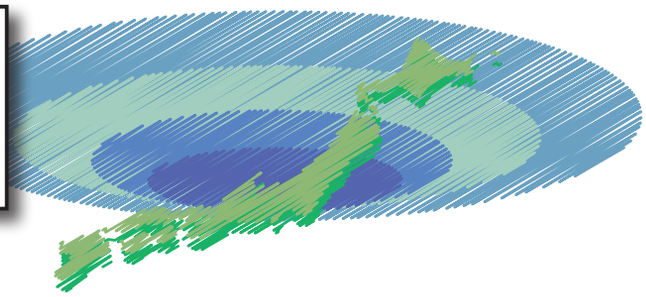
解説

大規模最適化問題への挑戦
―クラスタ&グリッド計算の適用例について―

藤澤 克樹

東京電機大学理工学部数理科学科／
独立行政法人 産業技術総合研究所 グリッド研究センター（兼任）
fujisawa@rnc.r.dendai.ac.jp



最適化問題は非常に広い応用範囲を持っているが，実用的なレベルでは問題サイズが大きくなり，必要な計算量も問題サイズに対して指数的に増加していくのでアルゴリズムの改良だけでなく，大規模な計算設備での並列計算も必要になる．本稿では大規模計算問題として最適化問題を取り上げ，組合せ最適化問題や数理計画問題などに対する最新の並列計算（グリッドやクラスタ計算なども含む）の手法とその成果，また具体的な事例について解説を行う．

最適化問題

近年の計算機や通信技術の急速な進歩によって，工学や理学などの多くの分野で大規模な数値計算が行われていることはよく知られている．最適化問題の分野においても，この数年で想定する問題の規模，適用する手法，実問題への適用可能性などの考え方に大きな変化が生じている．後で解説するように理論的な研究成果は大規模な最適化問題に対する並列計算の適用に対して否定的なものが多い．もちろん理論的な研究による予測を最近の計算機技術が覆したというわけではないが，アルゴリズムの発展や並列計算の適用によって目覚しい成果が達成されている．

次に最適化問題の説明を行う．最適化とは複数の選択肢から最善のものを選ぶことであり，数学的に表現すると与えられた制約条件を満たし，目的関数$f(x)$の値が最小または最大になるような決定変数xの値を見つける問題である．一般的には次のように記述する．

目的関数：$f(x) \rightarrow$ 最小(最大)

制約条件：$x \in S$

この場合Sは変数xが取ることが可能な値の集合であり，実行可能領域と呼ばれる．本稿で扱う最適化問題は数理計画問題と組合せ最適化問題である．代表的な組合せ最適化問題には，巡回セールスマン問題や二次割当問題などがあるが，計算量の理論ではこれらの問題は$\mathcal{NP}$-困難と呼ばれるクラスに属している．$\mathcal{NP}$-困難な問題は問題の大きさnの多項式時間では最適に解けないと考えられている．つまり問題を解くための計算量が$\mathcal{O}(n^2)$や$\mathcal{O}(n \log n)$のようなnの多項式オーダーではなく$\mathcal{O}(2^n)$や$\mathcal{O}(n!)$のようなnの指数オーダーになると考えられている．n台の計算機で並列計算を行っても高々n倍しか高速化されないのに対して，計算量が$\mathcal{O}(2^n)$の問題では，問題の規模が10倍になれば計算量が2^nに比例して1,000台以上(2^{10}=1024)の計算機が必要になる．この性質のために従来より並列計算の適用に否定的な意見も多かったのも事実である．しかしすでに述べたように近年の計算機と通信技術の進歩が追い風になって，主に以下の3つの方法やそれらの方法の組合せによって組合せ最適化問題を解く研究がさかんに行われている．

1．超高速，大容量の計算機を集めて従来より提案されている手法に並列計算の技術を適用して問題を解く．
2．アルゴリズムを改良して，計算時間を減らす工夫を行う．ただし$\mathcal{NP}$-困難な問題である以上，最悪の場合には計算時間はnの指数オーダーになる．
3．最適解を得ることをあきらめて，近似解法を適用して実用的な時間で優れた解を得ることを目指す．

