課題と問題：説明と解説

計算機間ネットワーク設計問題

スパコングリッド TSUBAME のように，多数の計算機があり，それらが互いに通信しながら計算する場合，その通信のために計算機間をつなぐネットワークをうまく設計する必要がある．この設計の中でも，今回の本選課題では，計算機をどのようにグループ化し通信コストを減らすか，という点に焦点をおいた設計問題を考えることにしよう．

直感的にいえば，互いに通信を多く行う計算機同士が遠くに配置されてしまうと，通信時間もかかるし，計算機間ネットワーク全体の混雑を招き，全体の計算効率に影響を及ぼす．したがって，通信量の多い計算機同士を近くに置き，それらを特別な超高速ネットワークで結びたい．ただし，1 つの超高速ネットワークに，あまり多くの計算機を参加させると，その負荷のために超高速ネットワーク全体の速度がかえって遅くなる．そこで，計算機群を上手にグループ化して，全体でなるべく小さな通信コストとなるようにしたい．このグループ化が今回の問題である．

具体的には，多数（たとえば 数千個）の計算機を用いて，ある種の計算を行う場合[1]を考える．その際に計算機間の通信量を表わす通信量行列が与えられ（図 1），その行列から，全体の通信コストを最小とする計算機のグループ化を求める問題である．

１．用語の説明

以下では，総計算機数 を n で表わし，各計算機に $0 \sim n-1$ の番号を付ける．

通信量行列とは，各番号の計算機同士の通信量を以下のように行列で表わしたものである．通信量行列は M で，その i 行 j 列の成分を $M_{i,j}$ で表わす．各 $M_{i,j}$ は，計算機 i と j の間の通信量である．通信量は $0 \sim 9$ の自然数である．

	0	1	2	3	4
0	0	9	1	0	1
1	9	0	1	1	0
2	1	1	0	9	9
3	0	1	9	0	9
4	1	0	9	9	0

図 1: 総計算機数 $n = 5$ の場合の通信量行列の例

[1]ここでは，行いたい計算ごとに「最適なネットワーク」を用いることができる，という理想的な状況のもとでの設計を考えている．一方，TSUBAME のように実際のスパコングリッドでは，様々な計算に対して適度な効率が得られる汎用的なネットワーク設計が用いられている．

通信量は対称性を持つものとする．すなわち，すべての i, j で $M_{i,j} = M_{j,i}$ である．したがって，M は対称行列である．

超高速ネットワークを束ねるものを<u>スイッチ</u>と呼ぶことにする．同じスイッチに接続される計算機が同じ超高速ネットワークに属する計算機である．これらを<u>同一スイッチ下の計算機</u>という．そうでない計算機（異なるグループに属する計算機）を<u>異スイッチ下の計算機</u>という．

スイッチはいくつ使ってもよく，同一スイッチ下にいくつ計算機を抱えてもよい．便宜上，スイッチには 0, 1, 2, ..., と番号が付けられているものとする．あるネットワーク設計において使用するスイッチ数を m とする．そのときの各 k 番目のスイッチ下の計算機数を n_k で表わす．したがって，

$$n_0 + n_1 + n_2 + \cdots + n_{m-1} \ = \ n \tag{1}$$

である．また，各 k 番目のスイッチ下の計算機の集合を V_k で表わす．

与えられた通信量行列 M と，それに対する計算機のグループ化 $(V_0, ..., V_{m-1})$ に対し，ネットワークの<u>総通信コスト</u> $C = C_1 + C_2$ を以下のような規則で求める．

1. 異スイッチ下の計算機 i, j 間の通信コストは，$M_{i,j}$ として加算される．つまり，各 i, j に対して，計算機 i, j が同一スイッチ下にいる場合には $m_{i,j} = 0$，そうでない場合には $m_{i,j} = M_{i,j}$ とすると，異スイッチ下の計算機間の総通信コスト C_1 は次のようになる．

$$C_1 \ = \ \sum_{0 \le i < j \le n-1} m_{i,j}. \tag{2}$$

2. 各スイッチ k の計算機間の通信コストは，そのスイッチ下の計算機数を n_k として，$n_k(n_k - 1)/2$ とする．スイッチ下の計算機数が多いと超高速ネットワークの構造が複雑となり，スイッチ内のコストがそれだけ増えてしまうからである．したがって，同一スイッチ下の計算機間の総通信コスト C_2 は次のようになる．

$$C_2 \ = \ \sum_{0 \le k \le m-1} \frac{n_k(n_k - 1)}{2}. \tag{3}$$

２．問題の詳細説明

今回の問題を正確に記述すると次ページのようになる．以下では，審査の方法などを明確にする．

<u>審査方法</u>

本選の審査は，全チームから 6 チーム程度（東京地区 3 チーム，大阪地区 3 チームを予定）を選抜する「プレリミナリィ」と，選抜された 6 チームの中で上位 3 チームを決める「ファイナル」の二段階で行う．どちらも同一課題であり，各チームが提出するプログラムは一つである．ただし，入力データとプログラムを実行する制限時間が異なる．

スーパーコン０６の問題

与えられた計算機数 n と通信量行列 M に対し，総通信コスト C をできる限り小さくする計算機のグループ化を求めよ．

入力データの条件と形式

- 計算機数 n は $n \leq 3000$ とする．
- 各計算機間の通信量は $0 \sim 9$ の自然数とする．
- 入力データファイルは，次のように，1 行目が n，2 行目以降が通信量行列となるように作成する．

```
5
09101
90110
11099
01909
10990
```

