Vol.2011-AL-133 No.4
2011/1/12

離散凸最適化ソルバと デモンストレーションソフトウェアの公開

土村展之[†1] 森口聡子[†2] 垣村尚徳[†3]
岩田覚[†4] 室田一雄[†3]

マトロイド・劣モジュラ関数の研究の流れを汲んだ離散凸解析による統一的枠組みの中で，離散最適化問題を効率的に解く各種アルゴリズムが提案されてきた．離散凸解析理論の普及を目的として，これらの成果を応用分野の研究者・実務家が享受できるように，関連するアルゴリズムを紹介するデモンストレーションソフトウェアと，アプリケーションソフトウェアを開発した．ソルバ及びデモンストレーションソフトウェアの WEB 公開について報告する．

Releasing Discrete Convex Optimization Solvers and Demonstration Softwares

Nobuyuki Tsuchimura,[†1] Satoko Moriguchi,[†2]
Naonori Kakimura,[†3] Satoru Iwata[†4]
and Kazuo Murota [†3]

In discrete convex analysis, which is a unified framework of discrete convex functions based on the theory of matroids and submodular functions, efficient discrete optimization algorithms were proposed. In order to disseminate these theoretical results in application fields, we release softwares and web applications implementing algorithms developed in discrete convex analysis.

1. はじめに

整数変数の非線形計画は，様々な分野や現実問題に数多く現れるが，解くのは非常に困難であることが知られており，効率的な汎用ソルバも存在していなかった．整数変数の非線形計画問題，難しい組合せ最適化問題を解く鍵として，離散凸関数の概念が考えられ，主に離散格子点上で定義された関数に対して様々な研究者により定義がなされ，理論研究が展開されてきた．その中でも，マトロイド・劣モジュラ関数の研究の流れを汲んだ離散凸解析による統一的枠組みが 90 年代以降注目され，今や多くの成果が報告されている[9),10),13)]．離散凸解析理論では，L 凸性と M 凸性という互いに共役な二種類の離散凸性が存在し，中心的な役割を担っている．近年の離散凸解析理論の研究により，非線形離散関数の中でも効率よく多項式時間で最小化できるクラスが解明されてきて，これまで，L 凸性，M 凸性を有する最適化問題に対して，効率的なアルゴリズムが研究されてきた．

現実問題で現れる非線形離散関数の例として，在庫管理と離散凸解析の関係性が挙げられる．在庫管理の理論は，古くから多くの研究がなされており，さらに，省在庫と環境負荷低減の観点から，近年重要視しなくてはならないテーマとなった．近年の要請を満たすためには，大規模な問題を扱わなくてはならなくなる．このような背景で様々な在庫モデルが存在するが，予備品在庫管理問題を一例として取り上げたい．この問題は，あらかじめ予備品を購入しておくための代金と，品切れ時のバックオーダーに対する罰金からなるコストをできるだけ減らしたいというモデルで，従来は「Miller の離散凸関数」[4)] の最小化アルゴリズムによる求解が試みられていたが，このアルゴリズムの理論計算量は効率的なものではなく，極めて小規模な問題しか解けていなかった．また，非線形離散関数の最小化を行える高性能な汎用ソルバが存在しないため，これまでは厳密解法による予備品在庫管理は現実的でなかった．しかしながら，近年の離散凸解析の知見から，この問題の多項式時間での求解が可能であることが明らかになった．我々の研究では，高速な L 凸関数最小化ソルバの開発に

†1 関西学院大学
Kwansei Gakuin University
†2 産業技術大学院大学
Advanced Institute of Industrial Technology
†3 東京大学
The University of Tokyo
†4 京都大学
Kyoto University

より，インタラクティブな条件パラメータの変更・調整に対して最適在庫量を求解できる予備品在庫管理のための WEB アプリケーションを提供することにより，この分野への離散凸解析理論の普及を試みたい．

本研究では，離散凸解析理論の普及を目的として，その成果を様々な応用分野の研究者・実務家が享受できるように，関連するアルゴリズムを紹介するデモンストレーションソフトウェアと，アプリケーションソフトウェアを開発し，WEB 公開を行った．これまでは，予備知識がなければ離散凸解析の研究に携わることも，成果を利用することも難しかった．広く応用研究が成功している線形計画問題と同様に，データを生成すれば，アルゴリズムの詳細を理解しなくても離散凸解析関連の研究や応用事例が行えるようになることを目指し，教育，研究，応用の題材として普及させるための取り組みについて述べる．

