

Consistent Rounding of Edge Weights in Graphs

Stefan Funke

Universität Stuttgart
funke@fmi.uni-stuttgart.de

Sabine Storandt

Julius-Maximilians-Universität Würzburg
storandt@informatik.uni-wuerzburg.de

Abstract

Often, the edge weights of graphs are given in implicitly infinite or overly high precision (think of Euclidean lengths) which leads to both theoretical as well as practical challenges. In this paper we investigate how to round edge weights of a given graph $G(V, E, w)$ such that the rounded weights of paths satisfy certain consistency criteria. Natural consistency criteria are, for example, preserving optimality of paths, and bounding relative change in weight after the rounding procedure. Low precision edge weights allow for more space efficient implementations, faster arithmetic operations, and in general more stable and efficient algorithms. We present an ILP based rounding approach as well as a greedy rounding heuristic. We show experimentally for large road networks and grid graphs that our new rounding approaches are significantly better than common deterministic or randomized rounding schemes.

Introduction

An optimal path from A to B in a weighted graph or network is normally defined as the one with the minimum accumulated edge weights among all A-B-paths. Thereby, edge weights can express a multitude of optimization goals, e.g. euclidean lengths to get short paths, travel times to get quick paths, risk factors to get safe paths, ascending slopes to get energy-saving paths and so on. For real-world networks, these weights have to be measured or estimated somehow. For example, euclidean lengths in road networks are normally inferred from node coordinates (latitude and longitude) and a formula for the spherical distance between those on the surface of the earth (which involves square root computation). GPS-based techniques allow to get latitude and longitude values with up to seven decimal places, which translates into a precision about 11 mm. But the question is, whether it is reasonable to have such precise edge weights, especially if e.g. most edges have a length of 10 meters or more. Overprecise edge weights are often not meaningful (is a path of length 15 km and 5 cm really perceptible longer than a 15 km long path?). Moreover they are naturally error-prone, demand a lot of space, and are expensive to carry out calculations with. A natural approach to deal with such

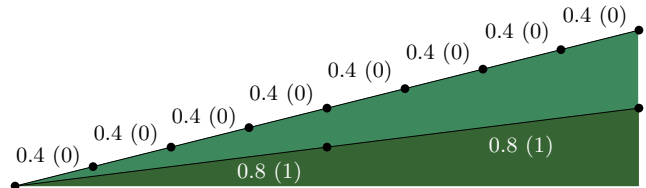


Figure 1: The image depicts two uphill paths with the edge weights reflecting the differences in altitude. The rounded weights are given in brackets. The lower path reaches an altitude of 1.6 and is rounded to 2. The upper path reaches twice this altitude but is rounded to 0, hence appearing to be flat.

edge weights would be to round them to the desired precision. But naive rounding rules lead to accumulated rounding errors which might distort the structure of optimal paths and do no longer reflect the real world sufficiently, see Figure 1 for an example.

The goal of this paper is to introduce rounding approaches which make sure that the accumulated rounding errors on optimal paths are bounded. Furthermore, we would like to preserve the structure of optimal paths, i.e. paths that were optimal before rounding should stay optimal after rounding as far as possible.

Related Work

Rounding a real valued $x \in \mathbb{R}$ to a $y \in \mathbb{Z}$ with $|x - y| < 1$ comes up in many contexts. In algorithmics and discrete optimization, rounding is a way of dealing with the fact that many problems have a natural formulation as *integer* linear programs, but obtaining a non-integral solution followed by rounding is much easier than determining an integral solution directly. Many approximation algorithms for NP-hard problems are obtained in that way, e.g. (Raghavan and Thompson 1987). In image processing, the digital halftoning problem asks for the transformation of a grayscale image with continuous intensities in $[0, 1]$ per pixel into black and white pixels only, see for example (Asano et al. 2002). In (Asano, Matsui, and Tokuyama 2000) the authors show that rounding a matrix with real-valued entries in $[0, 1]$ into a binary 0/1-matrix such that the difference in sums within any $k \times k$ submatrix of the original and rounded matrix is mini-

mized is NP-hard. In general, many such rounding problems turn out to be NP-hard.

For the very common scenario that edge weights are Euclidean distances in the plane, it is not even clear that the problem of computing shortest paths is in NP since it requires the comparison of sums of squares, which might require very expensive arithmetic operations. In fact, for graphs $G(V, E)$ where the vertices are points in \mathbb{Z}^4 and we have Euclidean edge weights, the shortest path problem is known to be sums-of-square-roots-hard (Kayal and Saha 2011). So to have polynomial-time algorithms outside the RealRAM model of computation, approximation of the edge weights seems to be necessary.

