Rounding in e-approximation algorithms

Kuipers, FA

DOI
10.1109/SCVT.2006.334379

Publication date
2006

Document Version
Accepted author manuscript

Published in
Proceedings: IEEE SCVT 2006

Rounding in ϵ-approximation algorithms

Fernando Kuipers
Network Architectures and Services,
Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands.
Email: F.A.Kuipers@ewi.tudelft.nl

***Abstract*— A common approach to deal with NP-hard problems is to deploy polynomial-time ϵ-approximation algorithms. These algorithms often resort to rounding and scaling to guarantee a solution that is within a factor $(1+\epsilon)$ of the optimal solution. Usually, researchers either only round up or only down. In this paper we will evaluate the gain in accuracy when rounding up and down. The main application of this technique upon which we focus is Quality of Service routing, and specifically the Restricted Shortest Path problem.**

I. Introduction

The digital revolution manifested a large-scale transformation of analog information into a binary representation of zeros and ones. The level of accuracy that can be attained is directly related to the amount of bits that is used. In general, a coarser granularity means a smaller (space and time) complexity. We will focus on the impact of the way of rounding on the accuracy of (Quality of Service) routing in computer networks.

We will first explain the notation that is used. A network is represented as a graph $G = (\mathcal{N}, \mathcal{L})$ consisting of a set $\mathcal{N}$ of N nodes and a set $\mathcal{L}$ of L links. Nodes represent the routers or switches in a network, while the links represent the communication links. We only consider connected graphs without self-loops and at most one link between a pair of nodes. A specific link in the set $\mathcal{L}$ between nodes u and v is denoted by (u, v). Each link $(u, v) \in \mathcal{L}$ from node u to node v is characterized by a cost and a delay. For these additive QoS measures, the value (further called the weight) of the QoS measure along a path is the sum of the QoS weights on the links defining that path. For min-max QoS measures, the path weight of the QoS measure is the minimum (or maximum) of the QoS weights of the links that constitute that path. Rounding min-max weights has therefore a smaller impact than for additive weights, because the round-off errors do not add up along the entire path.

The rest of this paper is organized as follows: in Section II we briefly review the related work. In Section III we apply our results to the Restricted Shortest Path (RSP) problem. Simulation results are provided in Section IV. We end this paper in Section V with the conclusions.

II. Related work

Rounding is very often used in ϵ-approximation algorithms. An ϵ-approximation algorithm is an algorithm that is not necessarily exact, but which can provide a solution quantifiably close to the exact solution. The solution provided by an ϵ–approximation algorithm is guaranteed to be within a factor $(1+\epsilon)$ of the exact solution, where $\epsilon > 0$. Such a performance guarantee is not provided by heuristics and in this sense ϵ–approximation algorithms are considered to be better than heuristics. Unfortunately, their complexity is a function of $\frac{1}{\epsilon}$ and therefore their running time in practice can be excessive.

Almost all proposed ϵ–approximation algorithms in the field of QoS routing focus on the Restricted Shortest Path (RSP) problem. The RSP problem is a subproblem of QoS routing, in which the goal is to find a path P with minimal cost $c(P)$ that obeys one constraint Δ (typically) on the delay. Warburton [10] was the first to develop a fully polynomial-time approximation scheme for the RSP problem, assuming acyclic graphs. Hassin [4] improved this algorithm and provided an ϵ-approximation algorithm with complexity $O((\frac{LN}{\epsilon}+1)\log\log B)$, where B is an upper bound on the cost $c(P)$ of a path. It is assumed that the link weights are positive integers. Hassin's ϵ-approximation algorithm initially determines an upper bound (U_B) and a lower bound (L_B) on the optimal cost denoted by OPT. Once these bounds are found, the approximation algorithm bounds the cost of each link by rounding and scaling it according to: $c'(u,v) = \left\lfloor \frac{c(u,v)(N-1)}{\epsilon L_B} \right\rfloor \ \forall\ (u,v) \in \mathcal{L}$. Finally, it applies a pseudo-polynomial-time algorithm on these modified weights. Lorenz and Raz [8] provided a further improvement, which is explained in Section IV. In [7] we have presented a QoS solution to device QoS algorithms that take into account the properties of the link-state update (LSU) policy (part of QoS protocol). When these two entities are considered separately, two sources of inaccuracy (in LSU policy and algorithm)are introduced or we require an exponential-time exact algorithm. Our solution eliminates the inaccuracy in the algorithm and thereby guarantees a polynomial-time complexity, whereas the inaccuracy in the LSU policy is bounded via ϵ and controlled by the ISP.