出力の形式

計算機のグループ化を表す，以下のデータを出力する．

- 総通信量 C
- 各計算機が接続されるスイッチの番号を示す，長さ n の配列

プレリミナリィ：

3 種類の入力データに対する実行．各入力データにつき制限時間は 2 分（= 120 秒）．

ファイナル：

1 種類の入力データに対する実行．制限時間は 10 分（= 600 秒）．

ファイナルでは，入力データについてプログラムを実行し，制限時間内に出力された総通信コストによって，プログラムの性能を評価する．総通信コストが少ないものを勝ちとし，同じ総通信コストの結果を出力した場合には，出力までの計算時間が短いものを勝ちとする．（注：プレリミナリィでは 3 つのデータの各々で総通信コストを総合して評価する．詳細はここでは省略する．プレリミナリィ，ファイナル，ともに，制限時間内であれば，何度でも結果を出力してよい．複数出力された場合には，最後に出力された結果のみを審査の対象とする．）

入出力ルーチンについて

プログラム作成の便宜をはかるため，また審査を公平に効率よく行うために，データの入出力については委員会が用意する関数を決められた方法で利用すること．具体的な方法や引数の受け渡しについては，別途コンテストのページを参照して欲しい．

３．解説

3.1 総通信量の別計算法

総通信コスト C は，次のような式でも求めても同じことである．

$$C \;=\; \sum_{0\le i<j\le n-1} M'_{i,j}, \quad \text{ただし，} M'_{i,j} \;=\; \begin{cases} M_{i,j}, & i,j \text{ が異スイッチ下にいるとき，} \\ 1, & i,j \text{ が同一スイッチ下にいるとき．} \end{cases} \tag{4}$$

つまり，同一スイッチ下にいる場合には，通信量がいくらでも（また無くても）つねに 1 の通信コストがかかる，と考えて，総通信量を求めたのが総通信コストである．

3.2 グループ化のヒント

スイッチをまたがる通信コスト　C_1　を小さくするためには，通信量の多い計算機どうしを同一スイッチに接続するのが良い．図 1 の例では，計算機 No.0 と No.1 の間の通信が多く，また計算機 No.2, No.3, No.4 同士の通信も多いことが分かる．これから，一つのスイッチに計算機 0 と 1 を，別のスイッチに計算機 2, 3, 4 を接続するのが良いと推測できる．

このときの C_1 は

$$C_1 \;=\; M_{0,2} + M_{0,3} + M_{0,4} + M_{1,2} + M_{1,3} + M_{1,4} \;=\; 4$$

である．一方，スイッチ内部の通信コスト C_2 については，前者が 2 台の計算機からなり，後者が 3 台の計算機からなるため，

$$C_2 \;=\; \frac{2(2-1)}{2} + \frac{3(3-1)}{2} \;=\; 1+3 \;=\; 4$$

である．よって総通信コスト $C = 4+4 = 8$ である．

このグループ化を出力するのであれば，各計算機のグループ番号を計算機番号の順にならべ，0, 0, 1, 1, 1 のような配列を出力する．なお，1, 1, 0, 0, 0 という配列でも同じことである．

さて，図 1 の例において，計算機の全てを 1 つのグループにまとめてしまう設計も考えられる．このとき $C_1 = 0$ となる．しかし，これでは 1 グループが 5 台の計算機を抱えることになり，そのコストは

$$C_2 \;=\; \frac{5(5-1)}{2} \;=\; 10$$

となるため，総通信コストは $C = 0+10 = 10$ となり，先ほどの $C = 8$ の場合よりも損になってしまう．(逆に，スイッチを 5 台とし，各計算機がそれぞれ別のスイッチに接続される場合はどうだろうか？　各自でチェックしておこう．)

3.3 局所探索法

この課題におけるグループ化の種類の総数は膨大なものであり，たとえば計算機数 n が 100 以上になると，TSUBAME をもってしてもすべてのグループ化を調べつくすのは不可能であろう．そのため，最良のグループ化を求めるのではなく,「なるべく良い」グループ化を探すよう努力するのが現実的である．

良く知られた手法は局所探索法と呼ばれるもので，ごく単純に述べると以下のようなものである．

(1) 適当なグループ化をまず作ってみる．
(2) 以前のグループ化から,「少し変えた」グループ化を作ってみる
(3) 新しいグループ化の総通信コストが，以前のものより少なくなっていれば，そのグループ化を採用し，(2) に戻る．

上記手順 (2) の，少し変える手法にはいろいろある．たとえば，1 つの計算機を選んで別のグループに移してみる，2 つのグループを選んで合併してみる，1 つのグループを選んで 2 つのグループに分けてみる，などが考えられる．ほかの手法はどうだろうか？ また，多数の MPI プロセスにどういう仕事をさせるのが良いだろうか？ このあたりが勝負の分かれ目になるかもしれない．

上記の局所探索法を適用しただけでは，一般には局所解に落ち着いてしまう．つまり,「少し変えた」解では，もう改良できないような解である．残念ながら，それが最もよい解 — 最適解 — である保証はない．むしろ，最適解でない場合が多い．

局所探索法で考えなければならない点は，局所解が得られたときに，それから先，どのように計算を進めるべきか，という点である．局所的な改良には行き詰まってしまった場合の対策である．