416 9

410 Invalid row basis type passed to <function>.

配列の要素<num>に無効な行基底タイプ<type>があります。関数<function>の引数にエラーがありました。配列の要素<num>に相当する行基底タイプは無効です。

411 Invalid column basis type passed to <function>.

配列要素<num>に無効な列基底タイプ<type>があります。関数<function>の引数にエラーがありました。配列の要素<num>に相当する列基底タイプは無効です。

412 Invalid parameter number passed to <function>.

パラメータ番号<num>は範囲外です。LPあるいはMIPパラメータと制御パラメータは、第1引数としてパラメータか制御名を引き渡すことによって、あるいは関連番号を引き渡すことによって関数で使用できます。この場合、番号<num>は既存のパラメータや制御パラメータに対応しないため、関数<function>には無効な引数です。第1引数として番号を引き渡す場合は、設定したい値または得たいパラメータか制御の名前でそれを代用して下さい。パラメータや制御名を既に渡している場合は、6章「コンソールとライブラリ関数」をチェックし、関数<function>が有効であることを確認して下さい。

413 Not enough spare rows in <function>.

入力前にEXTRAROWSを増やしてください。関数<function>を正常に終了するのに十分な予備行がありません。XPRSreadprob (READPROB)の前にEXTRAROWSを増やして再度実行して下さい。

414 Not enough spare columns in <function>.

入力前にEXTRACOLSを増やしてください。関数<function>を正常に終了するのに十分な予備列がありません。XPRSreadprob (READPROB)の前にEXTRACOLSを増やして再度実行して下さい。

415 Not enough spare matrix elements in <function>.

入力前にEXTRAELMSを増やしてください。関数<function>を正常に終了するのに十分な予備行列要素がありません。XPRSreadprob (READPROB)の前にEXTRAELMSを増やして再度実行して下さい。

416 Invalid bound type passed to <function>.

配列の要素<elem>に無効な上下界値タイプ<type>があります。関数<function>の引数にエラーがありました。配列の要素番号<num>の上下界値タイプ<type>は無効です。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 334

Error Codes 418 – 428 9

418 Invalid cut number passed to <function>.

配列の要素<num1>に無効なカット番号<num2>があります。配列の要素番号<num1>はカットブ

ールにないカットを含みます。<num2>が有効なカット番号であるかチェックして下さい。

419 Not enough space to store cuts in <function>.

入力前にCPMAXELEMを増やしてください。関数<function>を正常に終了するのに十分なスペースがありません。XPRSreadprob (READPROB)の前にCPMAXELEMを増やして、再度実行して下さい。

422 Solution is not available.

有効な解がありません。メモリにある問題が変更されたか、最適化が実行されなかったために起こります。最適化を行って再度実行して下さい。

423 Duplicate rows/columns passed to <function>.

配列要素<elem>に二重行 / 列番号<num>があります。関数<function>の引数でエラーがありました。引数配列の要素番号<elem> は連続番号<num> が繰り返される行や列です。

424 Not enough space to store cuts in <function>.

入力前にCPMAXCUTSを増やしてください。関数<function>を正常に終了するのに十分なスペースがありません。入力前にCPMAXCUTSを増やして再度実行して下さい。

425 Column already basic.

既に基底であるために列をピボットして基底に入れることができません。ピボットして基底に入れる前に、変数が非基底であることを確認してください。

426 Column not eligible to leave basis.

既に非基底であるために、その列は基底に残すために選択されません。基底に残す前にその変数が基底であることを確認してください。

427 Invalid column type passed to <function>.

配列要素<num>に無効な列タイプ<type>があります。関数<function>の引数にエラーがありました。配列要素<num>に相当する列タイプ<type>が無効です。

428 Increase EXTRAMIPENTS before input.

関数<function>を正常に終了するのに十分な離散要素がありません。入力の前にEXTRAMIPENTSを増やし、再度実行して下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 335

Error Codes 430 – 506 **9**

430 Column types cannot be changed during the global search.

Optimizerは、整数解探索を実行している間、列タイプを変更することはできません。整数解探索を開始する前、あるいは整数解探索が終了した後に、この関数を呼んでください。離散要素の問題のLP解を見つけた後に整数解探索を自動的に開始することを望まない場合は、XPRSmnim(MINIM)あるいはXPRSmxim (MAXIM)に1フラグを伴って呼ぶことでできます。

434 Invalid name passed to XPRSgetindex.

行列の行あるいは列でない名前をXPRSgetindexに渡しました。

436 Cannot trace infeasibilities when integer presolve is turned on.

XPRSmnim (MINIM) / XPRSmxim (MAXIM)に1フラグを付けて試してください。整数事前分析は問題制約の関係により作られた上下限と同様に列タイプによって設定できます。実行不能トレース機能は問題制約による実行不能についてのみ説明可能です。

473 Row classification not available.

501 Error at <line> Empty file. Read aborted.

ファイルが空のため、Optimizerは問題を読みこめません。

502 Warning: 'min' or 'max' not found at <line.col>. No objective assumed.

目的関数を指定するものがLPファイルの<col>列や<line>行目で見つかりませんでした。目的関数を指定したい場合には、'max', 'maximize', 'maximum', 'min', 'minimize' や 'minimum'が現れることを確認してください。

503 Objective not correctly formed at <line.col>. Aborting.

LPファイルの<line> 行目に指定された目的関数が正しくなかったため、Optimizerは途中終了しました。

504 No keyboard or empty problem at <line.col>.

LPファイルの<line>行目の<col>列目にエラーがあります。'Subject to', 'subject to:', 'subject to', 'such that' 's.t.', または'st' が見つかりません。修正してもう一度実行して下さい。

505 A keyword was expected at <line.col>.

キーワードはLPファイルの<line>行目の<col>列目にあったようです。修正して再度実行して下さい。

506 A variable name was expected at <line.col>

右辺は定数でないため、LPファイルの<line>行目は無視されます。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 336

Error Codes 508 – 518 9

508 The constant at <line.col> has no term.

LPファイルの<line>行目の<col>列目に定義された制約条件には項がないため無視されます。

509 The type of the constraint at <line.col> has not been specified.

LPファイルの<line>行目の<col>列目に定義された制約条件は定数でないため無視されます。

510 Upper bound at <line.col> is not a numeric constant.

LPファイルの<line>行目の<col>列目に宣言された上限は定数でないため無視されます。

511 Bound at <line.col> is not a numeric constant.

LPファイルの<line>行目の<col>列目に宣言された上下界値は定数でないため無視されます。

512 Unknown word starting with an 'f' at <line.col>. Treated as 'free'.

'f'ではじまる言葉でXpress-MPに分からない言葉がLPファイルの<line>行目の<col>列目に見つかりました。その単語はXpress-MPに'free'として読みこまれます。

513 Wrong bound statement at <line.col>.

<line>行目の<col>列目の上下界値状態は無効で無視されます。

514 Lower bound at <line.col> is not a numeric constant. Treated as -inf.

LPファイルの<line>行目の<col>列目に宣言された下限は定数ではありません。可能な限り下限で設定し直されてXpress-MPに読まれます。

515 Sign '<' expected at <line.col>.

予期されなかった記号'<'以外の文字がLPファイルの<line>行目の<col>列目で見つかりました。この行は無視されます。

516 Problem has not been loaded.

Xpress-MPに問題をロードすることができませんでした。このメッセージの他に更に情報を示すエラーメッセージがないかチェックして下さい。

517 Row names have not been loaded.

Xpress-MPに行名をロードすることができませんでした。このメッセージの他に更に情報を示すエラーメッセージがないかチェックして下さい。

518 Column names have not been loaded.

Xpress-MPに列名をロードすることができませんでした。このメッセージの他に更に情報を示すエラーメッセージがないかチェックして下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 337

Error Codes 519 – 528 9

519 Not enough memory at <line.col>.

割り当てられたメモリ全てが既に使用されているため、LPファイルの<line>行目の<col>列目の情報を読めません。仮想ページングスペースあるいは物理メモリを増やして、再度実行して下さい。

520 Unexpected EOF at <line.col>.

LPファイルの<line>行目で、予期していないEOF文字が見つけれ、Optimizerへの問題のロードは途中で終了しました。修正して再度実行して下さい。

521 Number expected for exponent at <line.col>.

LPファイルの<line>行目の<col>列目の入力は適切な実数でないので無視されます。

522 Line <line> too long (length>255).

LPファイルの<line>行目は長すぎるため、Optimizerへの問題のロードは途中で終了しました。その行の長さが255以内であることをチェックして再度実行して下さい。