現段階で公開しているソフトウェアの例としては，離散凸関数の最小化ソルバや，インタラクティブに離散 2 次凸関数を最小化できるオンラインソルバ，前述の予備品在庫管理アプリケーションなどが挙げられる．これらを抜粋して取り上げる．

2. 離散凸解析

連続最適化の分野では，凸関数の理論，凸解析による理論体系が大きな貢献をもたらしている．離散最適化においても，その離散版を確立することで効率的な解法を得ようと，離散的な凸性に纏わる様々な概念が提案されてきた．たとえば，Miller[4] は，凸関数の持つ性質である「最小値の局所的特徴づけ」に着目して，次のように，「Miller の離散凸関数」を定義した．

点 $x \in \mathbf{R}^n$ の整数近傍を

$$N(x) = \{y \in \mathbf{Z}^n \mid \lfloor x(i) \rfloor \leq y(i) \leq \lceil x(i) \rceil, i = 1, 2, \ldots, n\}$$

と定義する．関数 $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ は任意の $x, y \in \mathbf{Z}^n$ と任意の $\alpha \in \mathbf{R}, 0 \leq \alpha \leq 1$ に対して

$$\min\{f(z) \mid z \in N(\alpha x + (1-\alpha)y)\} \leq \alpha f(x) + (1-\alpha) f(y)$$

を満たすとき，Miller の離散凸関数といわれる[4]．

「Miller の離散凸関数」は，「凸関数の離散版」を考える上で一般的に望まれる性質である「凸拡張可能性」を有していない上に，効率的な最小化アルゴリズムが構築できていない．他にも離散的な凸性の概念が様々提案されたが，それらは「凸解析の離散版」と言える統一的な枠組みとしては進展しなかった．本研究で扱う「離散凸解析」の枠組みは，凸関数と劣モジュラ関数，マトロイド理論の研究を背景として認識されたパラダイムで，「最小値の局所的特徴づけ」，「凸拡張可能性」に加えて，双対性・共役性も満足する統一的な枠組みである．ここでは，離散凸解析において中心的な役割を担い，互いに共役な二つの離散凸性である L 凸性，M 凸性について述べる．その他の概念については，文献[9),10)] を参照されたい．

2.1 劣モジュラ関数と L 凸性

本節では，劣モジュラ性に基づいて導入された L 凸性の概念について述べる．

ベクトル $x, y \in \mathbf{Z}^n$ に対して，成分ごとに最大値，最小値をとって得られるベクトルを $x \vee y, x \wedge y$ と書くことにし，$\mathbf{1} = (1, 1, \ldots, 1) \in \mathbf{Z}^n$ とする．関数 $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ が L 凸関数であることは，f が 2 条件

(SBF) $f(x) + f(y) \geq f(x \vee y) + f(x \wedge y)$ $(x, y \in \mathbf{Z}^n)$,

(TRF) $\exists r \in \mathbf{R}:\ f(x+\mathbf{1}) = f(x) + r$ $(x \in \mathbf{Z}^n)$,

を満たすことと定義される．ここで不等式 (SBF) は $f(x)$ か $f(y)$ が $+\infty$ であるときには成立していると約束する．

離散凸解析では，L 凸関数と等価な概念である $\mathrm{L}^\natural$ 凸関数も重要である．関数 $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ が $\mathrm{L}^\natural$ 凸関数であるとは，

$$f(x_1, \ldots, x_n) = \tilde{f}(0, x_1, \ldots, x_n) \tag{1}$$

となる L 凸関数 $\tilde{f}(x_0, x_1, \ldots, x_n)$ が存在することである．$\mathrm{L}^\natural$ 凸関数は Miller の離散凸関数である．

実数ベクトルを変数とする関数についても L 凸性は定義されている．関数 $f : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ が連続 L 凸関数であることは，f が 2 条件

(SBF[R]) $f(x) + f(y) \geq f(x \vee y) + f(x \wedge y)$ $(x, y \in \mathbf{R}^n)$,

(TRF[R]) $\exists r \in \mathbf{R}, \forall x \in \mathbf{R}^n, \forall \alpha \in \mathbf{R} : f(x+\alpha\mathbf{1}) = f(x) + \alpha r$,

