Two- Way Rounding

[Originally published in SIAM Journal on Discrete Mathematics 8 (1995), 281-290.]

Hà Đỗ Thái An 21110362
Lương Chin Du 21110870
Nguyễn An Trần Điền 21110422
Poster đồ án cuối kỳ
Data Structure and Algorithm

HCMUTE

Main result

Given n real number $0\leq x_1 \leq 1, 0 \leq x_n<1$, and a permutation σ of {1,…,n}, we can always find integers $\hat{x}_1 \epsilon\{0,1\},\dots, x_n \epsilon\{0,1\}$ so that the partial sums $\hat{x}_1 + \cdots + \hat{x}_k$ and $\hat{x}_{1\sigma} + \dots + \hat{x}_{k\sigma}$ differ from the unrounded values $x_1 + \cdots + x_k$ and $x_{1\sigma} + \cdots + x_{k\sigma}$ by at most n/(n+1), for $1 \leq k \leq n$. The latter bound is best possible. The proof uses an elementary argument about flows in a certain network, and leads to a simple algorithm that finds an optimum way to round.

→The best possible discrepancy bounded by n/(n+1).

Two way rounding definition:

Let $x_1, \dots, x_n$ be real numbers and let σ be a permutation of {1,…,n}

$$\Rightarrow s_k = x_1 + \cdots + x_k, \qquad \Sigma_k = x_{1\sigma} + \cdots + x_{k\sigma}, \qquad 0\leq k \leq n,$$

for the partial sums in two independent orderings to find integers $\hat{x}_1, \dots, \hat{x}_n$ such that

$$\lfloor x_k \rfloor \leq \hat{x}_k \leq \lceil x_k \rceil,$$

and such that the rounded partial sums

$$\hat{s}_k = \hat{x}_1 + \cdots + \hat{x}_k, \qquad \hat{\Sigma}_k = \hat{x}_{1\sigma} + \dots + \hat{x}_{k\sigma}$$

also satisfy

$$\lfloor s_k \rfloor \leq \hat{s}_k \leq \lceil s_k \rceil, \qquad \lfloor \Sigma_k \leq \rfloor \hat{\Sigma}_k \leq \lceil \Sigma_k \rceil, \qquad 0\leq k \leq n.$$

Such $\hat{x}_1, \dots, \hat{x}_n$ will be called a two-way rounding of $x_1, \dots, x_n$ *with respect to* σ.

Algorithm:

Construct a network with nodes {s,$a_1 \dots a_m$, $u_1 \dots u_n$, $b_1 \dots b_m$,t} and the following arcs for $1\leq j \leq m$ and $1\leq k \leq n$:

s→ a_j and $b_j \to t$;

$u_k \to v_k$;

$a_j \to u_k$ if [j-1..j] ∩ [S_{k-1}.. S_k] ≠ 0;

$v_{k\sigma} \to b_j$ if [j-1..j] ∩ [Σ_{k-1}.. Σ_k] ≠ 0.

or

$a_j \to u_k$ if j-1+ϵ < S_k and j-ϵ > S_{k-1};

$v_{k\sigma} \to b_j$ if j-1+ϵ < Σ_k and j-ϵ > Σ_{k-1}.

(if ϵ is any fixed positive number < 1/(2m+2))

The capacities of the “source” arcs s→ a_j, the “middle” arcs $u_k \to v_k$, the “sink” arcs $b_j \to t$ all remain 1.