523 Xpress-MP cannot reach line <line.col>.

LPファイルの読みこみが内部問題により失敗しました。直ちにお近くの営業所へお問い合わせ下さい。

524 Constraints could not be read into Xpress-MP. Error found at <line.col>.

LP制約の読みこみが内部問題により失敗しました。直ちにお近くの営業所へお問い合わせ下さい。

525 Bounds could not be set into Xpress-MP. Error found at <line.col>.

LP上下境界値の設定が内部問題により失敗しました。直ちにお近くの営業所へお問い合わせ下さい。

526 LP problem could not be loaded into Xpress-MP. Error found at <line.col>.

LPファイルの読みこみが内部問題により失敗しました。直ちにお近くの営業所へお問い合わせ下さい。

527 Copying of rows unsuccessful.

LP行のコピーが内部問題により失敗しました。直ちにお近くの営業所へお問い合わせ下さい。

528 Copying of columns unsuccessful.

LP列のコピーが内部問題により失敗しました。直ちにお近くの営業所へお問い合わせ下さい。

Xpress-Optimizer Reference Manual Error Messages & Return Codes 338

Error Codes 529 – 530 9

529 Redefinition of constraint at <line.col>.

制約条件がLPファイルの<line>行目の<col>列目で再定義されています。繰り返し定義されたものは無視されます。

530 Name too long. Truncating it.

LPファイルに64文字以上の識別子があります。最大文字数で切り捨てられます。

Xpress-Optimizer Reference Manual Log and File Formats 339

付録 A — ログとファイル形式

A.1 ファイル形式

Optimizerは求解過程の中で様々なタイプのファイルをたくさん生成したり入力します。デフォルトでは、これらは全て3文字の拡張子がない問題名(*problem_name*)でファイル名が決定されます。Optimizerに関するファイルタイプには次のようなものがあります。

拡張子	解説	ファイルタイプ
.alt	行列変更ファイル。XPRSalter (ALTER)により入力されます。	ASCII
.asc	CSV形式の解ファイル。XPRSwritesol (WRITESOL) により出力されます。	ASCII
.bif	バイナリインタフェイス行列ファイル。XPRSreadprob (READPROB)により入力されます。	Binary
.bss	XPRSreadbasis (READBASIS)により入力され、XPRSwritebasis (WRITEBASIS)により出力される基底ファイル。	ASCII
.ctp	カットプールファイル(MIPのみ)。XPRSGlobalで使われます。	Binary
.dir	ディレクティブファイル(MIPのみ)。XPRSreaddir (READDIRS) により入力されます。	ASCII
.glb	グローバルファイル(MIPのみ)。XPRSGlobal (GLOBAL)で使われます。	Binary
.gol	目標計画入力ファイル。XPRSGoal (GOAL)により入力されます。	ASCII

.grp	目標計画出力ファイル。XPRsgoal(GOAL)で出力されます。	ASCII
.hdr	解ヘッダファイル。XPRswritesol(WRITESOL)とXPRswriterange(WRITERANGE)により入力されます	ASCII
.iis	IIS出力ファイル。XPRsiis(IIS)で出力されます。	ASCII
.lp	LP形式行列ファイル。XPRsreadprob(READPROB)により入力されます。	ASCII
.mat	MPS / XMPS形式ファイル。XPRsreadprob(READPROB)により入力されます。	ASCII

Xpress-Optimizer Reference Manual XMPS Matrix Files 340

次のセクションでは、これらのいくつかのファイル形式について説明します。CSVはカンマ区切り値のテキストファイル形式であることを表します。

A.2 XMPS 行列ファイル

Xpress-MP OptimizerはLPまたはMPS形式とその拡張であるXMPS形式の中の行列ファイルを受入れます。後者は工業標準を若干改良したものでここに詳細を示します。

XMPS形式は次のようなフィールドを定義します:

次のようなセクションが定義されます:

.prt	固定形式解ファイル。XPRswriteprtsol(WRITEPRTSOL)で出力されます。	ASCII
.rng	レンジファイル。XPRsrange(RANGE)で出力されます。	Binary
.rrt	固定形式レンジファイル。XPRswriteprtrange(WRITEPRTRANGE)で出力されます。	ASCII
.rsc	CSV形式レンジファイル。XPRswriterange(WRITERANGE)で出力されます。	ASCII
.sol	解ファイル。XPRsminim(MINIM)/XPRsmaxim(MAXIM)とXPRsglobal(GLOBAL)で出力されます。	Binary
.svf	Optimizer 状態ファイル。XPRssave(SAVE)で出力され、XPRsrestore(RESTORE)により入力されます。	Binary

フィールド 1 2 3 4 5 6

列 2-3 5-12 15-22 25-36 40-47 50-61

NAME	行列名
ROWS	行の定義を始める
COLUMNS	列の定義を始める
QUADOBJ	2次目的関数の定義を始める
SETS	SOSの定義を始める
RHS	右辺の定義を始める

拡張子 解説

ファイルタイプ

Xpress-Optimizer Reference Manual Log and File Formats 341

全てのセクション定義は列 1 から始まります。

NAMEセクション

ROWSセクション

に続いて行定義を書きます。

行タイプ (フィールド1) は次の通り:

RANGES	行範囲の定義を始める
BOUNDS	上下界値の定義を始める
RECURSE	反復定義の定義を始める
ENDATA	行列の終わりを示す

形式: 列 1-4 フィールド 3

NAME *model_name*

形式: 列 1-4

ROWS

フィールド 1 フィールド 2

type row_name

N 制約でない (目的関数に使用)

L 以下

G 以上

E 等しい

Xpress-Optimizer Reference Manual XMPS Matrix Files 342

COLUMNSセクション

に続いて行列の列を書きます。つまり、1つの列の全要素は別の列が始まるまでに終わる必要があります。そこでは、列 col と行 $row1$ (col , 行 $row2$ の $value2$) に $value1$ を入れることを示します。フィールド5/6の組はオプションです。

QUADOBJ / QMATRIXセクション(2次計画のみ)

2次目的関数はQUADOBJ、QMATRIXセクションを含むMPSファイルに指定されます。固定形式XMPSファイルでは、セクション形式は次の通りです:

または

2次の項の記述が続きます。各2次の項では以下の通りです:

$col1$ は2次の項での最初の変数で、 $col2$ が2番目の変数で、 $value1$ は Q 行列と関係のある係数です。

QMATRIXセクションでは、全ての非零 Q 要素は指定されます。QUADOBJセクションでは Q の上(または下)三角部分にある非零要素のみが指定されます。QMATRIXセクションでは、ユーザは Q 行列が対称であることを確認し、QUADOBJセクションでは Q の対称性が推定され、足りない部分については自動的に生成されます。

形式 列 1-7

COLUMNS

フィールド1 フィールド2 フィールド3 フィールド4 フィールド5 フィールド6

blank col row1 value1 row2 value2

形式:列1-7

QUADOBJ

形式列1-7

QMATRIX

フィールド1 フィールド2 フィールド3 フィールド4

blank col1 col2 value

Xpress-Optimizer Reference Manual Log and File Formats 343

QUADOBJとQMATRIXセクションはCOLUMNSセクションの後のどこかに置かれる必要があります、列セクションで前もって定義された列だけを含みます。問題行列内に要素のない列はCOLUMNSセクションで費用係数(大抵0)により指定されて定義されます。

SETSセクション(整数計画のみ)

このレコードは特殊順序集合(SOS)を指定するセクションを始めます。定義があるときには、COLUMNSセクションの後、RHSセクションの前に置かなくてはなりません。この後に、下に定義するように各集合のタイプ、名前、参照行、要素を指定するレコードが続きます。

これらは連続的である必要があります、つまり1つの集合の要素は別の集合が始まる前に終わらなくてはなりません。各集合の最初のレコードでは以下の通りです:

それ以後のレコードでは:

全てのレコードにおいて、 $vec2$ はオプションです。

集合の最初のレコードにある参照行名を空白にした場合、参照行はその次以降のレコードに含まれてはなりません。

形式: 列 1-4

SETS

フィールド1 フィールド2 フィールド3 フィールド4 フィールド5 フィールド6

type set vec1 blank vec2 blank

*type s1*はタイプ1の特殊順序集合です

*s2*はタイプ2の特殊順序集合です

set は集合名です

vec1 は参照行の名前です

*vec2*は集合の最初の列の名前です

type と *set* は空白です

vec1 と *vec2* は集合要素の名前です

*type*は空白です

Xpress-Optimizer Reference Manual XMPS Matrix Files 344

ここで参照入力が含まれると、それらはスケーリングされません。

RHSセクション

下で定義されるように右辺が続きます:

rhs と呼ばれ、行 *row1* (と行 *row2* の *value2* の値) の *value1* の値を持つ右辺を指定します。フィールド 5 / 6 の組はオプションです。

