

数式による記述で数理計画問題を解く試み

01701240 (株) 数理システム *高橋 良徳 TAKAHASHI Yoshinori
(株) 数理システム 山下 浩 YAMASHITA Hiroshi

1. はじめに

コンピュータを利用して数理計画問題を数値的に解く際に、どのようにして問題自身を定義するか、すなわち利用するソフトウェアにどのようにして解くべき問題を教えるかというテーマは、アルゴリズムの研究やソフトウェアの開発と並んで常に研究されるべき重要な問題である。この周辺の技術の発展無くしては数理計画法の普及も無い、と言っても過言ではないであろう。本発表は、この問題に対する一つの試みを議論する。

線形計画問題に対する係数データを与えるデータ形式として、古くからいわゆる MPS ファイル形式がデファクト・スタンダードとして利用されてきたことはよく知られている。この形式は行列の非零要素を並べただけの極めて原始的なものであった。このデータから元の問題を把握するのは(特殊な人を除いて)ほとんど不可能であろう。また、当然のことながら非線形問題を定義・記述することは出来ない。

一方、より広い範囲の問題を柔軟に記述するために1980年代の初頭から「数理計画のためのモデリング言語」の研究がなされるようになった。代表的なものとして GAMS, AMPL などが知られている。これらは、問題の数式による記述をなるべく忠実に反映するための言語である。興味深いことに、それぞれがそれらが提案された時代に普及していたプログラミング言語の色彩を強く持っている。すなわち、GAMS は FORTRAN, AMPL は C のフレーバーが強い。

我々のグループは汎用数理計画ソフトウェア NUOPT の開発と並行して、モデリング言語 SIMPLE[1, 2] を提案し開発してきた。非線形問題も含むより広い範囲の問題を対象にするためには高速自動微分法を利用する必要があり、問題の計算グラフによる表現を獲得するために、演算子のオーバーローディングの機構のあるプログラミング言語上での実現というアプローチを取った。そのようなプログラミング言語として C++ を利用している。(S 言語上での実装も完成している。)

我々は、SIMPLE の設計方針としていくつかの指針を立てたが、その中の一つとして「問題の数式表現なるべく忠実に、プログラミング的にならないように」という方針がある。この点は、上記のモデリング言語など

とは大きく異なる点であると考えられる。この性質とその他の様々な機能によって、複雑かつ巨大な数理計画問題が多くの NUOPT のユーザーによってモデリングされ解かれている。

本発表では、さらに数理計画のモデリングを容易にするために、数式を直接入力として数理計画問題を解くことを考える。すなわち、教科書などに記述されている数理計画モデルがほぼそのままソフトウェアへの入力として利用でき、答えを得ることが最終目的である。このような技術が実用化されれば、教育等の現場で学生に数理計画モデルを理解させるために余計な労力を使うことなく教育出来ることなどが期待される。また、研究や産業の現場で様々な数理計画問題を解く際に学習のためのバリエーションが低くなり、数理計画の技術の普及と応用が促進されることが期待される。

2. 数式記述の例

自由な数式記述と言っても、コンピュータへの入力であるから必然的にある種のルールが必要になる。しかし、細かいルールを記述することは止めて、本システムで可能な数理計画モデルの例をいくつかあげることによって、本システムのモデル記述機能の概要を示す。(以下では、キーワードは英語で入力しているが、日本語も可能とする。)

● Hock and Schittkowski No.105

index	i, j
variable	x_j
parameter	$y_i, \pi = 6 \arctan(\frac{1}{\sqrt{3}})$
Expression	a_i, b_i, c_i $a_i := \frac{x_1}{x_6} \exp(-(y_i - x_3)^2 / (2x_6^2))$ $b_i := \frac{x_2}{x_7} \exp(-(y_i - x_4)^2 / (2x_7^2))$ $c_i := \frac{(1-x_2-x_1)}{x_8} \exp(-(y_i - x_5)^2 / (2x_8^2))$
minimize	$-\sum_{i=1}^{235} \log((a_i + b_i + c_i) / \sqrt{2\pi})$
subject to	$1 - x_1 - x_2 \geq 0$ $0.001 \leq x_j \leq 0.499, j \in \{1, 2\}$ $100 \leq x_3 \leq 180$ $130 \leq x_4 \leq 210$ $170 \leq x_5 \leq 240$ $5 \leq x_j \leq 25, j \in \{6, 7, 8\}$