一方，数理計画問題を解くためのソフトウェアもアルゴリズムの改善や計算機能力の向上によって大幅に性能を向上させている．数理計画問題は問題の大きさnの多項式時間で解けるものも多いが，想定される数理計画問題が巨大であるときには単体のみの性能向上だけでは短時間に問題を解くことは困難である．それらの困難を克服するための並列計算の手法としてクラスタ計算（Cluster Computing）とグリッド計算（Grid Computing）が最近注目を集めている．次章以降では，組合せ最適化問題や数理計画問題になどに対する最新の並列計算（グリッドやクラスタ計算なども含む）の手法とその成果，また具体的な事例について解説を行う．

最適化問題に対するクラスタ＆グリッド計算

クラスタ計算に関する研究はPCやワークステーションなどの高性能化，低価格化に伴って1990年代半ばより開始された．特に比較的に安価なPCで構成されたクラスタ計算機はPCクラスタと呼ばれており，現在のクラスタ計算機の主流になっている．クラスタ計算機は複数の独立した計算機を高速LANで結合し，単一システムのイメージを提供する並列計算技術である．特に一般のPC技術を活用するPCクラスタ（**図-1**）では，近年のPCの高性能化と低価格化によって，従来のスーパーコンピュータの数十倍から数百倍のコストパフォーマンスを達成している．

クラスタ計算に関する技術は主にハードウェアとソフトウェアの部分に分けることができる．PCクラスタでは使用するPCはCPUやメモリなどの直接数値計算にかかわる部分に重点的に投資されている．またネットワークカードやスイッチはGigabit EthernetやMyrinetやInfinibandなどの高価で高性能の部品が使用されることが多い．逆にビデオカードや外部入出力装置は必要最小限の部品で済まされることが多い（ただし最近のビデオカード自体の計算能力は非常に高い）．またソフトウェアでは並列計算を実現するためのソフトウェアMPI[☆1]やOpenMP[☆2]などが有名である．またPCクラスタの全体性能を上げるために直接ハードウェアの制御を行ったり，効率のよいプロセスのスケジューリングを行うSCore[☆3]が使用されている．高速な計算機の上位500台を集めたWebサイトTOP500[☆4]によると，近年スーパーコンピュータだけでなく，クラスタ計算機が上位に多くランキングしている．

図-1　PCクラスタ

次にグリッド計算を実現するためのソフトウェアについて説明を行う．グリッド計算システムでは，すべての計算機がLAN等で接続されている場合だけでなく，遠隔地のクラスタ計算機同士を高速なネットワーク（インターネット等）で接続して，お互いの計算機資源（特にCPUパワー）を有効に活用し，大規模な問題を効率よく解くことが主要な目的になっている．グリッド計算システムはNinf[☆5]（Ninf-1 & Ninf-G[5]），NetSolve[☆6]，Condor[☆7]などが有名である．たとえばNinfには以下のような特徴がある．

1．通常の関数に似たGridRPCのインタフェースを持っているので，簡単な関数等の記述でグリッド上でClient-Serverモデルを実現することが可能である．
2．インターネット等によって広域に接続，提供されているハードウェア，ソフトウェアの利用が可能である．
3．多様な言語（C, C++, Fortran, Javaなど）を開発に用いることが可能である．

またNinfでは，同期，非同期の呼出しがサポートされており，非同期の場合では効率よく大量のプロセスの並列動作を行うことができる．またNinf-G[5]は，産業技術総合研究所で開発されているGridRPCシステムで，従来のNinf（Ninf-1）をGlobus Toolkit[☆8]を用いて再構築し

☆1 http://www-unix.mcs.anl.gov/mpi/
☆2 http://www.openmp.org/
☆3 http://www.pccluster.org/
☆4 http://www.top500.org/
☆5 http://ninf.apgrid.org/
☆6 http://icl.cs.utk.edu/netsolve/
☆7 http://www.cs.wisc.edu/condor/
☆8 http://www.globus.org/

