Finding near optimal flows

Magdy A. Ahmed
Computer Sciences Dept., Faculty of Engineering, Alexandria University, Alexandria, Egypt
e-mail : magdy_aa@hotmail.com

This paper deals with the problem of approximating optimal flows between a single origin-destination pair in data networks. We consider the problem of splitting a unit flow among N paths having different costs under a special, yet important class of cost functions. Despite the existence of analytical methods for finding exact solutions to this problem, these solutions contain in many instances irrational values that cannot be implemented exactly. We present a fast algorithm that produces near optimal flows of the form b^{-i} where b,i are integers.

يتناول البحث مشكلة إيجاد التدفقات المثلى تقريبا بين اى عقدة كمصدر وأخرى كنهاية فى شبكات البيانات، بحيث يتم تقسيم وحدة من التدفق على N من المسارات بين هاتين العقدتين وكل مسار له تكلفة مختلفة. بالرغم من وجود طرق تحليلية لإيجاد الحل الأمثل ولكن نتائجها دائما تعطى قيم غير نسبية لايمكن تطبيقها عمليا. ويقدم البحث خوارزم سريع لإيجاد التدفقات المثلى تقريبا على الشكل b^{-r} بحيث ان r,b تكون قيم صحيحة ويعطى نتائج نسبية يمكن تطبيقها عمليا.

Keywords: Network flows, Optimal routing, Generalized huffman procedure

1. Introduction

Routing is a sophisticated data network function that requires coordination between the network nodes through distributed protocols and has a direct effect on the average packet delay, and the network throughput.

A more sophisticated alternative is optimal routing based on flow models. Several algorithms were given for the computation of optimal routing, both centralized and distributed [1-5].

In this paper, we consider the general problem of finding the optimal flows over an N-link network with one origin and one destination shown in fig. 1, that minimize the general cost function:

$$C = \sum_{i=1}^{N} P_i \, X_i^{v} \quad , \qquad (1)$$

subject to the constraints:

$$X_i > 0\,,\ X_1 + X_2 + ... + X_N = 1\ ,$$

where each P_i is a positive real constant and v is a negative real value.

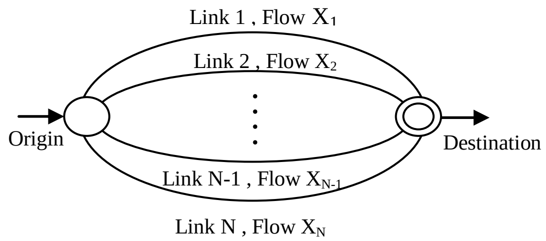

Fig. 1. Routing problem involving a single *OD* pair and *N* paths.

The particular cost function C of eq. (1) with v being negative applies to the cases where each channel has an abundance of available bandwidth and users are penalized for leaving a channel idle.

When v is negative the given optimization problem has a simple closed form solution, [6, 7]:

$$X_i^* = \frac{\sqrt[1-v]{P_i}}{\sum_{i=1}^{N} \sqrt[1-v]{P_i}} \, . \qquad (2)$$

The solution X_i^* is in general an irrational number. To be able to implement the splitting of the traffic input into the N-path flows, $X_1, X_2, X_3, ..., X_N$, each of these must be a rational number. For example, implementing the splitting of the input flow into

$\frac{1}{2}:\frac{1}{4}:\frac{1}{4}$ can be done by assigning 2 time slots to the first flow and 1 time slot to the second and the third, but how do we split into $\frac{1}{\sqrt{2}}$: $1-\frac{1}{\sqrt{2}}$. So an exact optimal solution cannot in general be implemented.

A possible solution is obtained by approximating the optimal solution to rational values but the length of the duty cycle may become too large (see sec. 3.2).

2. Proposed method

In this section, we describe the proposed algorithm, then we follow by examples to demonstrate the process.

2.1. Algorithm

The algorithm is based on finding flows of the form $X_i = b^{-r_i}$ that minimize the cost function in eq. (1) for a given $b \le N$, where b and r_i are integers . By substitution in eq. (1), the cost function is

$$C(b) = \sum_{i=1}^{N} P_i (b^{-r_i})^{\upsilon} = \sum_{i=1}^{N} P_i a^{r_i} \quad , \qquad (3)$$

where $a = b^{-\upsilon} > 1$ with υ negative.