Contribution

We first analyze the potential of deterministic and randomized rounding for our application. We show that both of these baseline approaches are not sufficient to solve the weight rounding problem in graphs, as the accumulated rounding errors become too large. We then introduce an ILP formulation, which allows to bound the maximum relative error which occurs on a path. As this ILP is too expensive to be set up already for small graphs, we introduce a greedy rounding heuristic which provides us with an a posteriori maximum relative error guarantee. We conduct experiments on real-world road networks and synthetic grid graphs. On both benchmark sets our rounding heuristic outperforms deterministic and randomized rounding significantly.

Notation, Baselines and Error Analysis

From now on, we will use $w : E \rightarrow \mathbb{R}^+$ to indicate the original edge weights and $w' : E \rightarrow \mathbb{N}$ for the rounded weights. This might involve some scaling first, e.g. if we have edge lengths with a precision of centimeters and want to have the precision of meters, we first divide each weight by 100. If we want our weights to fit in a certain data type, e.g. only allowing integers in the range $0, \dots, 255$, then we scale with $\max_{e \in E} \lceil w(e) \rceil / 255$. The resulting weight on an edge e is then $w(e)$. With $\lfloor w(e) \rfloor$ we refer to the largest integer smaller than $w(e)$ and with $\lceil w(e) \rceil$ to the smallest integer larger than $w(e)$. We only allow a weight to be rounded to the next smaller or next larger integer. If $w(e)$ already is integer, we do not allow any kind of rounding.

Deterministic Rounding

The classic deterministic rounding rule demands to round a number down when $w(e) - \lfloor w(e) \rfloor < 0.5$ and up otherwise (sometimes with special treatment for the $= 0.5$ case). The maximum error per edge is hence bounded by 0.5. Therefore, the maximum absolute error on a path p of length k is $0.5k$ and the maximum relative error if $w'(p) > w(p)$ is

$$w'(p)/w(p) = (0.5k + 0.5k)/0.5k = 2.$$

If $w'(p) < w(p)$, the relative error is unbounded, though, as $w'(p)$ might be zero if all $e \in p$ have weights in $]0, 0.5[$. If we assume that $w(e) \geq 1$ for all $e \in E$, the ratio $w'(p)/w(p)$ is maximized for $2/1.5 = 4/3$ and $w(p)/w'(p)$ for $1.49/1 \approx 3/2$. So we have an ad hoc maximum relative error of $\max(4/3, 3/2) = 3/2$ for any fixed path.

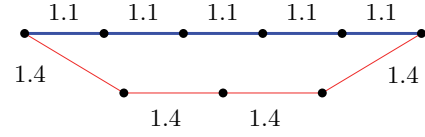


Figure 2: The upper blue path consists of 5 edges with a weight of 1.1, so the total weight is 5.5. The lower red path consists of 4 edges of weight 1.4, hence the total weight is 5.6 and therefore larger than the weight of the blue path. If all edge weights are rounded down to one, though, the red path gets a total weight of 4, while the blue path gets a rounded weight of 5. Hence the structure of the optimal path changes.

But in a graph, due to rounding, the structure of optimal paths might change. Hence it could be that after rounding some alternative path p becomes cheaper than the original path π , see Figure 2. The following Lemma shows that our maximum error bounds are still obeyed in this scenario.

Lemma 1 *Let $G(V, E, w)$ be a weighted graph with edge weights $w(e) \geq 1$ and $G'(V, E, w')$ the same graph with deterministically rounded weights. For every $s, t \in V$, the optimal s - t -path π in G and the optimal s - t -path p in G' fulfill the following inequalities:*

$$\frac{w(\pi)}{w'(p)} \leq \frac{4}{3} \quad \frac{w'(p)}{w(\pi)} \leq \frac{3}{2}$$

Proof. Assume for contradiction that $w(\pi) > 4/3 \cdot w'(p)$. As π is an optimal path in G , we know that $w(p) \geq w(\pi)$. Hence we get $w(p) > 4/3 \cdot w'(p)$ which contradicts our maximum relative error bound for path p .

Assume further for contradiction that $w'(p) > 3/2 \cdot w(\pi)$. We know that $w'(\pi) \leq 3/2 \cdot w(\pi)$, hence it follows $w'(p) > w'(\pi)$. But in this case, p can not be an optimal path in G' , which contradicts our definition of p . ■

Observation 2 *If we allow any kind of rounding to the next smallest or largest integer (deterministic, arbitrary, randomized), the maximum relative error becomes 2 if all original weights w are larger or equal to 1. The proof from Lemma 1 carries over to this scenario (as it is oblivious to the used constants), hence the bound of 2 also applies for alternative s - t -paths which become optimal after rounding.*

Randomized Rounding

The idea of randomized rounding is to interpret

$$(w(e) - \lfloor w(e) \rfloor) \in [0, 1]$$

as probability for rounding up. So 1.2 is rounded to 2 with a probability of 0.2 and to 1 with a probability of 0.8. Hence the expected path cost E is the same as the original cost $w(p)$ of a path, as

$$E = \sum_{e \in p} (\lfloor w(e) \rfloor + 1 \cdot (w(e) - \lfloor w(e) \rfloor)) = \sum_{e \in p} w(e) = w(p).$$

Unfortunately, this does not mean that the probability of some path exhibiting a large relative rounding error is small as the following example shows.