RANGESセクション

次に定義するように右辺範囲が続きます:

rng と呼ばれ、行 *row1* (と行 *row2* の *value2* の値) の *value1* の値を持つ右辺レンジ列を指定します。フィールド 5 / 6 の組はオプションです。

b が RHS セクションで与えられる値であるとする、RANGES セクションで与えられる値が *r* のとき、いかなる行についても、以下のようなアクティビティ限度が適用されます:

vec1 は集合要素の名前です

vec2 は参照行の要素です

形式: 列1-3

RHS

フィールド1 フィールド2 フィールド3 フィールド4 フィールド5 フィールド6

blank rhs row1 value1 row2 value2

形式: 列1-6

RANGES

フィールド1 フィールド2 フィールド3 フィールド4 フィールド5 フィールド6

blank rng row1 value1 row2 value2

行のタイプ *r* の符号 上限 下限

G +

L +

E +

b r + b

b br-

b r + b

Xpress-Optimizer Reference Manual Log and File Formats 345

BOUNDSセクション

変数に有効な上下界値が続きます:

線形計画の上下界値タイプは:

整数計画には5つの追加上下界値タイプがあります:

指定された値はとれる値の上限で、値はUP, FR, UI, SC, SI のタイプがあります。;
 下限についてはLOとLIのタイプがあります。;

固定値のタイプはFX;

無視されるタイプはBV, MI, PL

PIはスイッチ変数です:

下は変数を整数とし、上は変数を連続とします。非整数値がUI かLI タイプで与えられると、値の整数部分だけが使われます。

E -

形式: 列 1-6

BOUNDS

フィールド1 フィールド2 フィールド3 フィールド4

タイプ ブランク 列 値

UP	上限
LO	下限
FX	変数の固定値
FR	自由変数
MI	非正 (マイナス) 変数
PL	非負 (プラス) 変数 (デフォルト)
UI	上限のある一般の整数変数
LI	下限のある一般の整数変数
BV	バイナリ変数
SC	準連続変数
SI	準連続性数変数
PI	部分整数変数

b br+

Xpress-Optimizer Reference Manual XMPS Matrix Files 346

整数変数 : 0から上限までの整数値をとります。1を上限とする整数変数はバイナリ変数として扱えます。

バイナリ変数 : 0と1の値をとります。0/1変数とも呼ばれます。

部分整数変数 :ある値以下では整数、それ以上では連続変数として扱われます。

準連続変数 : 0かある下限値から上限値の範囲の値をとります。デフォルトでは、下限が1.0です。他の正の値は明示的な下限値として指定できます。例えば、

BOUNDS

x .SC. 12.3

x >= 0.8

は、x は零か0.8と12.3の間の値をとれることを表します。

準連続整数変数 :零か下限値から有限の上限値の範囲の整数変数をとります。

RECURSEセクション(反復のみ)

逐次線形計画として知られている反復法は、線型計画法によりある種の非線型計画問題を解く方法です。LP問題の係数の一部は、LP変数の最適値の関数として定義されます。LP問題が解けると、係数を再評価して再度LPを解きます。適当な条件のもとで、この過程は局所最適解(必ずしも大域的最適解ではありません)に収束します。OptimizerはXMPS行列ファイルの特定のセクションにある前もって指定された関数の独立性を前提とした上で反復をできるようにします。

反復された行列要素は、MPS形式の拡張である行列ファイルのRECURSEセクションにおいて

Optimizerに指定されます。反復は、XPRSrecurse (RECURSE)を用いて問題を最小化か最大化した後で実施されます。RECURSEセクションはXMPS .mat ファイル中のENDATAの前の最後のセクションに、次のような形式で反復のための要素を定義するラインと共になくてはなりません。先

の反復でのベクトル j の最適値を $v(j)$ とします。そして、以下のように各反復によって行列要素は変更されます。 $a(i,j)$ は行列 j の行列要素を示します。 i, j, k, l, m はベクトルの名前で、 j, k, l は列を表します。

DV $j i c(i,j) k l a(i,j) c(i,j)*v(k) / v(l)$

Xpress-Optimizer Reference Manual Log and File Formats 347

注意

- CV, NE, NL, DV, MU, CL, BX, SBは列2-3に、 j は列5-12に、 i (あれば)は列15-22に、 $c(i,j)$ または tol (あれば)は列25-36に、 k, m (あれば)は列40-47に、 l (あれば)は列50-57に置かなくてはなりません。 $c(i,j)$ が存在しない場合には、値を1.0と仮定します。
- 行列と上下界値要素だけが反復できます。右辺の反復はできません。
- SBは「対称上下界値」です。ベクトル x の上下界値は $-U \times U$ の形であるとし、初期上限値 U にはBOUNDSセクション(下限があっても無視します)の x の上限を採用します。両上下界値は同時に拡大, 縮小されますので、常に対称のままです。
- CV要素がなければ、以下の場合に反復は停止します。
新しい目的関数値、古い目的関数、値のところで反復は最大RECMAXPASSES回実施されます。
- NEまたはNL要素がある場合、KEEPPNROWSは問題が読み込まれる前に1に設定されなければなりません。
- 事前分析機能が反復と共に使われる場合、行列は各反復の開始時に事前分析され、終了時に事後分析されます。事前処理が再度最適化する時間に比べて非常に長いのであれば、事前処理をオフにした方がよいと思われます。ニュートンバリア法が使われるときは、XPRSrecurse(RECURSE)を使う前にクロスオーバーを実施してください。

MU $j i c(i,j) k l a(i,j) c(i,j)*v(k)*v(l)$

CL $j i c(i,j) k a(i,j) c(i,j)*v(k)$

BX $j i m a(i,j) a(i,j)*EXPAND$ m が基底でないとき

$a(i,j) a(i,j)*SHRINK$ m が基底のとき

SBj $m UB(j) UB(j)*EXPAND$ m が基底でないとき

$UB(j) UB(j)*SHRINK$ m が基底のとき

CV $j tol j$ 現在の変数 j の値と最後の繰返しでの変数 j の値だったところで収束します

NE j は反復が始まる前にNタイプ行を等式にします

NL j は反復が始まる前にNタイプ行を「より少ない、もしくは等しい」の式にします

$v_{now} v_{old} - tol v_{old} 1.0e 6 - + () \cdot <$

$v_{now} v_{old}$

..

$obj_{new} obj_{old} - RECSTOP obj_{old} 1.0e 6 - + () \cdot <$

$obj_{new} obj_{old}$

Xpress-Optimizer Reference Manual ASCII Solution Files 348

ENDATAセクション

ファイルの最後のレコードです。

A.3 ASCII解ファイル

解の情報は用途に応じて様々なファイル形式でOptimizerから利用可能です。XPRSwritesol (WRITESOL) コマンドは2つのファイル $problem_name.hdr$ と $problem_name.asc$ を作成し、その出力はカンマ区切り形式フィールドで主に他のプログラムに入力するためのものです。対照的に、XPRSwriteprtsol (WRITEPRTSOL) コマンドは固定形式出力ファイルを作成し、プリンタに直接

送るためのものです。ファイル名は `problem_name.prt` です。これら 3 つのファイルについて、以下に説明します。

解ヘッダファイル(.hdr)

このファイルには .asc ファイル (問題の各行各列のデータが入っています) を読むのに使われる 1 行の文字からなるヘッダ情報が入っているだけです。この 1 行には次のようなカンマで区切られた 14 のフィールドがあります。

形式: 列 1-6

ENDATA

フィールド	タイプ	字数	説明
1	文字列	10	行列名
2	整数型	4	問題中の行数
3	整数型	6	問題中の構造列の数
4	整数型	4	目的関数行の番号
5	文字列	3	問題の状態 (下記注意参照)
6	整数型	4	最適化の方向 (0:なし, 1:最小化 2:最大化)
7	整数型	6	繰返し回数
8	整数型	4	実行不能性の最終数
9	実数型	12	最終の目的関数値

Xpress-Optimizer Reference Manual Log and File Formats 349

注意

- 文字フィールドは文字列を 2 重引用符で囲んだものです
- 整数フィールドは右詰めの 10 進数です
- 実数フィールドは小数部分 6 桁の 10 進数を右詰めにしたものです
- 問題の状態 (フィールド 5) は以下のような 1 文字です

Csv 形式解ファイル(.asc)

解についての大部分の情報がこのファイルには含まれます。各行、列に 1 行の文字列が、行を最初に入力順に並んでいます。行には次に示すような 10 のフィールドがあり、カンマで区切られています。