It was shown [8, 9] that a modified Huffman procedure can be used to minimize the cost function in eq. (3), when a is any real positive number. For a>1 and a particular value for $b \le N$ we use the modified Huffman procedure to find optimal solutions to (3) which will result in flows of the form $X_i = b^{-r_i}$.

The procedure is then repeated for all possible values of b and the one yielding the minimum value for $C(b)$ is chosen as our solution.

The number of times needed to try is just the number of different tree orders for which a complete tree with N leaves exists. This number is particularly small and is given by integer solutions to the equation:

$$b = \frac{(N+k)}{(k+1)} \quad , \quad k = 0,1,2,3,....,N-2 \ .$$

It should be noted that if $b1$, $b2$ are two possible values for b and $b2$ is a power of $b1$ then $C(b1) \le C(b2)$ and so evaluation is not needed for values of b that are powers of previously evaluated ones. For example when N=49, there are 10 possible values for b, namely 2,3,4,5,7,9,13,17,25,49 of which only the 6 values 2,3,5,7,13,17 should be evaluated.

A formal description of the algorithm is shown in fig. 2.

3.2. Example

The following example demonstrates how the proposed algorithm is applied to a 5-link network with $P_1 = 7$, $P_2 = 6$, $P_3 = 3$, $P_4 = 2$, $P_5 = 1$ and υ=-1, the cost function in eq. (1) becomes:

$$C = \frac{7}{X_1} + \frac{6}{X_2} + \frac{3}{X_3} + \frac{2}{X_4} + \frac{1}{X_5}$$

For comparison purpose, we consider the following two cases.

```
Initial-Flows(N , v , p[N])
Begin
  Find the complete tree of order b=2
  Compute C(2)
  Let Min-Cost = C(2)
  For k=N-3 To 0 Step –1
    Begin
      Let  b=(N+k)/(k+1)
      If ( b is an integer value)
        Then
          If (b is not powers of previous b's)
            Then
                Compute C(b)
              If (C(b) < Min-Cost)
              Then
                    Min-Cost = C(b)
              End If
          End If
      End If
      Compute the flows X's
    End For
End.
```

Fig. 2. Formal description of the algorithm.

Case 1:

For optimal flows, applying eq. (2) results in

$X_1^* = 0.28629$, $X_2^* = 0.2650531$,
$X_3^* = 0.1874208$, $X_4^* = 0.1530284$,
$X_5^* = 0.1082074$,

and the minimum cost = 85.405446.

By rounding X_i^* to rational values, we get the flows:

$\tilde{X}_1^* = 0.3$, $\tilde{X}_2^* = 0.3$, $\tilde{X}_3^* = 0.2$, $\tilde{X}_4^* = 0.1$,
$\tilde{X}_5^* = 0.1$.

In this case the minimum cost = 88.3333 and the duty cycle = 10

Case 2:

By applying the proposed method. Since *N*=5, complete trees can be formed for *b*=2,3,4,5. The case *b*=4 is deleted since it is already contained in the case *b*=2, and only the three cases *b*=2,3,5 will be considered. For each case, the modified Huffman procedure is applied to construct the tree that minimizes the cost *C*, for the particular value of *b* considered, and the minimum value *C(b)* is determined. The results obtained are summarized in table 1 while the resulting tree for *b*=2 is shown in fig. 3.

From table 1, the minimum value for the cost function is *C*(2) and the flows are:

$$X_1 = \frac{1}{4},\ X_2 = \frac{1}{4},\ X_3 = \frac{1}{4},\ X_4 = \frac{1}{8},\ X_5 = \frac{1}{8}.$$

Note that, at the cost of increasing the duty cycle from 10 to 100, we can use the approximate values $\tilde{X}_1^* = 0.29$, $\tilde{X}_2^* = 0.26$, $\tilde{X}_3^* = 0.19$, $\tilde{X}_4^* = 0.15$, $\tilde{X}_5^* = 0.11$, which will give the better minimum cost of 85.41.

Our algorithm gives the minimum cost of 88 which is 3% of the optimal with duty cycle of 8.