Consider a path p consisting of k edges, each with weight 0.4 and let Z be random variable denoting the cost of the path after rounding. Clearly, $E(Z) = w(p) = 0.4k$. Z follows the binomial distribution $B(k, 0.4)$ with variance $V(Z) = 0.26k$, which implies for example, that for $k = 12$, the probability that after the rounding the path p has weight 0 is more than 0.002 and that it has weight 12 is more than 0.000016 – instead of the ideal weights 4 or 5. For larger graphs it is hence quite likely that such a deviation from the desired weights appears somewhere.

ILP-Formulation

Next, we are going to introduce a suitable ILP formulation that leads to correctly rounded edge weights.

Let $\pi(s, t)$ be an optimal s - t -path in $G(E, V)$. The weight of π is $w(\pi) = \sum_{e \in \pi} w(e)$ and its rounded weight is $w'(\pi) = \sum_{e \in \pi} w'(e)$. We now demand that for some fixed $X \geq 1$, for an optimal path π in G it yields:

$$\frac{1}{X} \leq \frac{w(\pi)}{w'(\pi)} \leq X \quad (1)$$

As seen in Observation 2, if all original weights are ≥ 1 , then for $X \geq 2$ every rounding rule produces a feasible solution automatically.

Note, that for $X \in [1, 2]$ or when edge costs smaller than 1 are allowed, it does not make sense to enforce constraint (1) on paths with only a few edges. Especially considering single edges, rounding to the next integer would be impossible for small X . For example, consider $X = 1.1$ which says that the rounded weight of a path should only be 10% smaller or greater than the original weight. But then an edge weight of e.g. 0.4 can not be rounded, as the target value has to be in the range $[0.36, 0.44]$ which does not include any integer. In general, enforcing this constraint on paths with small hop length often leads to infeasibility as there is not enough flexibility left. Therefore, we add a parameter k which describes the minimum hop length of a path which we take into consideration.

Basic ILP

To set up the ILP, we introduce a decision variable x_e for every edge $e \in E$. Here, $x_e = 0$ if $w'(e) = \lfloor w(e) \rfloor$ and 1 otherwise. This allows to rewrite the above introduced constraints as follows:

$$\begin{aligned} \forall \pi \in G, |\pi| \geq k : \quad & \sum_{e \in \pi} \lfloor w(e) \rfloor + \sum_{e \in \pi} x_e \leq w(\pi) \cdot X \\ & \sum_{e \in \pi} \lfloor w(e) \rfloor + \sum_{e \in \pi} x_e \geq w(\pi) \cdot 1/X \\ \forall e \in E : \quad & x_e \in \{0, 1\} \end{aligned}$$

But extracting and storing every optimal path in G is very time and space consuming, and leads to $\mathcal{O}(n^2)$ constraints in the ILP. Therefore we use the following remedy: It suffices to consider all optimal paths with a hop length between

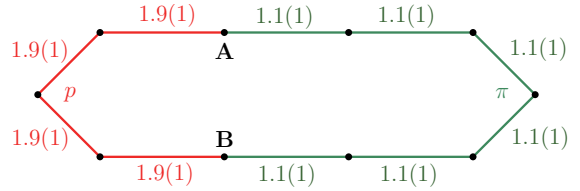


Figure 3: We consider the rounding problem with $k = 6$ and $X = 1.5$. The optimal path π from A to B is the green path. Obviously $w(\pi)/w'(\pi) = 6.6/6 < 1.5$. The red path p has a total weight of $w(p) = 7.6 > w(\pi)$. After rounding p becomes optimal as $w'(p) = 4 < w'(\pi) = 6$. But now $w(p)/w'(p) = 7.6/4 > 1.5$, which is legit as p has a hop length smaller than $k = 6$. This leads to $w(\pi)/w'(p) = 6.6/4 > 1.5$ also not obeying the maximum relative error bound.

k and $2k - 1$. All other paths are concatenations of such paths. And if for every subpath of length at least k the multiplicative error is at most X this also yields for the total path. For grids and typical road networks, this reduces the number of constraints to $\mathcal{O}(nk^2)$.

Preserving Optimal Path Structures

But the above ILP-formulation alone does not solve our rounding problem completely, as now a path p that was not optimal before rounding might become optimal after rounding. In general, a path p can only become optimal after rounding if

$$\sum_{e \in p} \lfloor w(e) \rfloor < \sum_{e \in \pi} \lfloor w(e) \rfloor. \quad (2)$$

One way to make sure that the structure of all optimal paths stays unaffected by rounding, is to enumerate all such paths p and add the constraint

$$\sum_{e \in p} (\lfloor w(e) \rfloor + x_e) \geq \sum_{e \in \pi} (\lfloor w(e) \rfloor + x_e)$$

for each such p to the ILP. But note, that there might be exponentially many such constraints to add for each optimal path; and not only for optimal paths up to length $2k - 1$ but also for longer ones. This makes setting up the ILP impractical already for small networks. Therefore, we will relax the constraint of unaffected optimal path structures, but instead demand that the weight of the optimal path p after rounding should obey the relative error of X compared to the weight of the optimal path π before rounding, i.e. $w'(p) \leq w(\pi) \cdot X$ and $w'(p) \geq w(\pi)/X$.