10	実数型	12	実行不能性の最終和
11	文字列	10	目的行名
12	文字列	10	右辺行の名前
13	整数型	1	フラグ: 整数解が見つかったとき (1), その他 0
14	整数型	4	行列のバージョン番号

O 最適解
 N 実行不能
 U 無限解 (非有界)
 Z 未完了

フィールド	タイプ	字数	説明
1	整数型	6	変数の入力連続番号
2	文字列	10	変数 (行または列ベクトル) 名
3	文字列	3	変数タイプ (C=列; N, L, G, E は行)
4	文字列	4	変数状態 (LL, BS, UL, EQ または**);
5	実数型	12	アクティビティの値
6	実数型	12	スラックアクティビティ (行) または入力費用 (列)
7	実数型	12	下限 (なければ -10000000000)
フィールド	タイプ	字数	説明

Xpress-Optimizer Reference Manual ASCII Solution Files 350

注意

- フィールドタイプは .hdr ファイルに同じです
- 変数タイプ (フィールド 3) は次により定義されます
- 変数タイプ (フィールド 4) は次により定義されます

固定形式解ファイル(.prt)

このファイルは XPRSwriteprtsol (WRITEPRTSOL) コマンドにより出力され、コンソールにおいて PRINTSOL により表示されるのと同様の形式のもので、この形式での表示例を以下に示します。この例は Xpress-MP Essentials ガイドで使われます。

最初のセクションでは求解過程と見つかった最適解についての概要が示されます。そこでは使われた行列 (問題) 名 (simple) と目的関数名、右辺について示されます。そして、行と列の番号が続き、そこでは最大化が実行されました。2 回の繰返し (シンプレックスピボット) で解かれた結果、最適解の値は 171.428571 です。

8	実数型	12	上限 (なければ 1000000000)
9	実数型	12	双対アクティビティ (行) または被約費用 (列)
10	実数型	12	右辺値 (行) または空白 (列)

C	構造列
N	Nタイプ行
L	Lタイプ行
G	Gタイプ行
E	Eタイプ行
LL	下限で非基底
**	基底で実行不能
BS	基底で実行可能
UL	上限で非基底
EQ	等式行
SB	変数が超基底
??	不明

フィールド タイプ 字数 説明

Xpress-Optimizer Reference Manual Log and File Formats 351

Problem Statistics

Matrix simple

Objective *OBJ*

RHS *RHS*

Problem has 3 rows and 2 structural columns

Solution Statistics

Maximization performed

Optimal solution found after 3 iterations

Objective function value is 171.428571

次に、*Rows Section* は問題の行の結果や制約を示します。

Xpress-Optimizer Reference Manual ASCII Solution Files 352

Rows Section

Number	Row	At Value	Slack Value	Dual Value	RHS
N 1	*OBJ*	BS	171.428571	-171.428571	.000000 .000000
L 2	second	UL	200.000000	.000000	.571429 200.000000
L 3	first	UL	400.000000	.000000	.142857 400.000000

最初の列は制約のタイプを表します。

L 等しいかそれ以下の制約

E 等式制約

G 等しいかそれ以上の制約

N 非拘束制約—つまり目的関数

次の列は通し番号で、制約条件の名前が続きます。At 列は制約の状態を示します。

ULはその行は上限にあることを意味します。この場合、型の行の値がそれを制約している右辺の値にぶつかっています。BSは制約式が効いていないため、問題から取り除いても最適値は変わりません。もし、型の制約式があると、LLが示されることがあり、これは制約が下限にあることを意味します。他の値には以下があります。

RHS列は原制約式の右辺で、Slack Valueは制約式が右辺値からの離れ量を示します。制約が効いている（ULまたはLL）ときは、スラックは0になります。

Dual Valueは制約の厳しさを表します。型の制約が上限にあれば、条件が少し緩和されると利益の増加を期待できます。双対変数値はその感触を正確に表すことのできる数値です。一般に、行の右辺の値を1増やすと、利益は行の双対値分だけよくなります。厳密に言うと、右辺をごくわずかに増やすと、利益は（限界）双対値だけ増えます。双対変数値は時々*shadow prices*として知られています。

最後に、*Columns Section* は列や変数の解を与えます。

Columns Section

Number Column At Value Input Cost Reduced Cost

C 4 a BS 114.285714 1.000000 .000000

C 5 b BS 28.571429 2.000000 .000000

** 基底で実行不能

EQ 等式行

?? 不明

.

.

.

.

TM

TM .

Xpress-Optimizer Reference Manual Log and File Formats 353

最初の列には列の情報であることを示すcがあります（上記行セクションと比較）。番号は通し番号です。Column と見出しがあるところは、決定変数の名前です。At は列の状態で、BSは解の状態が上限や下限から離れていることを表し、LLは下限、ULは上限にあることを示します。他に、Value 列は変数の最適値を示します。例えば、変数 a の最適値は114.285714で b は28.571429です。Input Cost列は目的関数での変数の係数を示します。

解の最終列には変数のReduced Cost（被約費用）が示され、これは変数が上下界値から離れている—ここでは0です。変数が0ということは、製品を作ることが引き合うには単位あたりの利益が十分でないことを意味します。被約費用はこの製品を作るとすればどれだけ単位あたりの利益が増加すればよいかを示しています。あるいは、*reduced cost*の名前が示すように、利益を減らすことなく製造するにはどれだけ製造費用が下がればよいかを示しています。

A.4 ASCII レンジファイル

レンジ（感度分析）情報をXPRsrange(RANGE)によって作成して画面表示させたり、直接ファイル出力させることができます。このためには2つの機能、即ちXPRswriteprtrange(WRITEPRTRANGE)とXPRswriterange(WRITERANGE)があります。このうち、XPRswriterange(WRITERANGE)は、2つのファイル*problem_name.hdr* と *problem_name.rsc* を作ります。これらは固定フィールドで、他のプログラムへの入力向けに作られたものです。対照的に、XPRswriteprtrange (WRITEPRTRANGE)コマンドはプリンタへ直接送るために作られた形式で

情報を出力します (*problem_name.rpt*)。両機能で得られる情報は基本的に同じで、単に出力する目的が違います。これらのファイルの様式は以下の通りです。

** 基底で実行不能
EQ 等式行
SB 変数が超基底
?? 不明

Xpress-Optimizer Reference Manual ASCII Range Files 354

解ヘッダファイル (.hdr)

このファイルには文字で 1 行のヘッダ情報が入っているだけで、*.rsc* ファイルを読むのに使われず。様式は XPRswritesol (WRITESOL) で作られるものと同じで、上記の解ヘッダファイル (*.hdr*) に説明されています。

Csv 形式解ファイル (.rsc)

レンジ情報の大部分がこのファイルには含まれます。問題の各行、列に 1 行の文字列が、行を最初に入力順に並んでいます。行には次に示すような 16 のフィールドがあり、カンマで区切られています。

注意

• フィールドタイプは *.hdr* ファイルと同じです

フィールド	タイプ	字数	説明
1	整数型	6	変数の番号
2	文字列	10	変数 (行または列ベクトル) 名
3	文字列	3	変数タイプ (C=列; N, L, G, E は行)
4	文字列	4	変数状態 (LL, BS, UL, EQ または**);
5	実数型	12	アクティビティの値
6	実数型	12	スラックアクティビティ (行) または入力コスト (列)
7	実数型	12	下側アクティビティ
8	実数型	12	下側単位コスト
9	実数型	12	下側利益
10	文字列	12	限界処理
11	文字列	4	限界(LL, UL)における限界処理状態
12	実数型	12	上側アクティビティ
13	実数型	12	上側単位コスト
14	実数型	12	上側利益
15	文字列	10	限界処理
16	文字列	4	限界(LL, UL)における限界処理状態

固定形式レンジファイル (.rpt)

このファイルは XPRswriteprtrange (WRITEPRTRANGE) コマンドの出力で、PRINTRANGE によりコンソールに示されるものと同じ形式です。この形式については以下の例により説明します。出力は 3 つのセクションに、概要データ、行データ、列データが示されます。このうち概要データについては、XPRswriteprtsol (WRITEPRTSOL) コマンド (上記参照) で得られるのと同じ情報で、以下のようなものです。

```
Problem Statistics
Matrix PLAN
Objective C0_____
RHS R0_____
C 構造列
N Nタイプ列
L Lタイプ列
```