As another example if we use the values P_i = {42 , 20 , 6 , 5 , 2} with υ=-1 the minimum cost is obtained for *b*=3 giving *C*(3) = 303 with the flows X_i = {1/3 , 1/3 , 1/9, 1/9 , 1/9} and duty cycle 9.

Table 1
Results of proposed method

Flows	*b*=2	*b*=3	*b*=5
X_1	1/4	1/3	1/5
X_2	1/4	1/3	1/5
X_3	1/4	1/9	1/5
X_4	1/8	1/9	1/5
X_5	1/8	1/9	1/5
Cost, *C(b)*	88	93	95
duty cycle	8	9	5

3. Experimental results

In this section, we compare the results obtained using our method with the optimal flows for networks with different number of paths (*N*) and different values for the costs P_i. To obtain reliable results, a number of independent replications where carried out and averaged [10-11]. Fig. 4 shows the percentage error ratio versus the number of paths in the range 5 to 50, indicating a deviation of no more than 6% of the optimal.

5. Conclusions

In this paper, we have introduced a new procedure for finding nearly optimal flows over *N*-paths between every *OD* pair of nodes

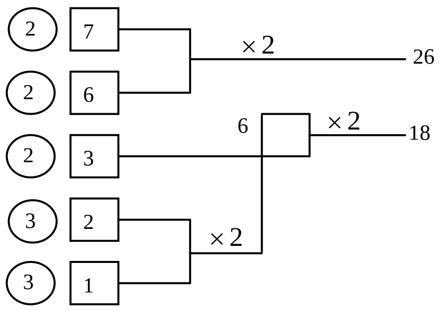

Fig. 3. A complete tree of order *b*=2.

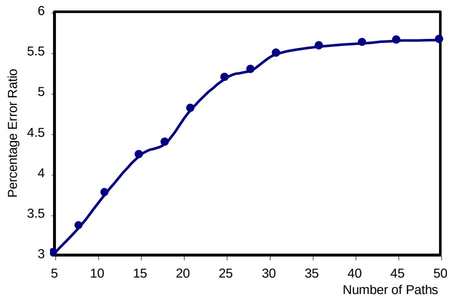

Fig. 4. Percentage error ratio of the proposed algorithm versus the number of paths.

in a network. The method determines the flows X_i by determining the order b, $b \le N$, and the depths of the leaves r_i of the complete tree that minimizes the cost function $C = \sum_{i=1}^{N} P_i \, X_i^{\upsilon}$,

where $X_i = b^{-r_i}$ and υ is any negative real value.

References

[1] D. Bertsekas and R. Gallager, Data Networks, Prentice-Hall (1987).

[2] S. Lanphongpanich and D. Hearn, "Simplicial Decomposition of the Asymmetric Traffic Assignment Problems", Trans Res. 18B:123-133 (1984).

[3] D.P. Bertsekas E.M. Gafni and R. G. Gallager, "Second Derivative Algorithms for Minimum Delay Distribution Routing in Networks", IEEE Trans Commun., COM-32:911-919 (1984).

[4] R.S. Dembo and J.G. Klincewicz, "A Scaled Reduced Gradient Algorithm for Network Flow Problems with Convex Separable Costs", Math Prog Studies. 15:125-147 (1981).

[5] D.P. Bertsekas, "Algorithms for Nonlinear Multicommodity Network Flow Problems", International Symposium on Systems Optimization and Analysis New York:Springer-Verlag,pp.210-224, (1979).

[6] L.S. Losdon, Optimization Theory for Large Systems, New York: Macmillan. (1970).

[7] R.T. Rockafellar, "Convex Analysis", Princeton, NJ: Princeton University Press, (1970).

[8] A.A. Belal, "A More General Optimization Problem for Uniquely Decodable Codes", IUTAM Symposium on Discrete Structural Optimization, Zakopane, Poland, Sept. (1993).

[9] A.A. Belal, "An Optimization Algorithm for Complete Code Trees", International Conference on Telecommunications, Dubai (1994).

[10] S.S. Lavenberg, Computer Performance Handbook, Academic Press (1983).

[11] H.A. Taha, Operations Research : An Introduction, Macmillan Publishing Co., 6th edition (2001).

Received December 2, 2003
Accepted December 29, 2003