There are two possible ways of rounding, namely rounding up and rounding down. Usually, researchers choose just one of these possibilities. For instance, Goel et al. [3] round down, whereas Chen and Nahrstedt [1] round up. Instead of only rounding up (or down), one could consider mixing between rounding up and down. Chen et al. [2] randomly round up or down to have an expected error of 0 (under an assumed uniform distribution of the link weights). Another possibility besides randomly rounding up or down, is to round to the nearest number, such that the worst-case round-off error that can be made is halved. In this paper we evaluate the performance of rounding up and/or down.

III. Theory

In this section we will apply our theory to the RSP problem. We will base our theory on the SEA algorithm [8] and its proof of correctness. SEA initially determines an upper bound (U_B) and a lower bound (L_B) on the optimal cost via a testing procedure. SEA improves upon Hassin's algorithm by finding better upper and lower bounds and by improving the testing procedure. Once these bounds are found, SEA bounds the cost of each link by rounding up and scaling it according to: $c'(u,v) = \left\lfloor \frac{c(u,v)(N+1)}{\epsilon L_B} \right\rfloor + 1 \ \forall \ (u,v) \in \mathcal{L}$. Finally, it applies an exact pseudo-polynomial-time algorithm on these modified weights. In this way SEA obtains the polynomial complexity of $O(LN(\log\log N + \frac{1}{\epsilon}))$. Let $S = \frac{\epsilon L_B}{N+1}$. The correctness of SEA is proved via three lemmas (see [8] for their proofs):

Lemma 1: Let P be any path, and $c'(u,v) = \left\lfloor \frac{c(u,v)(N+1)}{\epsilon L_B} \right\rfloor + 1 \ \forall \ (u,v) \in \mathcal{L}$, then

$$c(P) \leq c'(P)S \leq c(P) + NS$$

Lemma 2: Any path P returned by SEA satisfies

$$c(P^*) \leq c(P) \leq U_B + (N+1)S = U_B + \epsilon L_B$$

Lemma 3: If $U_B \geq c(P^*)$, then SEA returns a feasible path P that satisfies

$$c(P) \leq c(P^*) + \epsilon L_B$$

The SEA algorithm only rounds up via $c'(u,v) = \left\lfloor \frac{c(u,v)(N+1)}{\epsilon L_B} \right\rfloor + 1 \ \forall \ (u,v) \in \mathcal{L}$. We now investigate SEA when running it with costs that are rounded down[1] as follows:

$$c'(u,v) = \left\lfloor \frac{c(u,v)(N+1)}{\epsilon L_B} \right\rfloor \ \forall \ (u,v) \in \mathcal{L}$$

We set $U' = \left\lfloor \frac{U_B}{S} \right\rfloor + 1$. The correctness of the algorithm when using these rounded-down costs follows in an analogous way, with three lemmas.

Lemma 4: Let P be any path, and $c'(u,v) = \left\lfloor \frac{c(u,v)(N+1)}{\epsilon L_B} \right\rfloor \ \forall \ (u,v) \in \mathcal{L}$, then

$$c(P) - NS \leq c'(P)S \leq c(P)$$

Proof: For each $(u,v) \in \mathcal{L}$, we have $\frac{c(u,v)}{S} - 1 \leq c'(u,v) \leq \frac{c(u,v)}{S}$ and hence $c(u,v) - S \leq c'(u,v)S \leq c(u,v)$ and

$$\begin{aligned} c(P) &= \sum_{(u,v)\in P} c(u,v) \geq S \sum_{(u,v)\in P} c'(u,v) = c'(P)S \\ &\geq c(P) - (N-1)S \end{aligned}$$

■

[1] When rounding down we may create zero-valued link costs. Hence care must be taken to avoid loops. In our case, we have non-zero delays, which avoid looping.