たものである．詳しくは情報処理2003年6月号「グリッドコンピューティングの技術動向」を参照されたい．

図-2は非凸二次計画問題に対する逐次凸緩和法[3]（凸計画問題である半正定値計画問題や線形計画問題を子問題として非凸計画問題を解く方法）や多項式方程式に対するホモトピー法[4]（多項式方程式系のすべての孤立解を求めるための方法）を実行するために，Ninf（Ninf-1）を用いて実現されたClient-Server systemである．これらの問題では，いきなり元問題（親問題）を解くのは難しいので，一部の変数の値を固定するなどして複数の子問題を作成して，これらを代わりに解く方法が用いられる．この場合Ninf Clientは1台のPCであってNinf Clientから 非同期にNinf Server（PCクラスタ）上の複数のPCが呼び出される．そしてNinf Server上では同時に多くの子問題が生成されて解かれることになる．この場合Ninf Serverは複数のPCクラスタで構成されていてもよく，また複数のPCクラスタは，それぞれ遠隔地に設置されていてもよい．その場合は接続されているインターネットなどのネットワークの転送速度が重要になってくる．

またCondorは，アイドル状態の計算機を有効活用することを目的としたジョブスケジューリングシステムである．計算資源はCondorが管理するので，ユーザは実際にどの計算機を使用するのか意識する必要がない．さらにCondorはチェックポイントとマイグレーション機能をサポートしているので計算途中のジョブの停止，他の計算機への移動を行うことが可能であり，耐故障性に優れている．ただし単一のジョブを実行するためのシステムであるので，ジョブの自動並列などは行わない．

大規模最適化問題に対する適用例

■巡回セールス問題に対する分枝カット法

巡回セールスマン問題（Traveling Salesman Problem: TSP）は，最も有名な$\mathcal{NP}$-困難の組合せ最適化問題である．n個の点を一度ずつ訪れて元に戻ってくる巡回路（tour）を求める問題であり，目的関数は移動距離（tourの長さ）の最小化である．対称TSP（逆の順番に回っても同じtourとみなす）では巡回路の総数は$\frac{(n-1)!}{2}$であるのでtourを1つずつ列挙していくと30点ぐらいの小さな問題でも最適解を求めることは実質的に不可能である．そのため最適解を求めるためにさまざまな方法が提案されているが，現在まで最も成果を上げているのは，切除平面法（cutting-plane method）[2]である．TSPは整数計画問題として定義できるが，切除平面法ではいったんTSP（整数計画問題）を線形計画問題（LP）に緩和する．次に変数が整数条件を満たすように有効な1次不等式

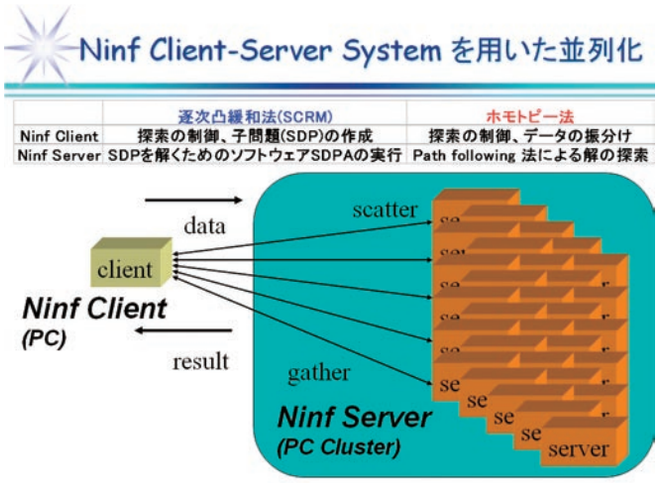


図-2　Ninf Client-Server system

（妥当不等式と呼ばれる）を加えていく．どのような妥当不等式を用いるか，また変数の数が膨大になる線形計画問題をどう解くかなどのさまざまな工夫が提案されている．詳しくは文献2）を参照のこと．実際に大規模なTSPを解くときには，Concorde[☆9]と呼ばれるソフトウェアを用いている．Concordeには分枝限定法に切除平面法を組み込んだ分枝カット法と呼ばれるアルゴリズムが実装されている．なおConcordeはTSP以外のネットワーク最適化問題を解くことも可能であり，学術的な利用に限ればソースコードを入手することも可能である．Concordeは以下のような特徴を持っている．