を満たすことと定義される．連続 L 凸関数から (1) 式に基づいて連続 $\mathrm{L}^\natural$ 凸関数も定義されれている．

$\chi_X \in \{0, 1\}^n$ を集合 $X \subseteq \{1, 2, \ldots, n\}$ の特性ベクトル，すなわち

$$\chi_X(v) = \begin{cases} 1 & (v \in X) \\ 0 & (v \notin X) \end{cases}$$

とする．$\mathrm{L}^\natural$ 凸関数の最小性規準は次の定理で与えられている．

定理 2.1 ($\mathrm{L}^\natural$ 凸関数最小性規準[9),10)])．$f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ は $\mathrm{L}^\natural$ 凸関数とする．任意の $x \in \operatorname{dom} f$ に対し，

$$f(x) \leq f(y)\ (\forall y \in \mathbf{Z}^n) \quad \Longleftrightarrow \quad f(x) \leq f(x \pm \chi_X)\ (\forall X \subseteq \{1, 2, \ldots, n\}).$$

ここで，上記の局所最小性の判定は二つの劣モジュラ集合関数

$$\rho_x^+(X) := f(x+\chi_X) - f(x),$$
$$\rho_x^-(X) := f(x-\chi_X) - f(x)$$

の最小化に帰着できる．すなわち，劣モジュラ集合関数最小化アルゴリズム[2),12)] により，多項式時間で調べることができる．

2.2 マトロイドと M 凸性

本節では，マトロイドの交換公理に基づく M 凸性の概念について述べる．

$$\mathrm{dom}\, f = \{x \in \mathbf{Z}^n \mid f(x) < +\infty\}, \qquad \mathrm{dom}_{\mathbf{R}} f = \{x \in \mathbf{R}^n \mid f(x) < +\infty\},$$
$$\mathrm{supp}^+(x-y) = \{w \in \{1,2,\ldots,n\} \mid x(w) > y(w)\},$$
$$\mathrm{supp}^-(x-y) = \{w \in \{1,2,\ldots,n\} \mid x(w) < y(w)\}$$

とおき，$\chi_i \in \{0,1\}^n$ は $i = 1,2,\ldots,n$ に対する単位ベクトルの特性ベクトル，すなわち

$$\chi_i(v) = \begin{cases} 1 & (v = i) \\ 0 & (v \neq i) \end{cases}$$

とする．便宜上，$i = 0$ のときは $\chi_i = (0,0,\ldots,0)$ と約束する．

関数 $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ が M 凸関数であるとは，f が交換公理 (M-EXC[$\mathbf{Z}$]) を満たすことである．

(M-EXC[$\mathbf{Z}$]) 任意の $x, y \in \mathrm{dom}\, f$ と任意の $u \in \mathrm{supp}^+(x-y)$ に対して，ある $v \in \mathrm{supp}^-(x-y)$ が存在し，

$$f(x) + f(y) \geq f(x - \chi_u + \chi_v) + f(y + \chi_u - \chi_v).$$

M 凸関数の実効定義域は成分和が一定の超平面の上に乗っているので，ある座標方向に沿って射影しても情報量は失われない．1 次元高い空間内で定義された M 凸関数から射影によって得られる関数は $\mathrm{M}^\natural$ 凸関数と呼ばれる．すなわち，関数 $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ が $\mathrm{M}^\natural$ 凸関数であるとは，

$$\tilde{f}(x_0, x_1, \ldots, x_n) = \begin{cases} f(x) & (x_0 = -\sum_{i=1}^n x_i) \\ +\infty & (x_0 \neq -\sum_{i=1}^n x_i) \end{cases} \tag{2}$$

で定義される関数 $\tilde{f}(x_0, x_1, \ldots, x_n)$ が M 凸関数となることである．$\mathrm{M}^\natural$ 凸関数は M 凸関数と等価な概念で，離散凸解析で同様に重要な概念である．

実数ベクトルを変数とする関数についても，M 凸性は定義されている．関数 $f : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ が連続 M 凸関数であるとは，f が交換公理 (M-EXC[$\mathbf{R}$]) を満たすことである．

(M-EXC[$\mathbf{R}$]) 任意の $x, y \in \mathrm{dom}_{\mathbf{R}} f$ と任意の $u \in \mathrm{supp}^+(x-y)$ に対して，ある $v \in \mathrm{supp}^-(x-y)$ と正の数 $\alpha_0 \in \mathbf{R}$ が存在し，すべての $\alpha \in [0, \alpha_0]$ に対して

$$f(x) + f(y) \geq f(x - \alpha(\chi_u - \chi_v)) + f(y + \alpha(\chi_u - \chi_v))$$