●ポートフォリオ最適化

index i, j
 parameter $N, r_i, \sigma_{ij}, \rho$
 variable x_i
 minimize $risk = \sum_{i=1}^N \sum_{j=1}^N \sigma_{ij} x_i x_j$
 subject to $\sum_{i=1}^N r_i x_i = \rho$
 $\sum_{i=1}^N x_i = 1$
 $x_i \geq 0, i \in \{1 \dots N\}$

●最小コストネットワークフロー問題

Graph $G = (N, A)$
 element $i \in N, e \in A,$
 $e_{out} \in out(G, i), e_{in} \in in(G, i)$
 parameter a_e, s_i
 variable x_e
 minimize $cost = \sum_e a_e x_e$
 subject to $\sum_{e_{out}} x_{e_{out}} - \sum_{e_{in}} x_{e_{in}} = s_i$
 $x_e \geq 0$

3. 実装のアウトライン

数式をコンピュータに入力するためのソフトウェア技術は既に実用化されているものがいくつか存在する。我々は、新規にそのようなものを開発する意味は特に無いと考え、既に存在しているものを利用することにした。そのようなものとして最も将来性があると思われるのは MathML であるが、現在のところ、この利用は時期尚早と考え、今後の発展・成熟を待つことにした。そこで、当面は Design Science 社の提供しているソフトウェア MathType を数式入力手段として利用する。MathType は GUI によって数式を入力することが可能で、その数式を TeX 形式で保存することを可能にする。そこで、我々は得られた TeX 形式から上述の SIMPLE 言語による記述に変換するソフトウェアを開発した。その情報は NUOPT に直接入力可能で、最適解を得た後、必要であれば最適解の情報を数式によって表示するという方針である。

MathType に数式を入力する手段は色々考えられるが、今回は Microsoft Word 経由の手段を採用した。したがって、問題記述の前後に Word 文書として説明を入れたり、最適解の情報を数式として出力したり、Excel などとの連携により表やグラフを表示したりすることが可能になる。

これを実現させるために、特別な Word テンプレートを用意し、Word を起動する際に通常読み込まれる標準テンプレートの代わりに、この特別なワードテンプレートが読み込まれるように設定しておく。この特別なテン

プレートファイル (tex2simple.dot) は VB マクロで記述されており、MathType の数式 OLE を TeX 形式に変換する機能、TeX 形式の文字情報を字句・構文解析して SIMPLE のコードに変換する機能などを順次呼び出す。

4. 問題点と今後の機能拡張

(i) 文書整形機能を主とする TeX 形式を数理計画モデル記述言語の SIMPLE の形式に変換する部分に色々な問題点が生じる事は明らかである。TeX 形式には数式情報として意味を持たないものがある。その代表がフォント情報である。不要なフォント情報を取り除きプレーンな TeX 形式に変換するモジュールを別に用意した。例えば、次の記述は全く同じ意味を持たなくてはならない。

```
{\rm{Variable}}\,x,y
Variable x,y
```

(ii) 最も大きな問題点是我々が通常使用している数式表現の曖昧さである。例えば、乗算記号の省略である。我々の開発したソフトウェアでは乗算記号の省略を許可している。しかし、もちろん SIMPLE モデル (C++ プログラム) では乗算記号は明確に書いてある必要がある。したがって、乗算記号が省略されていた場合には乗算記号を補う必要性が出てくる。たとえば、 abc という表式が現れたとき、 abc が既に宣言されていれば一つのオブジェクト (たとえば変数) として扱われる。そうでなければ、これは $a \cdot b \cdot c$ (a, b, c の積) として解釈される。 $a \cdot bc$ という表式が現れたときは、 bc の (積への) 分解は試みない。(常識的に不自然であるから。)

同様のことが変数やパラメータの添え字にも言える。すなわち、 x_{ijk} という表式が現れたときは、 ijk というオブジェクトが宣言されていれば一つの添え字、そうでなければ $x_{i,j,k}$ と解釈する。 $x_{i,j,k}$ はそのまま i と jk の二つの添え字付きと解釈される。

以下のような曖昧性も存在する。 $\sin xy$ は $\sin(x \cdot y)$ あるいは $\sin(xy)$ であると判断されるであろうが、 $\sin x \cos y$ は $(\sin x) \cdot (\cos y)$ と解釈するのが普通である。今回開発したバージョンでは簡便のために関数の引数には括弧をつけてもらうことにした。

我々が通常使用している常識的な (しかし曖昧性を伴った) 数式表現を可能な限りストレートに使用できるようにするのが今後の課題であろう。

参考文献

[1] 山下, 田辺, 逸見 "数理計画のためのモデリング言語 SIMPLE I 概要" OR 学会春季研究発表会 1997年.
 [2] (株) 数理システム "SIMPLE マニュアル".