G Gタイプ列

E Eタイプ列

LL 下限で非基底

** 基底で実行不能

BS 基底で実行可能

UL 上限で非基底

EQ 等式行

?? 不明

LL 下限で非基底

UL 上限で非基底

Xpress-Optimizer Reference Manual ASCII Range Files 356

Problem has 7 rows and 5 structural columns

Solution Statistics

Minimization performed

Optimal solution found after 6 iterations

Objective function value is 15.000000

次のセクションは問題の行や制約条件のデータを示します。各制約ごとにデータは2行に示されます。この例では、ちょうど制約1行分のデータが示されます。

Rows Section

Vector Activity Lower actvty Unit cost DN Upper cost Limiting AT

Number Slack Upper actvty Unit cost UP Process

G C1 10.000000 9.000000 -1.000000 x4 LL

LL 2 .000000 12.000000 1.000000 C6 UL

2行のうちの1行目には、行のタイプ(N, G, L, E)が行名の前に置かれます。そして、アクティビティの値が続きます。次にLower actvty、即ちUnit cost DN列で与えられる減少単価で、下げられるアクティビティの下限を表します。この値を超えると減少単価が変化します。Limiting Processは、行のアクティビティが下限アクティビティを超えて減少する場合、状態が変わる列や行の名前です。AT列は限度に達した場合の限度過程の状態を示します。LLは下限で基底から出るか入るかを表し、ULは上限で基底から出るか入るかを表します。Lower actvtyを計算する場合、行列のRHSセクションに指定された行の下限は無視されます。

2行目は行の現在の状況と通し番号で始まります。そして、行のスラック値が示されます。次の4つのデータは、上のデータに似ています。Upper actvtyの計算においては、やはり、そのアクティビティの上限は無視されます。

列や変数は同様に2行に示されます。例は2つ分です。

Columns Section

Vector Activity Lower actvty Unit costDN Upper cost Limiting AT

Number Input cost Upper actvty Unit costUP Lower cost Process

C x4 1.000000 -2.000000 5.000000 6.000000 C5 LL

BS 8 1.000000 3.000000 1.000000 .000000 C1 LL

C x5 2.000000 -1.000000 2.000000 6.000000 X3 LL

Xpress-Optimizer Reference Manual Log and File Formats 357

UL 9 4.000000 3.000000 -2.000000 -very large X2 LL

ベクトルタイプは常に列を表すCです。Activityは最適値です。Lower/Upper actvtyは費用係数についてInput costからUpper/Lower cost (最小化問題として)までの間を増加/減少した場合に得られるアクティビティレベルです。この計算では、列の上下限は無視されます。Unit cost DN/UP はアクティビティをLower/Upperアクティビティにまで減少/増加させたときの目的関数の変化量です。Limiting Processes とATの意味は行の場合と同じです。2行目には列の状態と通し番号があります。列が非基底の場合には、Unit costsは常に被約費用の(絶対)値です。

A.5 ディレクティブファイル (.dir)

このファイルは分枝の優先順位、分枝方向、擬コストを指定する任意の並びのレコードで、XPRSreaddirs (READDIRS)コマンドを使ってOptimizer に読まれます。これはモデルファイルのDIRECTIVESセクションで作られるか、マニュアルで作成されます。デフォルトでは、*problem_name.dir*の形で名前が付きます。ディレクティブファイルのレコード形式は以下の通りです。

列 2-3 列 5-12 列 25-36

タイプ 要素 値

PR 優先順位を与えます (値は優先順位を0-1000の整数で与えます。1000を超えた値については受け付けられません。また実数のときは小数点以下を切り捨てて整数にします。優先順位の値は小さい方が優先的に分枝に採用されます)

UP 要素が押し上げられます (値を使用しません)

DN 要素が押し下げられます (値を使用しません)

PU 押し上げ擬コスト入力 (値がコストを与えます)

PD 押し下げ擬コスト入力 (値がコストを与えます)

MC モデルカット入力 (値が使用されません)

Xpress-Optimizer Reference Manual The Matrix Alteration (.alt) File 358

要素は離散要素名 (ベクトルか特殊順序集合) かマスクです。マスクは通常、指定された文字に適合する文字で構成されます。?は1文字単位、*は文字列単位を表します。*はマスクの最後にのみ置くことができます。

valueはタイプを表す値です。

例:

PR x1* 2

名前がx1で始まる離散要素 (整数変数など) の優先順位を2とします。*をマスクで使うと、x1に続くの全ての文字列が該当することになります。

A.6 行列変更ファイル (.alt)

変更ファイルは行列の修正指示を含むASCIIファイルであり、XPRSalter (ALTER) コマンドを使って読まれ、デフォルトでは*problem_name.alt*と言う名前が付けられます。各指示はファイルに別々の行に書かれてあり、最後は空白行です。指示は変更する目的関数を指定する*identifiers*と、その目的関数に適用する*actions*とからなります。一般に、1行で目的関数を指定し、例えばR2というのは、そのような名前が行2に割り当てられているとして、2番目の行を指定します。係数を変更するのであれば、その変数について指定します。例えばこうです。

RRRRRRRR

CCRider

2.087

これにより、行RRRRRRRRR のCCRider の係数が2.087に変更されます。actionについては以下のようなものを設定可能です。

上下限の変更

列の上下限を変更する場合、特定の行における上下限について各々**UP、**LOと指定します。

右辺係数の変更

行の右辺係数を変更する場合、列の値を右辺'に変えます。

Xpress-Optimizer Reference Manual Log and File Formats 359

制約タイプの変更

制約の向きが変更できます。最初に行名を与え、次にアクションである**NTx を与えます。x は次の中の1つです。

XPRSreadprob (READPROB)の前に制御KEEPNROWSに0が設定してあるとNタイプの行はメモリ内の行列に存在できません。

A.7 グローバルログ

分枝限定木探索の間、(XPRSglobal (GLOBAL)をご覧ください)について6つの情報項目の概要ログが、 n ノードごとに出力されます。この n については値をMIPLOGで設定できます。これらの項目は以下から成ります。

このログはまた、整数実行可能解が見つかったと出力されます。星印(*)がログの両側に出力され、解が見つかったことを示します。

MIPLOGが3に設定してあると、探索木の各ノードごとに8つの項目の詳細ログが出力されます。

N	新しい行タイプを制約にする
L	新しい行タイプを「より小さいか等しい」にする
G	新しい行タイプを「より大きい等しい」にする
E	新しい行タイプを等式にする
Node	ノードの通し番号
Sols	見つかった整数実行可能解の数
Best Solution	見つかった整数実行可能解の一番よい値
Best Bound	見つけれられる最良の整数実行可能解の限界値
Active	分枝限定木探索のアクティブノード数
Time	経過時間
Branch	ノードの通し番号
Parent	このノードの親のノード番号

Xpress-Optimizer Reference Manual The Global Log 360

全てのノードに上の情報全てがあるわけではありません。LP緩和がカットオフされたときは、BranchとParent(多分Solution)のみを表示します。LP緩和が実行不能の場合には、BranchとParentのみが表示されます。整数解が得られれば、ログ行の前にそのことが強調して示されます。

MIPLOGが2に設定されていると、整数実行可能解だけについての詳細ログが出力されます。MIPLOGが0か1に設定されていると、ログは表示されず、状態メッセージのみが探索の最後に示されます。LP繰返しログは出なくなりませんが、大きな計算上の困難が発生するとLP Optimizerからのメッセージが表示されます。

Solution	そのノードにおけるLP緩和の最適値
Estimate	このノードから得られる最適整数解の推定値
Infeas	整数実行不能な数
Active	探索木中のアクティブノード数
Entity	このノードから探索を続ける必要があるとき、分割する整数要素
Value / Bound	分割のため上で選ばれる要素の現在の値。UまたはLが後につきます。U(L)ならば、新しい上限(下限)を最初に適用します。つまり、この要素からの最初の分枝として押し下げる(上げる)方向とします。