を満たす．

連続 M 凸関数から (2) 式に基づいて連続 $\mathrm{M}^\natural$ 凸関数も定義されている．

定理 2.2 (離散 $\mathrm{M}^\natural$ 凸関数最小性規準[9),10)])． $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ は離散 $\mathrm{M}^\natural$ 凸関数とする．任意の $x \in \mathrm{dom}\, f$ に対し，

$$f(x) \leq f(y)\ (\forall y \in \mathbf{Z}^n) \iff f(x) \leq f(x - \chi_u + \chi_v)\ (\forall u, v \in \{0,1,\ldots,n\}).$$

3. アルゴリズム

定理 2.1 から $\mathrm{L}^\natural$ 凸関数 f の，定理 2.2 から $\mathrm{M}^\natural$ 凸関数 f の，最小化に対する最急降下法が導かれる[3),9),10)]．

$\mathrm{L}^\natural$ 凸関数に対する最急降下法

手順 0: x を $\mathrm{dom}\, f$ に含まれる任意のベクトルとする．

手順 1: $\varepsilon \in \{1,-1\}$ と $X \subseteq \{1,\ldots,n\}$ を次のように決定する．

手順 1-1: X^+ を $\rho_x^+(X) := f(x+\chi_X) - f(x)$ の任意の最小解とする．

手順 1-2: X^- を $\rho_x^-(X) := f(x-\chi_X) - f(x)$ の任意の最小解とする．

手順 1-3:(ε, X) を以下のように定める:

$$(\varepsilon, X) = \begin{cases} (1, X^+) & \text{if } \min \rho_x^+ \leq \min \rho_x^- \\ (-1, X^-) & \text{if } \min \rho_x^+ > \min \rho_x^- \end{cases}$$

手順 2: もし $f(x) \leq f(x + \varepsilon\chi_X)$ ならば終了．x は f の最小解．

手順 3: $x := x + \varepsilon\chi_X$ とおく．手順 1 へ戻る． □

上記の最急降下法は多項式時間アルゴリズムではないが，最急降下法にスケーリング技法を適用した効率的な多項式時間アルゴリズムが，$\mathrm{L}^\natural$/ $\mathrm{M}^\natural$ 凸関数最小化それぞれに対して提案されている[6),9),10)]．

連続緩和を用いた実用上さらに高速な最小化法も提案された[8)]．離散 $\mathrm{L}^\natural$ 凸関数 $f : \mathbf{Z}^n \to \mathbf{R} \cup \{+\infty\}$ に対して，

$$f(x) = \bar{f}(x)\ \ (\forall x \in \mathbf{Z}^n) \tag{3}$$

を満たす連続凸関数 $\bar{f} : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ が与えられているとき，連続最適化の手法により $\bar{f}$ の最小解 $x' \in \mathbf{R}^n$ を求め，これを連続緩和解とする．x' の各成分に対して整数丸めを行って得られるベクトルを，最急降下法の手順 0 における初期点として採用する．

式 (3) を満たす連続凸関数 $\bar{f}$ が与えられているという仮定について，一般に，$L^\natural$ 凸関数の凸拡張可能性より，その存在性は示せるが，実装上見つけることは容易ではないことに注意が必要である．しかし，(4) 式のように，離散変数の関数 f が，連続変数の関数 $\bar{f}$ の変数を離散点に限定することで定義されている場合が多い．この連続緩和解からの最急降下法が理論的にも効率的であることを保証する近接定理も証明された[8]．

同様の結果が，M 凸性に対しても示された[7]．

4. WEB 公開

本研究では，前述の離散凸性（M 凸性/L 凸性）を有する関数の最小化アルゴリズムや関連する問題のアルゴリズムを実装し，離散凸関数最適化ソルバを始めとするアプリケーションソフトウェアを公開している．また，ソルバの動作を手軽に試せるように，いくつかは WEB 上のデモンストレーションソフトウェアとしても公開している．ここでは，その中から，離散凸関数最小化ソルバと，在庫管理アプリケーションを取り上げる．