While $w'(p) > w(\pi) \cdot X$ cannot happen (as otherwise π would still be optimal instead of p), $w'(p) < w(\pi)/X$ might occur. To avoid this, we add the constraint $w'(p) \geq w(\pi)/X$ to the ILP for every path up to length $2k - 1$ if $\sum_{e \in p} \lfloor w(e) \rfloor \leq w(\pi)/X$. These might still be exponentially many paths, but only a small fraction of all paths in the network. Note that it does not suffice to only add the constraints for paths in the range of $k, \dots, 2k - 1$ as otherwise the X bound might still be violated, see Figure 3 for an example.

1.3	1.3	1.4	X1	X2	X3	max	%
1	1	1	1.3	1.35	1.3	1.35	29.4
1	1	2	1.3	1.1	1	1.3	19.6
1	2	1	1.16	1.1	1	1.16	12.6
2	1	1	1.16	1.35	1	1.35	12.6
*				≥ 1.25	≥ 1.25		25.8

Table 1: The first three columns show the rounding result. The * summarizes all results with at least two values rounded to 2. Xi indicates the relative error for path Pi. The last column gives the probability of the rounding being result of randomized rounding.

Example

We provide a small example to show that the ILP solution can be significantly better than the baseline solutions. Consider a path of three edges with weights 1.3, 1.3 and 1.4 and $k = 2$. So we have three subpaths to consider: P1 of weight $1.3 + 1.3 = 2.6$, P2 of weight $1.3 + 1.4 = 2.7$ and P3 the complete path of weight 4. Table 1 sketches all possible rounding results. The first row corresponds to deterministic rounding, which creates a relative rounding error of 1.35. The best solution would be rounding to 1, 2, 1 with a relative error of only 1.16. This would be the outcome of our ILP, e.g. with $X = 1.2$. The probability that this solution is derived from randomized rounding is less than 13%.

Heuristic Edge Weight Rounding

The ILP formulations are not suitable to solve large instances, though, as even with the introduced modifications the number of constraints is huge. Therefore, we now describe a heuristic approach that allows to achieve small rounding errors in practice while requiring significantly less processing time.

Algorithm

The basic idea of our approach is to also consider the paths of length $k, \dots, 2k - 1$ like in the ILP. But instead of establishing constraints and using the ILP solving machinery, we apply much simpler greedy strategy.

We assume we are given a set of paths P and the original weight w for each edge. The goal is to compute the rounded weight w' for each edge. Initially, we set $w' = w$ for all edges. We then build an index data structure, which returns for some edge $e \in E$ the set $P'(e) \subseteq P$ of paths that contain e . Afterwards, we sort the edges in E decreasingly by the number of paths they appear in. Then, we parse through the list of edges and overwrite the weight of the actual edge e with its rounded weight w' (such that $w'(e) \in \{\lfloor w(e) \rfloor, \lceil w(e) \rceil\}$) as follows: For each path $p \in P'(e)$, we compute its original weight $w(p)$ and its (partially) rounded weight $w'(p)$. We calculate $w_1(p) = w'(p) - w(e) + \lfloor w(e) \rfloor$ and $w_2 = w'(p) - w(e) + \lceil w(e) \rceil$ to simulate what happens if we round $w(e)$ down or up. We then compute the relative error of w_1 and w_2 compared to $w(p)$. We keep track of the largest occurring maximum relative errors X_{up}, X_{down} over all paths in P' when round-

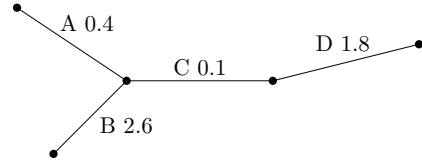


Figure 4: Example instance. The greedy algorithm considers the edges in order C, A, B, D setting the rounded costs to 0, 1, 3, 2.

ing up or down. In the end, we set $w'(e) = \lfloor w(e) \rfloor$ if $X_{up} > X_{down}$ and $w'(e) = \lceil w(e) \rceil$ otherwise.

