7A−9

制約プログラミングのBundle Price問題への応用*

赤坂 克也　大和田 勇人　溝口 文雄†

東京理科大学　理工学部†

1 はじめに

制約充足問題は，オペレーションズリサーチ，LPなどでも重要な領域である．

Constrained Logic Problem(CLP) は，このような問題を解決するためにPrologなどで次のように拡張されている．

- 言語に $X+Y<5$ のような制約を含む．
- 制約が相互に充足されるかの決定．

これらの特徴を持ったシステムとして代表的な物は，CHIP，CLP(ℜ) がある．この2つのシステムでは，実数で表現された制約を扱うことが可能になるが，直接的に最適化を行う機構は存在しない [2]．

本研究では，最適化を実際に行う機構を構築し，Mathematical Programmingの分野に適応させる事を試みる．混合整数計画問題 (mixed integer linear problem) の一つである，Bundle Price問題を対象とする．

Bundle Pricingの問題の特徴は選言的なプログラムによってのみ，混合整数計画問題として定式化することである．このことは，制約プログラミングの有効な利用を示すことが可能となる．

Bundle Price問題の研究が進むにしたがって，いくつか問題が持ち上がってきている．それは次のようなものである．

- 混合0/1整数計画問題は，比較的小さな問題領域でも計算時間が膨大になってしまう．
- モデル化が補助変数が多くなりすぎて複雑になってしまう．

これらの問題は，各顧客セグメントについて分散して最適化された結果を処理して，顧客全体の最適化する機構を設計すれば良い．

本研究では，このような経緯から，分散処理を行なった結果を制約プログラミングを利用してBundle Price問題での顧客全体に対しての各Bundleの価格を最適化するシステムを実現することを目的としている．

2 Bundle Price問題

Bundle Pricing問題は，いくつかの構成要素の組み合わせで顧客の欲求を満たす商品，またはサービスについて，価格を最適化する事である [1]．

ソフトウェア会社が制作したいくつかのモジュールを組み合わせでパッケージを売るときの価格をどの様に設定するかという問題があるとする．このとき，異なる顧客のセグメントは各モジュールに対してそれぞれ異なる評価をしている．

ある顧客は全ての商品の組合せを購入したいと考えている一方で，他の顧客は特定の商品にしか関心を示さないことが考えられる．また，商品のコストはそれぞれかなり大きく異なっている．

このような状況では，生産者は顧客の購入意欲を保たせるような，(Bundleについては，割安感が強くなるように) 個々の商品の価格設定と，それらの組合せで構成されるBundleの価格設定を行なわなければならない．この問題がBundle Price問題である．

3 本システムの特徴

本研究のシステムの特徴を挙げると次のようになる．

1. *MathLink* を用いた *Mathematica* とC言語のリンク
2. socket通信を用いた分散環境
3. 選言的な制約式を制約プログラミングで表現し，制約を充足させる

システムの構成は以下のようになる．本研究のシステムは，

server部の処理 socket通信から獲得したデータを各serverで部分最適化

client部の処理 各serverの結果を通信でclientが制約充足させる

という過程に大別する事ができる．

本システムの開発環境は図1となっている．使用計算機はSun4/1(OSはSun-OS 4.0.3)，開発言語は *Mathematica*，その外部環境である *MathLink*，C言語である．

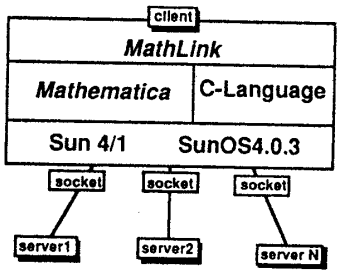

図1: 計算機環境

*Application of Constraint Logic Programming to Optimal bundle pricing

†Katsuya AKASAKA, Hayato OHWADA, Fumio MIZOGUCHI

‡Faculty of Sci. and Tech., Science Univ. of Tokyo

3.1 socket ベースの接続

先のモデルの最適化を各 Bundle ごとに適応させ，複数の計算機にその計算を分散し，計算を行う．その結果を「商品原価より，reservation price が低い事はありえない」などの制約でまとめれば，上記で述べた，選言的な制約を満たす最適な Bundle Price が求められる．