4.1 開発ソルバ ODICON

我々の公開しているソルバの代表例として，ODICON[14](Optimization algorithms for DIscrete CONvex functions) を紹介する．このソルバは，離散凸関数最小化アルゴリズムの実装である．実装したアルゴリズムは $M/M^\natural/L/L^\natural$ 凸関数のそれぞれに対する最急降下法[3),9),10)]，スケーリング法[6),9),10)]，連続緩和法[7),8)] である．ODICON は C 言語によるオープンソースソフトウェアであり，単体で利用するよりも，むしろ別のプログラムに組み込まれることを想定している．最小化したい離散関数をもつ利用者は，その関数を C 言語上の関数として記述して，このソルバのしかるべきルーチンを呼び出せばよい．

一連のルーチンは，アルゴリズムの素直な実装を目指し，入出力インタフェースも自然なものになるように留意した．特に，C 言語において配列の要素数を，コンパイル時ではなく，実行時の状況に応じて決定しようとすると，その扱いに標準手法が確立されておらず，どのように実現するかは自明ではない．この点についても十分に検討して，標準となりうる手法を選ぶようにした．

このような工夫により，他人のソフトウェアを組み込む時にありがちな，どのように結合すればよいのかわからない，という問題を最小限にとどめている．

このソルバには以下のようなルーチンを用意している．

```
double mconv_minimize(int dim, double f(int dim, int x[]),
                      int init[], int lower[], int upper[]);
```

これは，dim 次元の M 凸関数 f() を最小化するルーチンである．すなわち，利用者は，最小化しようとする離散関数が $M/M^\natural/L/L^\natural$ 凸性のうち，どの離散凸性を有するかに従って，該当するルーチンを呼び出すことになる．上記の例では，初期解 init[] から探索を行い，最急降下法で最小解（複数ある場合はそのうちの 1 つ）にたどりついて，その最小解を init[] に入れ（元々の初期解は破壊される），最小値を関数自身の値として返す．探索範囲は lower[i]≤init[i]≤upper[i] ($0 \le i <$dim) に限定される．

上記の例で最小化する離散 M 凸関数は，次の型で宣言しておく．なお，最小化したい離散関数としては，いずれのルーチンでもこの型を受け取るよう統一している．

```
double f(int dim, int x[]);
```

例えば，次のような 3 次元の離散 M 凸関数

$$f(x) = x_0^4 + (x_1 - 3)^2 + 5(x_2 - 7)^2$$

を C 言語で実装すると，次のようになる．

```
double f(int dim, int x[]) {
    double r = 0;

    assert(dim == 3);  /* check dim */
    r += x[0] * x[0] * x[0] * x[0];
    r += (x[1] - 3) * (x[1] - 3);
    r += 5 * (x[2] - 7) * (x[2] - 7);
    return r;
}
```

この関数を，原点から探索して最小化するには，次のようにする．探索範囲は $-100 \le x_i \le 100$ とする．

```
int x[3]     = { 0, 0, 0 };
int lower[3] = { -100, -100, -100 };
int upper[3] = {  100,  100,  100 };
double min = mconv_minimize(3, f, x, lower, upper);
```

この呼び出しの後，min には最小値，x[] にはそれを実現する最小解の 1 つが代入される．

以上のようなルーチンを，離散 $M/M^\natural/L/L^\natural$ 凸関数の 4 種類の関数クラスの最小化に対して用意し，それぞれの関数クラスについて，複数の最小化アルゴリズム (最急降下法，スケーリング法，連続緩和法) を実装し利用者がアルゴリズムを指定できるようにした．ルー

チンの引数はほぼ共通しており，呼び出すルーチン名を変更するだけで異なるアルゴリズムを試すことができる．

ただし，ソルバの正しい動作が保証されるのは，離散関数の性質とアルゴリズムの組み合わせが正しい時（最小化しようとする関数がどの離散関数のクラスであるか正しく指定し，その関数クラスに対する最小化アルゴリズムを採用した時）のみであり，そうでなかった場合は結果が不正となるだけである．入力された最小化しようとする関数が，アルゴリズムの要求する M($M^\natural$) 性/L($L^\natural$) 性が満たされているかどうかを探索中に判定することは，残念ながら理論的にできない．