The optimum discrepancy δ=1-ϵ is obtained when ϵ is just large enough to reduce the network to the point where no m unit flow can be sustained. We can in fact find an optimum rounding as follows: Let :

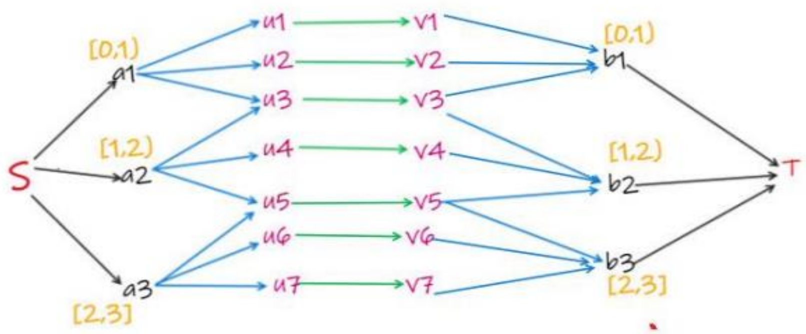
f(j,k) =min(j- S_{k-1}, $S_k - j + 1$) be the “desirability” of the arc $a_j \to u_k$

g(j,kσ)=min(j- Σ_{k-1}, Σ_k-j+1) be the “desirability” of the arc $v_{k\sigma} \to b_j$

Sort these arcs by desirability, and add them one by one to initial arcs {s→ a_j, $u_k \to v_k$, $b_j \to t$ } until an integer flow of m unit is possible. Then let $\hat{x}_k$ be the flow in $u_k \to v_k$, for all k; this flow has discrepancy equal to 1 minus the desirability of the last arc added, and no smaller discrepancy is possible.

The running time of this algorithm is bounded by O(mn) steps.

m=3, n=7



Applications:

-*Linear programming*: if we have a linear program with a large number of variables and constraints, we may use two-way rounding to find a feasible solution that is close to optimal.

-*Graph algorithms:* two-way rounding can also be used in graph algorithms to approximate solutions to problems such as shortest path or minimum spanning tree. For example, if we are trying to find the shortest path between two points on a map, we may use two-way rounding to approximate the distance between the points, rather than calculating the exact distance.

-*Numerical integration*: this is a method for estimating the area under a curve. By approximating the curve using a series of straight lines, we can use two-way rounding to find an approximate value for the area under the curve.

-*Data compression*: this also is a method for reducing the size of a data file. For example, if we are trying to compress an image, we may use two-way rounding to approximate the values of the pixels in the image, which can help to reduce the size of the file.

References:

[1] Géza Bohus, “On the discrepancy of 3 permutations,” Random Structures & Algorithms 1(1990), 215-220.
[2] Jack Edmonds and D. R. Fulkerson, “Bottleneck extrema,” Journal of Combinatorial Theory 8 (1970), 299-306.
[3] L. R. Ford, Jr., and D. R. Fulkerson, Flows in Networks(Princeton, New Jersey: Princeton University Press, 1962).
[4] Giorgio Gallo, Michael D. Grigoriadis, and Robert E. Tarjan, “A fast parametric maximum flow algorithm and applications,” SIAM Journal on Computing 18 (1989), 30-35.
[5] Andrew V. Goldberg, Serge A. Plotkin, and Pravin M. Vaidya, “Sublinear-time parallel algorithms for matching and related problems, “ Journal of Algorithms 14 (1993), 180-213.
[6] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik, Concrete Mathematics(Reading, Massachusetts: Addition-Wesley, 1989).
[7] Donald Enuth, The Stanford GraphBase (New York: ACM Press, 1993).
[8] L. Lovász, J.Spencer, and K. vesztergombi, “Discrepancy of set-systems and matries,” European Journal of Combinatorics 7 (1986), 151-160.
[9] L. Mirsky, Transversal Theory (New York: Academic Press, 1971).
[10] Christos h. Papadimitrious and Kenneth Steiglitz, Combinatorial Optimization(Englewood Cliffs, New Jersey: Prentice-Hall, 1982).
[11] Alexander Schrijver, Theory of Linear and Interger Programming (London: Wiley, 1986).
[12] Joel Spencer, “Ten Lectures on the Probabilistic Method,” CBMS-NSF Regional Conference Series in Applied Mathematics 52 (Phila-Delphia, Pennsylvania: SIAM, 1987), Lecture 5.