Lemma 5: Any returned path P satisfies

$$c(P^*) \leq c(P) \leq U_B + \varepsilon L_B$$

Proof: By definition $c(P^*) \leq c(P)$ and $c'(P) \leq U' = \left\lfloor \frac{U_B}{S} \right\rfloor + 1$. Since $c'(P) \leq U'$, we have $c(P) \leq c'(P)S + (N-1)S \leq U'S + S + (N-1)S = U'S - S + (N+1)S \leq U_B + \epsilon L_B$. The result follows from Lemma 4. ■

Lemma 6: If $U_B \geq c(P^*)$, then a feasible path P is returned that satisfies

$$c(P) \leq c(P^*) + \epsilon L_B$$

Proof: For each $(u,v) \in P^*$, we have $c'(u,v) \leq \frac{c(u,v)}{S} + 1$. Thus

$$c'(P^*) = \sum_{(u,v)\in P^*} c'(u,v) \leq \frac{c(P^*)}{S} \leq \frac{U_B}{S} \leq U'$$

Since the algorithm includes U', we surely know that P^* is examined. By Lemma 4, $c'(P^*)S \leq c(P^*)$. Since P^* is examined by the algorithm and since it seeks to minimize c', we must have for the returned solution P that $c'(P) \leq c'(P^*)$. Combining with Lemma 4 we get

$$c(P) \leq c'(P)S + \epsilon L_B \leq c'(P^*)S + \epsilon L_B \leq c(P^*) + \epsilon L_B$$

■

The above proofs are intuitively explained when considering that the maximal rounding error (for both rounding up as rounding down) that can be made on a link equals S. Since a path can have no more than $(N-1)$ hops, the total error that can be accumulated along the path equals $(N-1)S < \epsilon L_B$. If we extend this reasoning to rounding to the nearest integer, then the total error that can be accumulated along the path equals $(N-1)\frac{S}{2} < \frac{\epsilon}{2} L_B$.

By computing two paths via rounding up and down, we can return the one that has minimum cost. The worst-case complexity remains $O(LN(\log\log N + \frac{1}{\epsilon}))$. Moreover, if the two paths are the same, there is a high probability that we have found the exact path. Note, that it is easily shown that if both paths are the same, that this is not a guarantee for having returned the exact path. We will demonstrate this for the shortest path problem, in Figure 1, where the rounding granularity $g = 0.1$. The left graph is the original graph G, the middle graph G_- is the graph G with link weights rounded down, and the right graph G_+ equals G with link weights rounded up. It is clearly seen that although the shortest paths

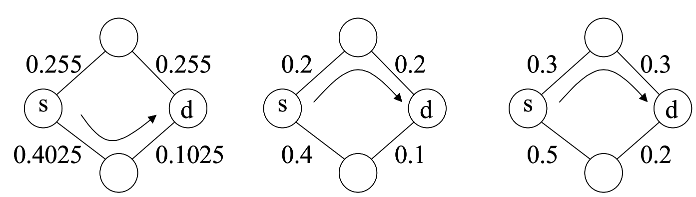

Fig. 1. $g = 0.1$, the left graph is G, the middle graph G_-, and the right graph equals G_+

in the rounded graphs are the same, they do not equal the shortest path in the original graph G.