Example

As an example, we consider the graph shown in Figure 4. Using $k = 2$, the set of paths consists of AB with cost 3, AC with cost 0.5, ACD with cost 2.3, BC with cost 2.7, BCD with cost 4.5 and CD with cost 1.9. We observe that C appears in 5 of these 6 paths, all other edges in 3 paths, respectively. With arbitrary broken ties, let's assume the algorithm considers the edges in order C, A, B, D. Setting the cost of C to 1, the highest relative error would arise from path AB with $X_{up} = 1.4/0.5 = 2.8$. For C=0, the highest relative error would also arise from AB, but the ratio $X_{down} = 0.5/0.4 = 1.25$ is smaller than X_{up} . Hence we round C down to 0. Then we proceed with edge A. As C was rounded to 0, we observe that rounding A down to 0 as well would lead to $X_{down} = \infty$. Hence A is rounded up to 1 for sure. Next, we consider edge B. $X_{down} = 2.7/2 = 1.35$ is defined by BC, $X_{up} = 4/3 = 1.3$ by AB. Hence, B is rounded up to 3. Last, we consider D. $X_{down} = 1.9/1 = 1.9$ arises from CD and $X_{up} = 3/2.3 < 1.4$ from ACD. So finally D is rounded up to 2 and all edges have integer weights.

A Posteriori Error Guarantee

If P is the set of all paths of length k to $2k - 1$ in G , and X is the maximum observed rounding error that resulted from any edge weight rounding in the course of the greedy algorithm, we can certify that for all optimal paths in G' longer than k the maximum relative error will never exceed X (if the respective optimal path in G has a length of at least k as well).

In the experiments, we will also test our approach on P containing only the shortest paths of length k to $2k - 1$. And to tackle graphs with millions of nodes and edges even only a subset of those. While this does not provide us with an a posteriori approximation guarantee, we will show that the outcome is still superior to rounded edge weights produced by deterministic or randomized rounding.

Experiments

We implemented our proposed heuristic as well as the baseline approaches in C++. Experiments were conducted on a single core of an Intel i5-4300U CPU with 1.90GHz and 12GB RAM.

	# nodes	# edges	min w	max w
SA	78,413	151,009	0.070606	591.98
SL	279,268	553,662	0.022239	1436.77
TU	669,875	1,375,845	0.014850	1723.52
ST	1,012,381	2,059,668	0.011119	1042.33
SWG	2,362,948	4,833,341	0.007428	1809.92
SG	6,546,614	13,367,955	0.007217	2747.86
GER	21,721,465	44,108,723	0.006560	4799.16

Table 2: Benchmark road networks of various size (SA – Saarbrücken, SL – Saarland, TU – Tübingen, ST – Stuttgart, SWG – South-West-Germany, SG – Southern Germany, GER – Germany). The min/max weights are given in meters.

	$n \times n$	# edges	min w	max w
10a	10×10	180	0	5
10b	10×10	180	1	100
100a	100×100	19,800	0	5
100b	100×100	19,800	1	100
1000a	1000×1000	1,998,000	0	5
1000b	1000×1000	1,998,000	1	100

Table 3: Grid graph benchmarks of various size and with two kinds of weight ranges.

Data Sets

We use real and synthetic data in our evaluation. We extracted several real-world road networks from OSM¹ along with high precision node coordinates from which we interfere the Euclidean lengths of the edges. Table 2 provides an overview of this data set. We observe that edge weights in the range of a few centimeters (or less) as well as several kilometers occur. Note that computing the edge weights, e.g. using the haversine formula for getting the great-circle distance between two points, involves square root calculations and the usage of trigonometric functions. Therefore, the resulting weights might have no finite representation. We will use our developed rounding approaches to make all weights integer. Furthermore, we will look for the smallest data type to store the weights, such that reasonable route planning is still possible. In addition, we generated grid graphs of various size. Edge weights are chosen uniformly at random from some prespecified interval. We differentiate between instances where rounded costs of 0 can occur (and hence the maximum relative error can be unbounded) and instances with all weights ≥ 1 for which the maximum error bound of 2 applies according to Observation 2. Table 3 lists our grid benchmarks.

Baseline Evaluation

We will first investigate the quality of the classic deterministic and randomized rounding schemes. For each of our benchmark graphs, we compute the maximum absolute and relative error for 1,000 example shortest path queries. Moreover, we count the number of paths for which the structure of the optimal path changes after rounding. Table 4 summa-

¹www.openstreetmap.org

	deterministic			randomized		
	A	X	%	A	X	%
SA	23.71	1.006	7.8	26.54	1.010	18.8
SL	27.74	1.003	14.5	40.34	1.003	23.0
TU	35.36	1.003	29.3	60.80	1.002	45.1
ST	58.08	1.001	40.2	77.96	1.003	52.9
SWG	65.51	1.001	42.9	71.99	1.002	54.8
SG	81.25	1.001	55.2	102.75	1.001	63.6
GER	85.61	1.001	70.4	126.43	1.001	81.6
10a	2.86	∞	28.3	6.43	62.978	45.3
10b	2.89	1.081	4.8	4.96	1.109	1.2
100a	13.14	1.222	90.3	18.41	1.293	94.6
100b	8.97	1.008	16.8	11.15	1.008	15.2
1000a	118.17	1.086	99.8	117.20	1.225	99.9
1000b	24.33	1.002	74.8	38.12	1.003	80.4

Table 4: Maximum errors observed in 1,000 random queries. 'A' denotes the maximum absolute error, 'X' the maximum relative error and % the percentage of queries where the structure of the optimal path changed after rounding.

izes the results for all our benchmark graphs when rounding the given weights to integers. The results give the impression that these rounding approaches work well, especially for larger graphs. But in reality, just the percentage of the paths with a large relative error is smaller in large graphs. Hence a random sample of 1,000 queries is unlikely to detect them.