なお，我々のソルバには次のプログラムを利用している．

- 劣モジュラ関数最小化に，岩田による Iwata–Fleischer–Fujishige[2] の実装．
- 同じく，劣モジュラ関数最小化に，藤重・磯谷による minimum-norm-base アルゴリズムの実装[1]．
- 連続関数最小化に，J. Nocedal による quasi-Newton 法の実装である ‘L-BFGS’[⋆1] と，工藤による C++言語へのラッパー[⋆2]．
- 乱数を生成するために，斎藤・松本による ‘SIMD-oriented Fast Mersenne Twister’ [⋆3]．

4.2 在庫管理アプリケーション

このソルバを最適化エンジンとして組み込んだ在庫コスト最小化アプリケーションも開発して，オンラインソルバとして公開している[15]．アプリケーションでは，部品数 n をはじめとする各種パラメータ $p, c_j, \lambda_j (j = 1, \ldots, n)$ を対話的に入力し (図 1)，最適化を実行すると，最適な予備品の準備個数と，その際のコストが表示されるようになっている (図 2)．

ここで用いた予備品在庫管理モデルは，Miller[4] が提案した，需要量，発注量が離散値をとる（バックオーダーを考慮した多品種モデル）ものであり，航空機整備における部品管理等に用いられるモデルである．その在庫費用関数最小化のため，Miller[4] では「Miller の離散凸関数」を定義し，多項式時間ではない最小化アルゴリズムが構築された．後に，離散凸解析の研究において，この在庫費用関数が $L^\natural$ 凸関数であることが判明した[5]．

品種（部品）数が n からなる完成品に対して，バックオーダーを許し，完成品の品切れ時のバックオーダーによる罰金と，あらかじめ予備品（スペア）を購入しておくための代金からなるコストをできるだけ減らしたいという在庫モデルを考える．$c_j > 0$ を品種 j の単価とし，$x_j \in \mathbf{Z}$ を品種 j の発注量とする．$F_j(\cdot)$ は品種 j の需要に対する非負離散確率変数の累積分布関数で，$\varphi_j(m) \geq 0$ $(m \geq 0,\ 1 \leq j \leq n)$ を用いて

$$F_j(k) = \sum_{m=0}^{k} \varphi_j(m) \quad (k \in \mathbf{Z}_+)$$

と書き表されるものとする．このモデルでは $\lambda_j > 0$ で

$$\varphi_j(m) = e^{-\lambda_j} \frac{\lambda_j^m}{m!} \quad (m \in \mathbf{Z}_+)$$

である．次の関数 $f : \mathbf{Z}_+^n \to \mathbf{R}$ が最小化すべき目的関数である:

$$f(x) = p \sum_{k=0}^{\infty} \left(1 - \prod_{j=1}^{n} F_j(x_j + k) \right) + \sum_{j=1}^{n} c_j x_j. \tag{4}$$

ここで $p > 0$ は完成品 1 つに対してバックオーダーが生じた際の罰金である．完成品のバックオーダーの件数はバックオーダーが最大となる品種（部品）のバックオーダーの量に左右されることに注意されたい．(4) 式の最初の項は定常状態におけるバックオーダーが最大となる品種のバックオーダー量の期待値と，完成品単位個数あたりの罰金の積，すなわちバックオーダーによる罰金の期待値を表している．(4) 式の第二項は，スペアの購入コストを表している．二つの項を結合し，(4) 式は全体でバックオーダーによる罰金とスペアの購入代金からなるコストを表すことになる．

(4) 式は「Miller の離散凸関数」として導入されたが，上述のように，f は $L^\natural$ 凸関数である．このことから，$f(x)$ の最小化は $L^\natural$ 凸関数の最小化アルゴリズムにより，Miller[4] の方法より効率的に行えることがわかる．

ここでのオンラインソルバは，利用者が対話的にパラメータ入力・最適化を行い，瞬時に結果を得られる問題サイズに対して提供しているが，さらに大規模な在庫管理を必要とする場合は，開発ソルバ ODICON をダウンロードし，利用者のローカル環境で最適化されたい．

謝辞 本研究は科学研究費補助金の助成を受けたものである．

参 考 文 献

1) Fujishige, S. and Isotani, S.: A Submodular Function Minimization Algorithm Based on the Minimum-norm Base, *Pacific J. of Optim.*, to appear.
2) Iwata, S., Fleischer, L., and Fujishige, S.: A Combinatorial Strongly Polynomial Algorithm for Minimizing Submodular Functions, *J. ACM*, Vol. 48, pp. 761–777 (2001).
3) Kolmogorov, V. and Shioura, A.: New Algorithms for Convex Cost Tension Problem with Application to Computer Vision, *Discrete Optimization*, Vol.6, pp.378–393