1．切除平面法のための妥当不等式を生成する．また各変数の値を特定するための分枝限定法の枠組みを持っている．
2．LPを解くために商用のLPソルバー CPLEXの使用が可能．
3．分枝限定法の上界を計算するためにLin-Kernighan法や2-opt, 3-optなどの近似解法を含む．

現在（2003年12月），最適解が判明している最も点数の大きなTSPは点数が15,112個のd15112と呼ばれている問題であり，TSPLIBというTSPのベンチマーク集に含まれている．**図-3**はd15112の最適解（最適順回路）[☆10]である．この問題の場合ではConcordeを用いて米国のPrinceton大学やRice大学の合計110個のプロセッサを用いて最適解が得られている．また，実行時間は1プロセッサ（Alpha 21264 500MHz）で換算した場合22.6年分に相当する．**表-1**はd15112を解くために用いられた計算機資源を表している．このConcordeによる並列計算は，マスタ＆ワーカモデルで実現されており，通信

☆9 http://www.math.princeton.edu/tsp/concorde.html
☆10 http://www.math.princeton.edu/tsp/

図-3　d15112の最適解

計算機	CPUの種類	CPU数
1	Alpha 21264 500MHz	5
2	Alpha 21264 500MHz	6
3	Alpha 21264 500MHz	32
4	Alpha 21264 667MHz	12
5	Athlon 800MHz	24
6	Pentium III 450MHz から 1GHZ	19
7	Alpha 21164 400MHz	12

表-1　d15112を解くために用いられた計算機資源

実行時間	6:22:04:31
平均使用計算機数	653
最大使用計算機数	1007
総 CPU 時間	約11年
分枝木の点数	11,892,208,412
LP 緩和問題数	574,254,156,532
並列化効率	92%

表-2　nug30の実行結果

はNSFソケットを用いている．マスタは3番の計算機の1プロセッサに割り当てられていて，ワーカの仕事を行うのは3番の残りと1, 2, 4, 5番の計算機である．マスタは分枝限定法の分枝木を保持し，ワーカには分枝した子問題が割り当てられる．残りの6と7番の計算機には切除平面法の過程で用いられる膨大な1次不等式が保存されている．実際には膨大な変数を持つLPをメモリ上に保持して計算することは大変困難であり，またLPを解く時間が全体の中で大きな割合を占めている．なお現在は1,904,711 点という非常に巨大なTSPに対する計算が行われている．

■Condorを用いた二次割当問題に対する分枝限定法

二次割当問題（Quadratic Assignment Problem: QAP）も代表的な$\mathcal{NP}$-困難の組合せ最適化問題である．その用途は，工場内の機械のレイアウトや外部記憶装置上でのデータ配置など多岐に及んでいる．QAPは$V=\{1,\ldots,n\}$と$n\times n$の対称行列$F=(f_{ij})$と$D=(d_{k\ell})$が与えられたとき，次のような費用関数Cを最小化するような順列$\pi: V\to V$を求める問題である．

$$C(\pi)=\sum_i\sum_j f_{ij}d_{\pi(i)\pi(j)}$$

この問題は次のように解釈できる．求める順列πはn個の対象物のnカ所への配置を示していてf_{ij}は対象物iとj間のフロー量（たとえば建物iとj間の移動人数など），$d_{k\ell}$は場所kとℓの間の距離を示している．