Because of that, we also compute the maximum relative error over all paths of length $k, \dots, 2k - 1$ in the network for several k . This provides us with the real maximum error. The results are shown in Table 5. Only for the small road networks (SA and SL), we get a bound at all. In the larger road networks, both deterministic and randomized rounding lead to paths of length at least k having a rounded weight of 0. Hence the maximum relative error is ∞ . For the grids, due to the edge weights being chosen randomly, this only happened for the 10a instance and the deterministic algorithm. But we observe that the weight range has an enormous impact on the result quality. For 10a, the randomized strategy produces a maximum relative error of over 62, while for all grids of type b we know, that it can be no more than 2. Also the absolute error and the number of affected paths are much smaller for the grids of type b with all edge weights ≥ 1 .

ILP Solutions

Ideally we would like to compare with an optimum solution. Unfortunately, solving our ILP formulation turns out to be too expensive even on the SA dataset. So additionally we extracted a small part ($n = 1000$ nodes, $m = 1982$ edges) of the SA dataset evaluating deterministic and greedy rounding in comparison to ILP generated upper and lower bounds. The lower bound was determined by the largest X value where we could certify infeasibility of the ILP; the upper bound likewise was determined by the smallest X where our ILP could still be shown to be feasible within a reasonable time frame. See Table 6 for the results. For $k = 6$ solving the ILP for values of X close to the optimal X value

k		3	4	6	8	12
SA	d	1.390	1.362	1.332	1.249	*1.118
	r	1.502	1.430	1.336	1.264	*1.135
SL	d	1.390	1.362	1.332	1.270	*1.243
	r	1.685	1.531	1.304	1.232	*1.174
others	d	∞	∞	∞	∞	∞
	r	∞	∞	∞	∞	∞
10a	d	2.169	1.727	1.571	*1.317	*1.190
	r	∞	2.978	2.104	*1.536	*1.599
10b	d	1.036	1.022	1.020	*1.016	*1.012
	r	1.127	1.052	1.037	*1.039	*1.031
100a	d	∞	∞	∞	*2.799	*1.801
	r	∞	∞	∞	*3.205	*1.887
100b	d	1.144	1.109	*1.048	*1.030	*1.020
	r	1.208	1.136	*1.078	*1.052	*1.024
1000a	d	∞	∞	∞	∞	**2.103
	r	∞	∞	∞	∞	**2.401
1000b	d	*1.286	*1.151	**1.064	**1.033	**1.023
	r	*1.451	*1.248	**1.082	**1.052	**1.034

Table 5: Maximum relative error for paths with a hop distance longer than k for r(andomized) and d(eterministic) rounding. Entries with a * are only based on all shortest (instead of all) paths in the network of hop length $k, \dots, 2k-1$, as enumerating all paths was infeasible. Entries with ** are only based on a subset of these shortest paths.

	k	3	4	6
small	deterministic	1.212	1.145	1.104
	greedy	1.099	1.073	1.060
	ILP upper bound	1.064	1.055	1.088
	ILP lower bound	1.058	1.040	(1.000)

Table 6: Comparison of rounding schemes with ILP lower (and upper) bounds. For $k = 6$, a meaningful lower bound using the ILP formulation could not be computed within reasonable time.

took too much time. For $k = 3$ and $k = 4$, though, our ILP lower bound shows that the greedy rounding approach gets reasonably close to the optimum whereas the deterministic rounding scheme fares considerably worse (this is probably also true for larger k). The more advanced ILP formulation for preserving the structure of optimal paths unfortunately is still too inefficient to be solved on anything larger than miniscule toy graphs.

Heuristic Solutions

Next, we evaluate the performance of our greedy heuristic for several choices of k .

We use three variants, depending on the graph size and k . If possible, we extract all paths of length $k, \dots, 2k-1$ in the network and use them as basis for our heuristic. Unfortunately, this is not always possible. For example, for the TU graph and $k = 8$ this would be more than 159 million paths each of length at least 8, for GER with $k = 3$ there are more than 256 million paths already, which making calculations on exceeds the capability of our used hardware. If we consider only shortest paths, the number of paths for TU and $k = 8$ reduces to about 25 million. Hence we can tackle larger instances when restricting ourselves to shortest

paths. But note, that the a posteriori approximation guarantee then only holds for paths for which the optimal path structure does not change after rounding. And even extracting all shortest paths of a certain length is not always possible. In that case, we permuted the vertices of the graph randomly, and then extracted the shortest paths starting at a vertex (in the permuted order) until some upper bound on the number of paths was reached. We used a bound of 100 million in our experiments. For edges not contained in any extracted path, we use deterministic rounding as a fall back. Then, no a posteriori guarantee can be given, but our evaluations show that our heuristic outperforms the baseline approaches even with this restricted input.

