大域的最適化問題に対する区間法の並列計算

1U－9

藤井康雄　市田浩三
（京都大学 情報処理教育センター）（京都産業大学 経営学部）

1．はじめに

これまでに我々は、区間法（区間解析）を利用して、多目的関数の最適化ならびに多峰性多変数の大域的最適解を求めるアルゴリズムを検討してきた[1,2]。ここでは、さらに多くの変数を持つ目的関数の最適化を短時間で評価するために、並列計算機上で区間解析を用い大域的な最大値を探索するアルゴリズムを検討した結果について報告する。プログラムの開発の効率化のために並列計算機は、はじめにPVM(ParallelVirtual Machin)とし、FDDI ネットワーク上の複数台のワークステーションを使い検討を行った。さらに 8-CPU の並列計算機上に移植し、アルゴリズムの検討を数値実験を通して実施する。

2．区間解析を用いた関数の最大値探索

これまで区間解析を利用した大域的最適化では、多峰性多変数関数 $f(x_1, x_2, ..., x_k)$ の領域 $a_i \leq x_i \leq b_i, (i = 1, 2, ..., k)$ における最大値探索を概略以下のステップで実現している。

ステップ１：各変数の定義域を区間とし、初期値として与える。

ステップ２：変数定義域を分割し、各部分領域で中点による関数値 f_a, f_b を求める

ステップ３：f_a の中点の値より f_b の上限が小さいならば、その領域に最大値を含まないので消去。

ステップ４：残存領域から、$|J| = detJ(\boldsymbol{x})$ を求める。

ステップ５：$0 \in |J|$から、$0 \notin |J|$ なら、区間に拡張した Newton 法を適用し、指定した範囲に収束すれば解として保存。そうでない場合は、解が領域外に存在することを示すので、解を含まない領域とみなして消去する。

ステップ６：$0 \in |J|$ の場合、$\mathbf{o} \in \boldsymbol{f}(\boldsymbol{x})$ならば解を含む可能性があるので保存する。一方、$\mathbf{o} \notin \boldsymbol{f}(\boldsymbol{x})$ ならば解を含まないため領域を消去する。

ステップ７：残存領域から関数値の上限の最大の部分領域の変数定義域を分割し、ステップ２へ戻り繰り返す。

ここで、ステップ４～６の区間演算に拡張した Newton 法は、扱う目的関数が連続微分可能であれば適用できるが、そうでない場合には、残りの変数定義域の分割を行うステップのみを用いることになる。また制約条件の付いた目的関数では、ラグランジュ乗数法を導入して、その最大値を求める問題として解いてきた。

3．PVM での並列処理形態[3]

PVM では１台のマスターと複数だいのスレーブで構成するが、マスターはスレーブと同様な演算処

An Interval Arithmetic Method of Parallel Computing for Global Optimization.

Yasuo FUJII *, Kozo ICHIDA**.
*)Educational Center for Information Processing, Kyoto University, Kyoto 606, Japan.
**)Faculty of Business Administration, Kyoto Sangyo University, Kamigamo, Kyoto 603, Japan.

理も分担させる。マスターからスレーブへのタスクの生成は、マスターの pvm 関数の実行によって行い、変数定義域である初期値の配布も同様な関数によって行う。また各部分領域の関数値の比較は、各スレーブのタスク間の通信を利用して行っている。ここで、各スレーブはネットワーク上の異なったワークステーションが利用でき汎用性が高いが、我々はマスター・スレーブとも同一構成のワークステーションを利用した。アルゴリズムの検討には、

$$f(x_1,x_2) = -2x_1^2 + 1.05x_1^4 - (1/4)x_1^6 - x_1x_2 - x_2^2 \qquad (1)$$

の 3-Hump Camel-Back 関数の最大値を、 $-5 \le x_1 \le 5, -4 \le x_2 \le 4$ の定義域で探索した。

この定義域内で最大値は、 $x_1 = x_2 = 0.0, f(\boldsymbol{x}) = 0.0$ であり、

$x_1 = 1.7475523..., x_2 = -0.8737761..., f(\boldsymbol{x}) = -0.2986384...$

の極大値が原点対象に存在している[4]。

4．並列処理での考慮事項

PVM は、並列計算機でも利用が可能なので、ここまでの検討したプログラムが利用できるので、当面、並列計算機上でも利用できる。一方並列計算機上で PVM を利用しない場合、区間演算アルゴリズムにおいても、一般の数値計算アルゴリズムでの並列処理と同じように、以下のようなハードに依存した注意が必要である。

(1)キャッシュメモリの構成からくる演算実行性能の低下を避ける。これはキャッシュ効果に依存しており、演算パイプラインに十分な速度でデータを供給できるように配慮する。

(2)キャッシュとメインメモリ間のデータ転送は、ダイレクトマッピングを考慮して、プログラム中の配列の寸法を確保しなければならない。

(3)区間に拡張した Newton 法を利用するため、行列演算が必須となるが、ループアンローリングを行い、ソフトウェアパイプラインを有効に働かせる。

5．おわりに

我々はこれまでに、関数の最適化問題に区間解析の手法を用いて高精度の最適解を得てきたが、変数が増加すると関数評価に要する時間が掛かり困難となってきていた。したがって並列計算機を利用した利用するのは一つの解決方と考えている。ここでは、並列アルゴリズムの検討を開始したばかりであり、区間演算むきの最適なアルゴリズムの一方法を検討したにすぎない。したがって、さらに並列アルゴリズムの検討と、多変数の場合についての数値実験を、より多数の CPU を持つ並列計算機上で継続して行おうと考えている。

参考文献

[1]K. Ichida and Y. Fujii:Multicriterion Optimization Using Interval Analysis, Computing,44,47-57(1990).

[2]Y. Fujii and K. Ichida:Maximization of multivariable functions using Interval analysis, Interval Mathematics 1985(K.Nickel,ed), Springer-Verlag Lecuture Notes in Computer Sci.212 ,17-26(1986).

[3]A.Geist, J. Dongarra (村田英明訳):PVM3 User's guide and reference manual, 1995.2.

[4]藤井、市田:分散環境を活用した区間演算による最適化,電気関係関西支部連合大会論文 G40(1994).