QAPもTSPと同じく目的関数を最小にする順列を求める問題であるが，最適に解くのは大変難しく，現在解かれている問題の大きさは$n=30$程度である[1]．QAPを最適に解く場合も分枝限定法を用いるが，分枝数が大変多いので大量の計算機資源が必要である．しかし分枝限定法は工夫によって通信量を少なくすることも可能であり，分枝木を用いた解の探索は複数の個所で並列に行うこともできる（ただし定期的に情報を集約する必要がある）．そのためグリッド計算に向いた手法であり，実際にCondorを用いてグリッド上で大規模な計算が行われている[1]．**表-2**はnug30というQAPLIBというベンチマーク問題集に含まれている大きさ30の問題を解いたときの結果である．実行時間は約1週間（6日と22時間）である．QAPも下界の計算にLP緩和問題を用いるが，その総数は5,742億個以上に達し，総CPU時間は11年にも達する．この結果から見ても最適に解くことは大変困難であるが，今後はグリッド環境において合計数千プロセッサ以上を集めることも可能になり，さらに大きな問題への挑戦も可能になると思われる．

■半正定値計画問題に対する内点法の並列化

半正定値計画問題（Semidefinite Programming : SDP）は，組合せ最適化，制御分野，データマイニングおよび量子化学などに幅広い応用を持ち，主双対内点法によって多項式時間で最適解を導き出すことができるので，最近では注目度も高く，さかんに研究が行われている数理計画問題である．SDPは線形計画問題（LP）や凸二次計画問題などを含んだより大きな凸計画問題の枠組みであるが，半正定値制約という非線形制約を持っている．したがってSDPとして定式化できる最適化問題が解けるだけでなく，非凸最適化問題に対する強力な緩和値を導き出すことができる．そのためSDPを繰り返して解くことによって（最適に解くことがきわめて難しいが非常に実用上重要な）非凸最適化問題を扱える可能性を持っている[3]．また，著者らが開発したソフトウェアSDPA

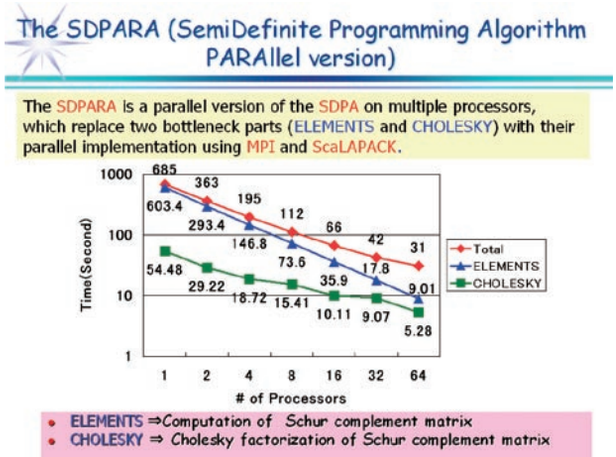


図-4　SDPARA

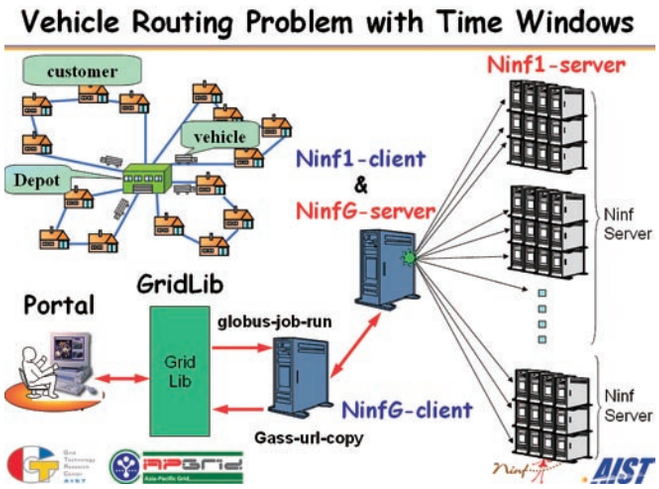


図-5　グリッド技術を用いた時間枠付き配送計画問題に対する求解サービス

(SemiDefinite Programming Algorithm) をはじめとして，複数の研究グループによってSDPに対するソフトウェアが開発され，インターネットより公開されている(詳しくは電子情報通信学会誌2003年10月号，半正定値計画問題に対するソフトウェアを参照のこと)．