Xpress-Optimizer Reference Manual Index 361

Index

A

ACTIVENODES 302

Advanced Mode (アドバンスモード) 1, 41

algorithms (解探索法) 2

default (デフォルト) 19

Newton barrier (ニュートン・バリア) も参照ください

simplex (シンプレックス法) も参照ください

ALTER 53, 326, 358

Archimedian model (アルキメデスモデル) 37

goal programming (目標計画法) も参照ください

array numbering (配列番号付け) 267

artificial variables (人為変数) 310

AUTOPERTURB 256, 290

B

BACKTRACK 25, 256, 299

BARAASIZE 302

BARCROSSOVER 302

BARDENSECOL 302

BARDUALINF 303

BARDUALOBJ 303

BARDUALSTOP 22, 257

BARGAPSTOP 22, 257, 260

BARITER 303

BARITERLIMIT 15, 258

BARLSIZE 303

BARMEMORY 258

BARORDER 21, 259

BAROUTPUT 22, 34, 259

BARPRIMALINF 303

BARPRIMALOBJ 304

BARPRIMALSTOP 22, 260

barrier method (内点法) . Newton barrier (ニュートンバリア) も参照ください

BARSTEPSTOP 22, 260

BARSTOP 304

BARTHREADS 261

basis (基底) 11, 240, 275

inversion (反転) 274

loading (ロード) 151, 163

reading from file (ファイルからの読み込み) 182

returning basis (基底のリターン) 121

batch mode (バッチモード) 5, 235

BESTBOUND 304

BIGM 261, 289

BIGMMETHOD 262

binary variables (バイナリ変数)

probing (詳細調査) 282

bitmaps (ビットマップ) 112, 230

bound tightening (上下界値強化) 32

BOUNDNAME 304

bounds (限度) 59, 114, 139, 226, 358

Branch and Bound (分枝限定法) 3, 23

branching (分枝) 55, 206

directions (方向) 103, 185, 357

variable (変数) 198

BREADTHFIRST 25, 262, 287

C

C 267

CACHESIZE 21, 263

callbacks (コールバック) 33

barrier log (バリアログ) 196

branching variable (分枝変数) 198

copying between problems (問題間コピー) 72

cut manager (カットマネージャ) 204

estimate function (推定機能) 206

global log (グローバルログ) 209

initialization of cut manager (カットマネージャの初期化) 213

integer solutions (整数解) 214

node cutoff (ノードカットオフ) 220

node selection (ノード選択) 201

optimal node (最適ノード) 211, 222

output (出力) 218

preprocess node (事前処理ノード) 224

separate (分割) 226

Xpress-Optimizer Reference Manual Index 362

simplex log (シンプレックスログ) 216

terminating global search (整数探索の終了) 208

Cholesky factorization (コレスキー因子分解) 21, 259, 263, 269, 303

CHOLESKYALG 263

CHOLESKYTOL 263

coefficient tightening (係数強化) 32

COLS 305

columns (列)

deleting (削除) 76

density (密度) 269, 302

extra columns (追加列) 270

names (名前) 48

nonzeros (非零要素) 93

ranges (レンジ) 91

returning bounds (限度のリターン) 114, 139

returning indices (添字の返却) 108

returning names (名前の返却) 115

types (タイプ) 95

comments (コメント) 285

Console Mode (コンソールモード) 1, 41

Console Xpress (コンソールXpress) 1

batch mode (バッチモード) 5

command line options (コマンドラインのオプション) 4

errors (エラー) 317

functions (機能、関数) 41

terminating optimization (最適化の終了) 15

termination (終了) 235

control parameter (制御パラメータ) controls (制御)を参照ください
controls (制御) 43, 255
changing values (値の変更) 255
copying between problems (問題間のコピー) 73
retrieve values (値の回復) 138
retrieving values (値の回復) 102, 112
setting values (値の設定) 228, 230, 234
COVERCUTS 264, 298
CPMAXCUTS 265, 328, 334
CPMAXELEMS 265, 328, 334
CPUTIME 265
CRASH 266
CROSSOVER 22, 266
crossover (クロスオーバー) 22, 190, 266, 302
CSTYLE 267
csv (カンマ区切りデータ) 340
Curtis-Reid scaling (Curtis-Reidスケーリング) 297
cut manager (カットマネージャ) 34
initialization (初期化) 213
routines (ルーチン) 36, 204
cut pool (カットプール) 35, 46, 77, 205, 226, 328
bounds (限定, 限度) 236
cuts (カット) 153, 238, 265
lifted cover inequalities 264
list of cuts (カトリスト) 98
list of indices (添字リスト) 96
number of coefficients (係数の数) 265
cut strategy (カット戦略) 268
CUTDEPTH 267
CUTFREQ 267
cutoff (カットオフ, 足切り) 24, 26, 220, 280, 281, 283, 307
CUTS 79, 305
cuts (カット) 35, 46, 226, 325, 327
deleting (削除) 78
generation (生成) 267
Gomory cuts (Gomoryカット) 273, 299
list of active cuts (アクティブカトリスト) 100
model cuts (モデルカット) 162
CUTSTRATEGY 268
cutting planes (カット面) cutsを参照ください
D
default algorithm (デフォルトアルゴリズム) 268
DEFAULTALG 19, 177, 268
degradation (劣化) 25, 34, 206, 269, 293, 299
DEGRADEFACTOR 269
DENSECOLLIMIT 21, 269
Devex pricing (Devexプライシング) pricingを参照ください

directives (ディレクティブ, 指示, 指令) 11, 25, 103, 165, 326, 328
loading (ロード) 154
read from file (ファイルからの読み込み) 184
dongles (ダングル) 4
dual values (双対値) 11
DUALINFfeas 305
Xpress-Optimizer Reference Manual Index 363

E

ELEMS 305
ELIMTOL 269
ERRORCODE 306, 318
errors (エラー) 43, 219, 231, 285, 306, 317
checking (チェック) 149
messages (メッセージ) 34
ETATOL 270
exit codes (終了コード) return codes (リターンコード) を参照ください
EXTRACOLS 30, 270, 329, 333
EXTRAELEMS 30, 54, 271, 327, 333
EXTRAMIPENTS 30, 271, 334
EXTRAPRESOLVE 31, 272, 329
EXTRAROWS 30, 272, 325, 333

F

fathoming (深さ測定) 23
feasible region (実行可能領域) 19
FEASTOL 273
files (ファイル)
alter (.alt) (変更) 53, 339
basis (.bss) (基底) 182, 240, 324, 339
binary matrix (.bif) (バイナリ行列) 4, 326, 339
cut pool (.ctp) (カットプール) 339
directives (.dir) (ディレクティブ) 25, 184, 339, 357
formats (様式) 339
global (.glb) (大域的探索) 143, 191, 320, 339
goal input (.gol) (目標入力) 144, 339
goal output (.grp) (目標出力) 339
IIS (.iis) (既約実行不可能集合) 147, 339
initialization (.ini) (初期化) 4
log (.log) (ログ) 10
LP matrix (.lp) (LP行列) 1, 187, 339
Matrix Alteration (.alt) (行列変更) 358
MPS matrix (.mat) (MPS行列) 187, 285, 339, 340
OMNI format (OMNI形式) 242
password (.pwd) (パスワード) 4, 149
range (.rng) (レンジ) 92, 129, 180, 340
range output (.rrt) (レンジ出力) 180, 246, 355
range output (.rsc) (レンジ出力) 250, 340, 354
save/restore (.svf) (保存) 191, 193, 340

solution (.asc) (解) 252, 339, 349
 solution (.prt) (解) 248, 340, 350
 solution (.sol) (解) 4, 32, 191, 324, 340
 solution header (.hdr) (解ヘッダ) 250, 252, 339, 348
 FIXGLOBAL 83, 181
 Forrest-Hirst-Tomlin Criterion (Forrest-Hirst-Tomlin基準) 25, 256
 Fortran (フォートラン) 267
 ftran 86

G

GLOBAL 10, 11, 84, 141, 175
 global entities (離散要素) 308, 315
 branching (分枝) 55
 extra entities (追加要素) 271
 fixing (固定) 83
 loading (ロード) 155, 167
 global log (グローバルログ) 209, 359
 global search (グローバル探索) 23, 24, 34, 80, 310, 335
 begin search (探索開始) 141
 branching variable (分枝変数) 198
 callbacks (コールバック) 34
 directives (ディレクティブ) 11, 184
 infeasible node (実行不能コード) 211
 integer solutions (整数解) 214
 MIP solution status (混合型整数計画問題状態) 309
 node selection (ノード選択) 201
 reinitialize search (最初期化探索) 150
 termination (終了) 208, 280, 283
 GOAL 37, 144
 goal programming (目標計画法) 37, 144, 328
 using constraints (制約) 37
 using objective functions (目的関数) 39
 GOMCUTS 273, 299

H

Haverly Systems (Haverlyシステム) 243
 HELP 146
 Hessian matrix (ヘシアン行列) 65, 68, 126
 Xpress-Optimizer Reference Manual Index 364