IV. SIMULATIONS

In this section we will present our simulation results. We will consider four ways of rounding, namely (1) rounding up, (2) rounding down, (3) two-pass rounding that selects the best path from (1) and (2), and (4) (one-pass) rounding to the nearest integer. We use an exact algorithm SAMCRA [9] to find the optimal cost $OPT = c(P^*)$. Based on this value we round the costs as $c'(u,v) = \left[\frac{c(u,v)(N+1)}{\epsilon OPT}\right]$, where $[x]$ refers to the rounded value of x (either rounded up, down, ...) and recalculate the RSP path based on the rounded costs. We have performed simulations on random graphs of the type $G_p(N)$, where p is the link density, and two-dimensional square lattices. In the class of random graphs, the delay and cost of every link $(u,v) \in \mathcal{L}$ were taken as independent uniformly distributed random integers in the range $[1, M]$. For the class of lattices, the delay and the cost of every link (u,v) were negatively correlated: the delay was chosen uniformly from the range $[1, M]$ and the corresponding cost was set to $(M+1)$ minus the delay. According to [5] this scenario constitutes a worst case for the RSP problem. We have chosen $M = 10^5$. In each simulation experiment, we generated 10^4 graphs and selected nodes 1 and N as the source and destination, respectively. For lattices, this corresponds to a source in the upper left corner and a destination in the lower right corner, leading to the largest minimum hop count. For the random graphs, this is equivalent to choosing two random nodes.

The delay constraint Δ was selected as follows. First, we computed the least-delay path (LDP) and the least-cost path (LCP) between the source and the destination using Dijkstra's algorithm. If the delay constraint $\Delta < d(\text{LDP})$, then there is no feasible path. If $d(\text{LCP}) \leq \Delta$, then the LCP is the optimal path. Since these two cases are easy to deal with, we compared between the algorithms considering the values $d(\text{LDP}) < \Delta < d(\text{LCP})$, as follows:

$$\Delta = d(\text{LDP}) + \frac{1}{2}(d(\text{LCP}) - d(\text{LDP})) \tag{1}$$

We have evaluated the algorithms that use rounding based on how successful they are in minimizing the cost of a returned feasible path, when compared to the exact algorithm. To this end we define the effective approximation α as

$$\alpha = \frac{c(P_x)}{OPT} - 1$$

where $c(P_x)$ is the cost of the feasible path that is returned by algorithm x. We plot $E[\alpha]$ based on the 10^4 iterations. Figure 2 displays the effective approximation α as a function of ϵ for the random graphs with $N = 100$ and $p = 0.2$ and lattices with $N = 25$.

We can see that $\alpha << \epsilon$, which has readily been observed and explained in [6] for the case of rounding up. The differences between the different ways of rounding are not big when compared to the value of ϵ, but the relative difference may be considerable as observed for the random graphs. The performance of rounding to the nearest integer outperforms rounding up or down, which was expected since the worst-case error is halved. However, it is interesting to see that rounding down outperforms rounding up. Given uniformly distributed link weights, this result is somewhat surprising, since the expected error in link weights should be the same in both cases. A possible explanation is that we are searching for a minimum-cost path (instead of just any path), and the (small) rounded weights x, which are part of the shortest path, may on average be closer to $\lfloor x \rfloor$ than to $\lceil x \rceil$. This effect is likely increased with a coarser granularity/accuracy. Furthermore, in the class of random graphs with uniformly distributed link weights, the shortest path may not be the minimum-hop path. However, when we are rounding up, we are adding to the link weights, which may result in a preference for minimum-hop paths.

The ranking in performance in the class of lattices with negatively correlated link weights is the same as for the class of random graphs, except for a larger effective approximation α. We refer to [6] for an explanation of this phenomenon.