現在，socket による分散計算を行なうために *Mathematica* の組み込み関数 *LinearProgramming*[] を *MathLink* により socket を介して分散計算する事を行なっている．

3.2 *LinearPrograming*[] の *MathLink* による分散

Mathematica の組み込み関数 *LinearPrograminng*[] を，*MathLink* を用いて socket で通信可能とすることを考える．

LinearPrograminng$[c, m, b]$ $\cdots$ $m.x > b$ および $x > 0$ を制約条件として $c.x$ を最小化するベクトル x を検索する関数

4 実行例

4.1 単純な実行例

ここで，LinearProgramming を socket で通信したときの実行例を示す．ここでは，目的関数 $2x-3y$，制約条件 $x+y > 10, x-y > 2, x > 1$ の時の目的関数の最小化で，分散計算が具体的におこなわれる例を取り上げる．

1. client 側の入力

```
sutna029\%linear.sockets
Enter socket name: 1113@sutna029
```

まず，プログラムを起動し，プロセス通信の socket を決定する．データ入力終了後は次のように *Mathematica* の分散された server が部分最適化をする．

2. sever 側の *Mathematica*，*MathLink* での処理

```
In[1]:=link=LinkOpen[]  (Mathematica とのリンク開始)
Out[1]=LinkObject[1113@sutna029,1,1]
In[2]:=LinkReadHeld[link]
Out[2]=Hold[LinearProgramming[
        List[2,-3],List[List[-1,1],
        List[1,-1],List[1,0] ],
        List[-10,2,1]
In[3]:= ReleaseHold[%]
Out[3]={6,4}
```

Mathematica とのリンクの開始の宣言を行い，次に socket の選択によって，タスクが決定される．そして，LinkReadHeld[] で *Mathematica* とリンクしながらのタスクの実行が行われる．結果はリスト構造で Hold されたものが与えられ，これをリリースして，最適値の結果が得られた．

4.2 Bundle Price のデータによる実行例

以下のように [1] のデータに reservation price を変え，セグメントの大きさを n 以外にしたものについて実験を行った．表 1からもわかるように，reservation price についても

Reservation Price								
customer	segment size	A	B	AB	C	AC	BC	ABC
segment 1	$2N$	40	70	105	15	55	90	130
segment 2	$2N$	65	60	120	35	90	90	155
segment 3	N	40	70	110	70	105	135	185
segment 4	$2N$	65	60	120	70	125	125	200
M.C		70	70	110	80	150	145	175

表 1: Bundling and Cost

劣加法的なデータで実験を行った．reservation price でも劣加法性が成立する事は以前に示されている（直感的には，消費者は Bundle 組み合わせに割安感を期待している事で示される）．結果は，Bundle B,Bundle AB,Bundle ABC が利益を産む Bundle 組合せになる．その時の価格は Bundle B が 70，Bundle AB が 125，Bundle ABC が 200 である．利益は $30N$ となる．

計算時間は表 2のようになった．CPU Time の測定には *Mathematica* の CPU Time 測定の関数 Timing[] を使用した．

以上のデータでの CPU Time の結果をこの節では示す．

CPU Time(sec)	
socket 使用	81.25–97.58
socket 未使用	112.54–125.68

表 2: CPU Time の比較

表 2の結果は，*Mathematica* 上のみの CPU Time である．socket を使う場合には，データの出入力，通信にかかる時間が *Mathematica* 上での計算時間の他に（10–20 秒は，かかる）がある．構成要素の数が n 個の時、組み合わせはは 2^n-1 となり，大きな問題空間になるに従って計算時間の短縮の効果が得られた．

5 おわりに

本研究は，従来の制約充足問題の概念と分散計算機構を用いて，混合整数計画問題の最適化可能に拡張を図った．実世界問題の Bundle Price 問題を対象にして，実験を行い，より変数が多い（探索領域の大きな）領域においても最適化可能な機構を提案した．

参考文献

[1] Ward Hanson and Kipp Martin. Optimal Bundle Pricing. *MANAGEMENT SCIENCE*.feb,1990.

[2] Pierre Lim,Micheal.L.Epstein. A Constraint Logic Programming Shell. *Proceedings of the 1990 Workshop on Programming Language Programming*.1990.