⋆1 http://www.ece.northwestern.edu/~nocedal/lbfgs.html
⋆2 http://chasen.org/~taku/software/misc/lbfgs/
⋆3 http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT/

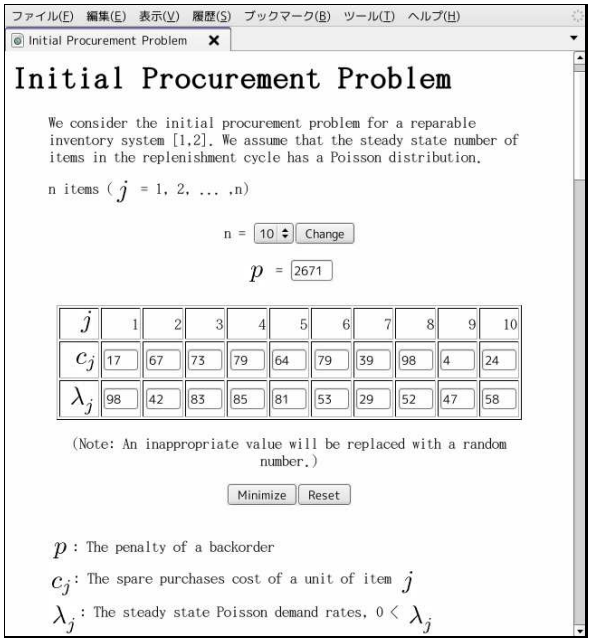

図 1 開発したオンラインソルバ (入力)
Fig. 1 Online solver (input)

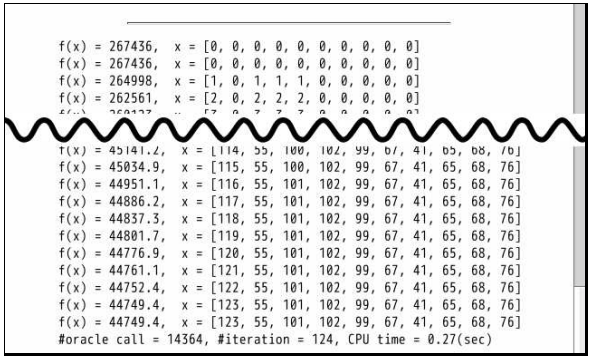

図 2 開発したオンラインソルバ (出力)
Fig. 2 Online solver (output)

(2009).
4) Miller, B.L.: On Minimizing Nonseparable Functions Defined on the Integers with an Inventory Application, *SIAM J. on Appl. Math.*, Vol.21, pp.166–185 (1971).
5) Moriguchi, S. and Murota, K.: Discrete Hessian Matrix for L-convex Functions, *Trans. IEICE*, Vol.E88, No.5, pp.1104–1108 (2005).
6) Moriguchi, S., Murota, K., and Shioura, A.: Scaling Algorithms for M-convex Function Minimization, *Trans. IEICE*, Vol.E85, No.5, pp.922–929 (2002).
7) Moriguchi, S., Shioura, A., and Tsuchimura, N.: M-convex Function Minimization by Continuous Relaxation Approach —Proximity Theorem and Algorithm, *Mathematical Engineering Technical Reports*, METR 2008-38, University of Tokyo (2008).
8) Moriguchi, S. and Tsuchimura, N.: Discrete L-convex function minimization based on continuous relaxation, *Pacific J. Optim.*, Vol.5, pp.227–236 (2009).
9) 室田 一雄：離散凸解析，共立出版 (2001).
10) Murota, K.: *Discrete Convex Analysis*, SIAM (2003).
11) Murota, K., and Shioura, A.: Fundamental Properties of M-convex and L-convex Functions in Continuous Variables, *Trans. IEICE*, Vol. E87, No. 5, pp. 1042–1052 (2004).
12) Schrijver, A.: A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time, *J. Comb. Theory, Ser. B*, Vol.80, pp.346–355 (2000).
13) 田村明久：離散凸解析とゲーム理論，朝倉書店 (2009).
14) 土村展之, ODICON: `http://www.misojiro.t.u-tokyo.ac.jp/~tutimura/odicon/`
15) 離散凸パラダイムの深化と拡大: `http://www.misojiro.t.u-tokyo.ac.jp/DCP/`