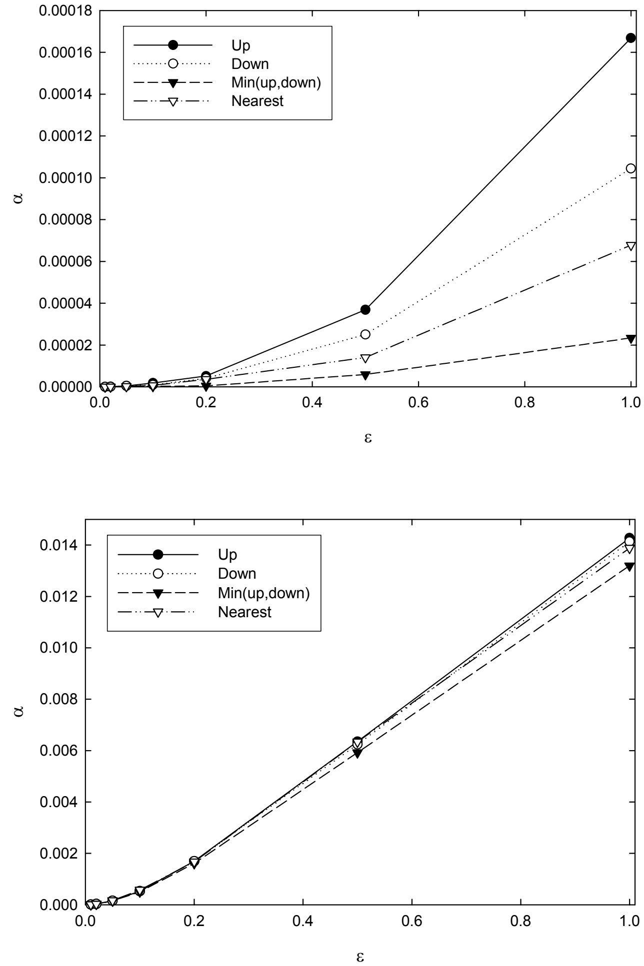

Fig. 2. Effective approximation in the class of (top) random graphs, and (bottom) lattices.

V. Conclusions

In this paper we have provided a theoretical and simulative analysis of the influence of rounding link weights either up or down. Rounding link weights often results in a better running-time, however it comes at the expense of accuracy. We have directed our attention to the Restricted Shortest Path (RSP) problem, which is an important subproblem of Quality of Service (QoS) routing. We have proposed to combine rounding up and down, either in one-pass or two-pass. Our simulations confirm the increase in accuracy, when not confining to one way of rounding.

References

[1] S. Chen and K. Nahrstedt, "On finding multi-constrained paths," Proc. of IEEE International Conference on Communications (ICC'98), June 1998.

[2] S. Chen, M. Song, and S. Sahni, "Two techniques for fast computation of constrained shortest paths," Proc. of IEEE Globecom, 2004.

[3] A. Goel, K.G. Ramakrishnan, D. Kataria, and D. Logothetis, "Efficient computation of delay-sensitive routes for one source to all destinations," Proc. of IEEE INFOCOM, April 2001.

[4] R. Hassin, "Approximation schemes for the restricted shortest path problem," Mathematics of Operations Research, vol. 17, no. 1, pp. 36–42, 1992.

[5] F.A. Kuipers and P. Van Mieghem, "Conditions that Impact the Complexity of QoS Routing," IEEE/ACM Transaction on Networking, vol. 13, no. 4, pp. 717-730, August 2005.

[6] F.A. Kuipers, A. Orda, D. Raz, and P. Van Mieghem, "A comparison of exact and ϵ-approximation algorithms for constrained routing", to appear in Proc. of Networking 2006 - Fifth IFIP Networking Conference, Coimbra, Portugal, May 15-19, 2006.

[7] F.A. Kuipers, "QoS protocol and algorithm join forces," Proc. of CHINACOM'06, Beijing, China, October 25-27, 2006.

[8] D.H. Lorenz and D. Raz, "A simple efficient approximation scheme for the restricted shortest path problem," Operations Research paper, vol. 28, no. 5, pp. 213-219, June 2001.

[9] P. Van Mieghem and F.A. Kuipers, "Concepts of Exact Quality of Service Algorithms," IEEE/ACM Transaction on Networking, vol. 12, no. 5, pp. 851-864, October 2004.

[10] A. Warburton, "Approximation of pareto optima in multi-objective, shortest path problems," Operations Research, 1:70-79, 1987.