Road Networks The results for our road network benchmarks are shown in Table 7. Naturally, the larger k , the better the error bound B gets. Comparing the bound results to those of our baselines in Table 5, we observe that the greedy heuristic produces significantly smaller maximum relative errors for SA and SL. For example, for SA and $k = 8$, the deterministic approach has a maximum error of 1.249 and the error produced by the heuristic is only 1.131. For all benchmarks larger than SL, neither of the baselines lead to a finite error bound. But with the heuristic, we always get a finite error bound. For small k , this bound tends to be large. This is due to short paths with very small edge weights only. If all such weights would be rounded to 0, the error bound would be ∞ . Our heuristic is explicitly designed to not let this happen; at least the last edge considered on such a path will be rounded to 1 as this minimizes the maximum error. But the relative error in such a case might still be huge, e.g. over 12 for SG and $k = 3$. For larger k , the bound decreases quickly and we always end up with a maximum relative error bound smaller than 2.

Also considering the 1,000 sample queries, the heuristic leads to smaller absolute and relative errors than the baselines (compare Table 7 and Table 4) for k large enough. For SA and $k = 6$, the maximum observed absolute error is less than half the one observed after deterministically weight rounding and the percentage of affected paths is reduced from 7.8% to 2%. For larger graphs, the percentage of affected graph also becomes quite large when using our greedy heuristic. In up to about 70% of the queries, the structure of the optimal paths changed. But on the one hand, the observed weight errors are nevertheless very small and on the other hand often only small sections of the optimal paths changed after rounding. Computing the percentage of edges that were in an optimal path before rounding, but are no longer after rounding, we end up with only about 5% also for the larger graphs.

Grid Graphs For the grid benchmarks, the results of the evaluation of the greedy heuristic are shown in Table 8. In general, we observe the same trend as for road networks: The heuristic leads to smaller bounds as well as absolute and relative errors in a set of sample queries. One notable exception is the 10a instance. Here, the maximum relative error observed in the example queries is ∞ for the heuristic. Note that queries in this sample are not restricted to have optimal paths of hop length at least k . And in a 10×10

k		3	4	6	8	12
SA	B	1.258	1.2021	1.147	1.131	*1.073
	A	14.18	10.09	10.44	16.61	*7.42
	X	1.003	1.002	1.002	1.004	*1.001
	%	9.4	8.0	2.0	8.8	*3.2
SL	B	1.258	1.202	1.166	1.156	*1.073
	A	18.76	17.14	13.22	11.86	*11.62
	X	1.001	1.001	1.001	1.001	*1.001
	%	8.6	8.2	6.1	8.8	*7.6
TU	B	2.595	2.103	1.760	*1.407	**1.071
	A	20.86	21.40	18.08	*19.03	**18.4
	X	1.001	1.001	1.000	*1.001	**1.000
	%	32.8	29.2	27.2	*27.2	**29.1
ST	B	2.014	1.907	1.496	*1.322	**1.175
	A	30.02	21.84	18.32	*14.06	**14.72
	X	1.002	1.001	1.001	*1.000	**1.000
	%	34.0	37.6	35.2	*29.6	**31.4
SWG	B	2.595	2.103	*1.602	**1.084	**1.094
	A	43.06	27.73	*30.18	**24.28	**29.92
	X	1.000	1.001	*1.001	**1.000	**1.000
	%	41.6	26.8	*24.4	**39.0	**33.1
SG	B	12.463	*1.906	**1.215	**1.167	**1.088
	A	51.51	*25.07	**42.14	**34.92	**38.74
	X	1.001	*1.000	**1.000	**1.001	**1.000
	%	52.4	55.3	**54.0	**38.7	**52.7
GER	B	**55.362	**1.580	**1.504	**1.182	**1.118
	A	**61.42	**62.39	**65.22	**65.99	**48.79
	X	**1.000	**1.000	**1.000	**1.001	**1.000
	%	**73.1	**70.2	**66.7	**68.0	**71.4

Table 7: Experimental results for the road networks using our greedy heuristic on a set of paths P . Entries without a * are obtained from P containing all paths, entries with a * are obtained from P containing only shortest paths (as all paths could not be stored anymore), and entries with ** of a subset of these shortest paths to make the heuristic feasible. B denotes the maximum relative error for P . A , X and $\%$ indicate the maximum absolute and relative error as well as the percentage of affected optimal path structures for 1,000 random queries.