I

IIS (既約実行不可能集合) 306
 IIS 16, 147, 306
 infeasibility (実行不能) 16, 19, 140, 147, 291, 335
 diagnosis (診断) 298
 integer (整数) 17, 308
 node (ノード) 211
 infinity (無限) 45
 initialization (初期化) 4, 88, 149, 328
 integer preprocessing (整数事前分析) 27, 282

integer presolve (整数事前分析) 335
integer programming (整数計画) 2, 23, 32
integer solutions (整数解) 275, 278, 308, 309
begin search (探索開始) 141
branching variable (分枝変数) 198
callback (コールバック) 214
cutoff (カットオフ、足切り) 220
loading quadratic problem (2 次問題のロード) 167
node selection (ノード選択) 201
reinitialize search (最初期化探索) 150
retrieving information (情報の復帰) 105
terminating search (探索終了) 208
global search (整数探索) も参照ください
interior point algorithm (内点法) Newtonbarrier (ニュートンバリア) を参照ください
inversion (反転) 20
INVERTFREQ 20, 274
INVERTMIN 20, 274
irreducible infeasible sets (可約実行不能集合) 16, 278, 306
begin search (探索開始) 147
retrieving (復帰) 107
sub-IIS isolation (部分既約実行不可能集合分離) 16
iteration log (繰返しログ) 276

K

KEEPALLCUTS 264
KEEPBASIS 275
KEEPMIPSOL 143, 253, 275
KEEPNROWS 276, 359

L

license (ライセンス) 9
lifted cover inequalities 298
line length (ライン幅) 337
log file (ログファイル) 231
logical preprocessing (論理事前処理) 282
LP relaxation (LP緩和) 23, 360
LPITERLIMIT 15, 31, 276, 319
LPLOG 21, 34, 216, 276
LPOBJVAL 12, 306
LPSTATUS 307

M

Markowitz tolerance (マルコヴィッツ許容範囲) 269, 277
MARKOWITZTOL 277
matrix (行列)
adding names (名前追加) 10
changing coefficients (係数変更) 53, 61, 63, 69
column bounds (列の上下界値) 59
columns (列) 29, 44, 76, 305
constraint senses (制約意味) 53

cuts (カット) 305
deleting cuts (カット削除) 78
elements (要素) 290
extra elements (追加要素) 270, 271, 272
input (入力) 10, 159, 186
modifying (改良) 29, 30
nonzeros (非零要素) 93, 130
quadratic elements (2 次要素) 312
range (レンジ) 70
rows (列) 29, 71, 81, 313
scaling (スケーリング) 194
sets (集合) 314
size (サイズ) 31
space (スペース) 31
spare columns (予備列) 314
spare elements (予備要素) 314, 333
spare global entities (予備離散要素) 315
spare rows (予備行) 315
viewing (検討) 29
MATRIXNAME 307
Xpress-Optimizer Reference Manual Index 365
MATRIXTOL 277
MAXCUTTIME 277
MAXIIS 16, 107, 147, 278
MAXIM 11, 175
MAXMIPSOL 278
MAXNODE 278
MAXPAGELINES 279
MAXSLAVE 279
MAXTIME 15, 31, 279
memory (メモリ) 82, 85, 291, 319, 320, 322, 326
MINIM 11, 175
MIPABSCUTOFF 280
MIPABSSTOP 280
MIPADDCUTOFF 26, 281
MIPENTS 308
MIPINFEAS 308
MIPLOG 34, 281, 359
MIPOBJVAL 12, 308
MIPPRESOLVE 27, 282
MIPRELCUTOFF 26, 283
MIPRELSTOP 283
MIPSOLNODE 308
MIPSOLS 309
MIPSTATUS 309
MIPTARGET 25, 256, 284
MIPTOL 284
model cuts (モデルカット) 185

MPS file format (MPSファイル形式) 285

filesも参照ください

MPSBOUNDNAME 284

MPSECHO 285

MPSERRIGNORE 285

MPSFORMAT 285

MPSNAMELENGTH 286

MPSOBJNAME 286

MPSRANGENAME 286

MPSRHSNAME 287

N

NAMELENGTH 310

Newton barrier (ニュートンバリア) 21, 175

controlling performance (性能操作) 21

convergence criterion (収束基準) 22, 304

crossover (クロスオーバー) 22

log callback (ログコールバック) 196

number of iterations (繰返し回数) 15, 21, 258

number of threads (スレッド回数) 261

output (出力) 22, 34

NODEDEPTH 310

NODES 310

nodes (ノード) 24

active cuts (アクティブノード) 100, 153

cut routines (カッターチン) 204

deleting (削除) 80

deleting cuts (カット削除) 78

depth (深さ) 310

infeasibility (実行不能) 211

maximum number (最大数) 278

number solved 310

optimal (最適) 222

outstanding (未検討) 302

parent node (親ノード) 78, 311

prior to optimization (事前処理) 224

selection (選定) 24, 34, 201, 287

separation (分離) 226

NODESELECTION 25, 262, 287

nonbinding rows (非束縛行) 276

nonlinear programming (非線型計画) 189

numerical difficulties (数値困難) 360

O

objective function (目的関数) 19, 29, 284, 286, 310, 311

changing coefficients (係数変更) 67

dual value (双対値) 303

optimum value (最適値) 12, 306, 308

primal value (主値) 304

quadratic (2 次問題) 29, 65, 68, 167, 171
retrieving coefficients (係数復帰) 117
OBJFIXED 67, 310
OBJNAME 311
OBJRHS 311
Xpress-Optimizer Reference Manual Index 366
OBJSENSE 311
OMNI format (OMNI形式) 242
OMNIDATANAME 287
OMNIFORMAT 288
optimal basis (最適基底) 11, 22, 36
OPTIMALITYTOL 288
optimization (最適化)
standard template (標準テンプレート) 9
optimization sense (最適化意図) 311
Optimizer output (Optimizer出力) 10, 20, 22, 218
OUTPUTLOG 219, 231, 247, 249, 288
OUTPUTMASK 251, 253, 289
OUTPUTTOL 289

P

parallel MIP (並列 MIP) 279
PARENTNODE 311
partial pricing (部分プライシング) pricing (プライシング) を参照ください
PENALTY 289
performance (性能) 31, 325, 327
PERTURB 256, 290
pivot (ピボット) 178, 296, 334
list of variables (変数リスト) 119
order of basic variables (基底変数順序) 118
PIVOTTOL 290
postoptimal analysis (事後分析) 180
postsolve (事後分析) 31
PPFACTOR 290
pre-emptive model (先制モデル) 37
goal programming (目標計画) も参照ください
PRESOLVE 31, 54, 291, 323, 326
presolve (事前分析) 16, 29, 31, 174, 269, 272, 291, 298, 326, 329
diagnosing infeasibility (実行不能診断) 16
integer (整数) 27
presolved problem (事前分析済み問題) 133
basis (基底) 121, 163
directives (ディレクティブ , 指示 , 指令) 103, 165
PRESOLVEOPS 292
PRESOLVESTATE 29, 312
pricing (プライシング) 292
Devex 20, 292, 322
partial 20, 290, 292
PRICINGALG 20, 292, 322

primal infeasibilities (主実行不能) 303, 312, 315
PRIMALINFEAS 312
PRINTRANGE 247
PRINTSOL 249
priorities (優先度) 103, 185, 321, 357
problem (問題)
creation (作成) 75
deletion (削除) 82
file access (ファイルアクセス) 186, 244
input (入力) 10, 159
name (名前) 4, 125, 232
pointers (ポインタ) 10
solving (求解) 11, 175
problem attributes (問題特性) 11, 301
prefix (接頭辞) 301
retrieving values (復帰値) 101, 111, 137
problem pointers (問題ポインタ) 75
copying (コピー) 74
deletion (削除) 82
pseudo cost (擬コスト) 25, 103, 185, 293, 299, 357
PSEUDOCOST 293

Q

QLEMS 312
quadratic programming (2次計画問題) 329, 330
coefficients (係数) 65, 68, 126, 312
loading global problem (整数計画問題のロード) 167
loading problem (問題のロード) 171
QUIT 179, 235

R

RANGE 11, 84, 180, 246, 250
RANGENAME 312
ranging (レンジ) 11, 70, 71, 91, 312
information (情報) 11, 180
name (名前) 286
retrieve values (値の復帰) 128, 129
READBASIS 11, 182
Xpress-Optimizer Reference Manual Index 367
READDIRS 11, 184, 357
READPROB 10, 18, 186
RECCONVERGE 313
RECEXPAND 293
RECMAXPASSES 293, 320
RECSHRINK 294
RECSTOP 294
RECURSE 189
recursion (反復) 189, 294, 313, 320, 323, 326
convergence (収束) 294

reduced costs (被約費用) 11, 83, 282, 288
REFACTOR 295
REL10STYLE 295
relaxation (緩和) LP relaxation (LP緩和) を参照ください
RELPIVOTTOL 296
RESTORE 191
return codes (リターンコード) 43, 179, 235, 317
RHSNAME 313
right hand side (右辺) 69, 127
name (名前) 287
ranges (レンジ) 180
retrieve range values (レンジ値の復帰) 128
rounding errors (使用回エラー) 17
ROWS 313
rows (行)
addition (追加) 50
deletion (削除) 81
extra rows (追加行) 272, 315
indices 108
model cuts (モデルカット) 162
names (名前) 48, 115
nonbinding (非束縛) 276
nonzeros (非零要素) 130
number (番号) 313
types (タイプ) 71, 132
running time (計算時間) 279