SDPに対する主双対内点法は反復解法であるが，1回の反復において線形方程式を解くことが必要である．しかしSDPが大規模になった場合には，この線型方程式の係数行列の計算とその行列のCholesky分解が全体の計算時間の大部分を占めている．文献6) では，並列計算機(PCクラスタ) 上で主双対内点法の並列化に成功している(ソフトウェアSDPARA：**図-4**)．この方法ではMPIおよび線形代数演算ライブラリーであるLAPACKのMPI並列版であるScaLAPACKを用いて，係数行列の計算ならびにCholesky分解を並列計算している．その結果，係数行列のCholesky分解に時間が必要となる線形制約の数が多いSDPや，係数行列の計算に時間を必要とする密なSDPを効率よく解くことが可能になり，スケーラビリティが高いことも検証されている．文献6) では東京工業大学松岡研究室のPCクラスタ(PrestoIII：Athlon 1900+) を用いて，制約式の数が24,503である超大規模なSDPを1,984秒(CPU64個) で解き，問題サイズの世界記録を更新している．

今後の展望

これまで触れてきたTSPなどの最適化問題は主に理論的な興味から研究されているが，より実用的な最適化問題を解く研究も行われている．しかし，現在ではクラスタやグリッド計算などは最適化問題を解く必要がある一般ユーザには敷居が高いのも事実である．そのためにポータルサイトなどを用いてユーザが解きたい問題をWebサーバを介してクラスタ計算機に送信して，クラスタ計算機上で並列アルゴリズムによって問題を解き，結果をユーザに返すサービスが産総研などで研究されている(**図-5**)．このシステムではWANではNinf-GをLANではNinf-1を用いて通信する2段階のClient-Serverモデルが採用されている．すでに見てきたようにクラスタやグリッド技術を用いて最適化問題に対して目覚しい成果が達成されている．しかし，実用上必要な最適化問題の種類と規模を考えると，さらに多くの計算機資源が必要となる．これからもCPU単体の性能が向上するとともに，数千台規模のCPUをグリッド環境で使用することも可能になることが予想される．その場合には，現在では解くことが不可能な規模の問題に対する挑戦も可能になり，周辺の分野も含めて大きな成果が期待できる．

参考文献
1) Anstreicher, K. M.: Recent Advances in the Solution of Quadratic Assignment Problems, Mathematical Programming, Series B, **97**, pp.27-42 (2003).
2) Applegate, D., Bixby, R., Chvátal, V. and Cook, W.: Implementing the Dantzig-Fulkerson-Johnson Algorithm for Large Traveling Salesman Problems, Mathematical Programming, Series B, **97**, pp.91-153 (2003) .
3) Takeda, A., Fujisawa, K., Fukaya, Y. and Kojima, M.: Parallel Implementation of Successive Convex Relaxation Methods for Quadratic Optimization Problems, J. of Global Optimization **24**, pp.237-260 (2002).
4) Takeda, A., Kojima, M. and Fujisawa, K.: Enumeration of All Solutions of a Combinatorial Linear Inequality System Arising from the Polyhedral Homotopy Continuation Method, Journal of the Operations Research Society of Japan **45**, pp.64-82 (2002).
5) Tanaka, Y., Nakada, H., Sekiguchi, S., Suzumura, T. and Matsuoka, S.: Ninf-G：A Reference Implementation of RPC-based Programming Middleware for Grid Computing, J. of Grid Computing **1**, pp.41-51 (2003).
6) Yamashita, M., Fujisawa, K. and Kojima, M.: SDPARA：SemiDefinite Programming Algorithm PARAllel Version, Journal of Parallel Computing **29**, pp.1053-1067 (2003).

(平成15年12月27日受付)