k		3	4	6	8	12
10a	B	1.457	1.720	1.257	*1.145	*1.103
	A	2.48	2.26	2.76	*1.93	*1.42
	X	2.053	∞	∞	∞	∞
	%	40.3	38.8	40.8	23.6	35.2
10b	B	1.023	1.022	1.017	*1.009	*1.005
	A	2.32	3.08	2.92	*2.03	*1.77
	X	1.049	1.026	1.044	*1.171	*1.086
	%	0.8	0.4	0.4	*0.4	*0.4
100a	B	8.895	2.372	*1.621	*1.372	*1.237
	A	9.05	10.28	*4.65	*3.34	*2.69
	X	1.087	1.130	*1.121	*1.110	*1.070
	%	95.6	94.0	*91.6	*91.6	*87.2
100b	B	1.071	1.056	*1.034	*1.023	*1.011
	A	10.88	7.44	*4.17	*3.59	*3.51
	X	1.005	1.011	*1.003	*1.003	*1.008
	%	20.4	19.6	*9.2	*18.0	*19.2
1000a	B	*29.821	*15.015	**1.809	**1.535	**1.226
	A	*56.39	*40.46	**24.16	**28.66	**29.36
	X	*1.045	*1.041	**1.025	**1.023	**1.043
	%	100.0	100.0	**100.0	**100.0	**100.0
1000b	B	*1.146	*1.078	**1.038	**1.016	**1.008
	A	*18.12	*13.98	**13.21	**13.21	**11.62
	X	*1.002	*1.001	**1.001	**1.002	**1.001
	%	*63.6	*66.1	**63.0	**68.1	**71.5

Table 8: Experimental results for the grid graph benchmarks using the greedy heuristic.

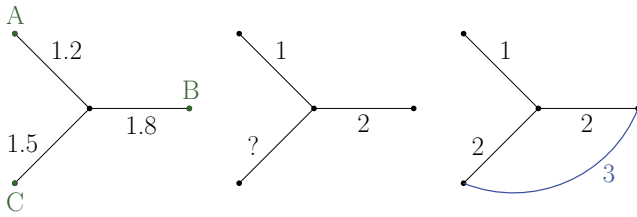


Figure 5: The goal is to round the weights in the leftmost image, such that the relative error on the paths $A \leftrightarrow B$, $B \leftrightarrow C$ and $C \leftrightarrow A$ is minimized. The smallest ratio would be achieved if all rounded path weights would equal 3. But the image in the middle illustrates that this is impossible, as at least one of the paths has a weight of 2 or 4 after rounding. However, if we allow to insert an additional edge $\{B, C\}$ with a weight of 3, the optimum can be realized.

grid, performing 1,000 random queries leads automatically to at least some of the optimal paths being short (sometimes containing only a single edge). For these paths, the bound B does not apply, of course. So while the absolute error is very small, the relative error is unbounded. But for the larger instances, the maximum relative error gets remarkably small. Still, the grids of type 'a' (with weights ≤ 1 being allowed) are more difficult to handle than those of type 'b'. The relative error is an order of magnitude larger, and almost all optimal path structures are affected by the rounding. For 100b, for example, less than 10% of the paths are affected when choosing $k = 6$.

We conclude that the greedy heuristic is suitable to get smaller error bounds than the baselines, by taking the structure of paths explicitly into account.

Conclusions and Future Work

We have shown that paying attention to consistently rounding edge weights in networks is worthwhile. We can achieve much smaller maximum relative errors compared to deterministic or randomized rounding, and optimal path structures are better preserved. Not only does rounding edge weights save memory, but having integer edge weights also allows to use more efficient algorithms and data structures. For example, sorting operations on the edges can now be carried out with RadixSort, taking only linear time. Also shortest paths could be computed with breadth first search instead of Dijkstra, by subsampling all edges. This would only blow up the graph by a factor of at most $\max_{e \in E} w'(e)$.

Natural extensions for future work are considering also negative edge costs and other rounding rules beside rounding to the next smaller or larger integer (but instead have some range Δ). The problem of accumulating rounding errors could also be mitigated by allowing additional operations besides simple rounding. For example, inserting additional edges might be an operation that could help to reduce the maximum rounding error further as indicated in Figure 5.

Also, new heuristics are needed to get an a posteriori maximum relative error guarantee also for larger graphs (and

larger k).

References

- Asano, T.; Katoh, N.; Obokata, K.; and Tokuyama, T. 2002. Matrix rounding under the l_p -discrepancy measure and its application to digital halftoning. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, 896–904. Society for Industrial and Applied Mathematics.
- Asano, T.; Matsui, T.; and Tokuyama, T. 2000. Optimal roundings of sequences and matrices. *Nordic J. of Computing* 7(3):241–256.
- Kayal, N., and Saha, C. 2011. On the sum of square roots of polynomials and related problems. In *Computational Complexity (CCC), 2011 IEEE 26th Annual Conference on*, 292–299.
- Raghavan, P., and Tompson, C. D. 1987. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica* 7(4):365–374.