S

SAVE 191, 193
SBBEST 296
SBITERLIMIT 296
SCALE 18, 194
SCALING 18, 195, 297
scaling (スケーリング) 17, 194, 325, 327
Curtis-Reid 297
geometric mean (幾何平均) 297
maximum element (最大要素) 297
security system (セキュリティシステム) 9, 330
sensitivity analysis (感度分析) 84
separation (分離) 23
SETMEMBERS 314
SETS 314
sets (集合) 308, 314
names (名前) 52
shadow prices (シャドウプライス) 181
simplex (シンプレックス法) 19, 22, 175
inversion (反転) 20
log callback (ログコールバック) 216

number of iterations (繰返し回数) 15, 313
output (出力) 20, 34
perturbation (摂動) 256
pivot (ピボット) pivot (ピボット) を参照ください
type of crash (クラッシュタイプ) 266
SIMPLEXITER 313
slacks (スラック) 79
slave processors (従プロセッサ) 279
solution (解) 11, 12, 15
beginnin
g search (探索開始) 175
output (出力) 135, 248, 252
SOLUTIONFILE 32, 297
SOSREFTOL 298
SPARE
COLS 314
SPAREELEMS 314
SPAREMIPENTS 315
SPAREROWS 315
special ordered sets (特殊順序集合) 3, 155, 167
STOP 179, 235
strong branching (強力な分枝) 296
student mode (スチューデントモード) 4, 325
sub-IIS isolation (部分既約実行不可能集合分離) 16
successive linear programming (逐次線形計画)
(反転) を参照ください
SUMPRIMALINF 315
Xpress-Optimizer Reference Manual Index 368

T

tightening (強化)
bound (限度 , 上下界値) 32
coefficient (係数) 32
tolerance (許容範囲) 273, 277, 280, 284, 288, 289, 290, 296
TRACE 16, 298
tracing (トレーシング) 335
tree search (木探索) global search (整数探索) を参照ください
TREECOVERCUTS 298
TREETGOMCUTS 299

U

unboundedness (非有界 , 無限解) 17, 23, 140

V

variables (変数)
artificial (人為) 310
binary (バイナリ) 3, 155, 167, 346
dual (双対) 109
infeasible (実行不能) 133
integer (整数) 3, 155, 167, 346
partial integer (部分整数) 3, 62, 155, 167, 346

primal (主) 109
selection (選択) 24
semi-continuous (半連続) 3, 62, 155, 167, 346
semi-continuous integer (反連続整数) 3, 62, 155, 167
slack (スラック) 11, 79
VARSELECTION 25, 299
VERSION 299
version number (バージョン番号) 299

W

warning messages (警告メッセージ) 34
WRITEBASIS 11, 240
WRITEOMNI 242
WRITEPROB 244
WRITEPRTRANGE 11, 246
WRITEPRTSOL 11, 248
WRITERANGE 11, 250
WRITESOL 11, 252, 348

X

Xpress-BCL 3
Xpress-Mosel 3
Xpress-MP Essentials 4, 12
XPRS_MINUSINFINITY 45, 78, 96
XPRS_PLUSINFINITY 45
XPRSaddcols 30, 44
XPRSaddcuts 35, 46
XPRSaddnames 10, 45, 48
XPRSaddrows 30, 50
XPRSaddsetnames 52
XPRSalter 53, 326, 358
XPRSbranchcut 55
XPRsbtran 57
XPRSchgbounds 30, 59
XPRSchgcoef 30, 61
XPRSchgcoltype 30, 62
XPRSchgmcoef 30, 61, 63
XPRSchgmqobj 30, 65
XPRSchgobj 30, 67, 332
XPRSchgqobj 30, 68
XPRSchgrhs 30, 69
XPRSchgrhsrange 30, 70
XPRSchgrowtype 30, 71
XPRScopycallbacks 72, 74
XPRScopycontrols 73, 74
XPRScopyprob 32, 74
XPRScreateprob 10, 75
XPRSDelcols 30, 76
XPRSDelcpcuts 36, 77
XPRSDelcuts 35, 77, 78

XPRSdelnode 80
XPRSdelrows 30, 81
XPRSdestroyprob 10, 75, 82
XPRSsetcbmessageVB 219
XPRSfixglobal 83, 181
XPRSfree 10, 85
XPRSgetbanner 88
Xpress-Optimizer Reference Manual Index 369
XPRSgetbasis 89
XPRSgetcolrange 30, 91
XPRSgetcols 30, 93
XPRSgetcoltype 30, 95
XPRSgetcpcutlist 36, 96
XPRSgetcpcuts 36, 98
XPRSgetcutlist 36, 100
XPRSgetdblattr 12, 101, 301
XPRSgetdblcontrol 102
XPRSgetdirs 103
XPRSgetglobal 105
XPRSgetiis 16, 107
XPRSgetindex 108, 335
XPRSgetinfeas 109
XPRSgetintattr 12, 111, 301
XPRSgetintcontrol 112, 255
XPRSgetlasterror 113
XPRSgetlb 30, 114
XPRSgetnames 29, 115
XPRSgetobj 29, 117, 332
XPRSgetpivotorder 118
XPRSgetpivots 119
XPRSgetpresolvebasis 33, 121
XPRSgetpresolvesol 123
XPRSgetprobname 125
XPRSgetqobj 29, 126
XPRSgetrhs 29, 127
XPRSgetrhsrange 29, 128
XPRSgetrowrange 29, 129
XPRSgetrows 29, 130
XPRSgetrowtype 29, 132
XPRSgetscaledinfeas 33, 133
XPRSgetsol 11, 32, 135
XPRSgetstrattr 12, 137, 301
XPRSgetstrcontrol 138
XPRSgetub 30, 139
XPRSgetunbvec 140
XPRSglobal 10, 11, 84, 141, 150, 175
XPRSgoal 37, 144
XPRSiis 16, 107, 147
XPRSinit 9, 75, 85, 88, 149

XPRSinitglobal 142, 150
XPRSloadbasis 11, 151
XPRSloadcuts 35, 153
XPRSloaddirs 154
XPRSloadglobal 10, 155
XPRSloadlp 10, 159
XPRSloadmodelcuts 162
XPRSloadpresolvebasis 33, 163
XPRSloadpresolvedirs 33, 165
XPRSloadqglobal 10, 167
XPRSloadqp 10, 171
XPRSloadsecurevecs 174
XPRSmxim 11, 175
XPRSmnim 11, 175
XPRSpivot 178
XPRSrange 11, 84, 91, 129, 180, 246, 250
XPRSreadbasis 11, 182
XPRSreaddirs 11, 184, 357
XPRSreadprob 10, 18, 186
XPRSrecurse 189, 313
XPRSrestore 191
XPRSsave 191, 193
XPRSscale 18, 194
XPRSsetcbbarlog 22, 34, 196
XPRSsetcbchgbranch 34, 198
XPRSsetcbchgnode 34, 201
XPRSsetcbcutlog 203
XPRSsetcbcutmgr 36, 204
XPRSsetcbestimate 34, 206
XPRSsetcbfreecutmgr 36, 208
XPRSsetcbgloballog 34, 209
XPRSsetcbinfnode 34, 211
XPRSsetcbinitcutmgr 36, 213
XPRSsetcbintsol 34, 214
XPRSsetcblplog 21, 34, 216
XPRSsetcbmessage 10, 34, 218, 231
XPRSsetcbnodecutoff 34, 220
XPRSsetcboptnode 34, 222
XPRSsetcbprenode 34, 224
XPRSsetcbsepnode 34, 55, 226
XPRSsetdblcontrol 228
XPRSsetdefaults 229
XPRSsetintcontrol 230, 255
Xpress-Optimizer Reference Manual Index 370
XPRSsetlogfile 10, 21, 22, 231
XPRSsetprobname 232
XPRSsetstrcontrol 234
XPRSstorebounds 236
XPRSstorecuts 36, 238

XPRWritebasis 11, 240
XPRWriteomni 242
XPRWriteprob 244
XPRWriteprtrange 11, 246
XPRWriteprtsol 11, 248
XPRWriterange 11, 250
XPRWritesol 11, 252, 348
XPRWriteprtsol 11, 248
XPRWriterange 11, 250
XPRWritesol 11, 252, 348