A Competitive Strategy for Distance-Aware Online Shape Allocation

Sándor P. Fekete* Nils Schweer* Jan-Marc Reinhardt*

Abstract

We consider the following online problem, motivated by grid computing: Given an $N \times N$ square S, and a sequence of numbers n_i with $\sum_{j=0}^{i} n_j \leq N^2$; at each step i, select a region C_i of previously unassigned area n_i in S. The objective is to minimize the maximum average Manhattan distance between points from the same set C_i. We present a competitive online strategy, based on a thorough analysis of space-filling curves; for continuous shapes, we prove a factor of 1.8092, and 1.7849 for discrete shapes.

1 Introduction

Many optimization problems deal with allocating point sets to a given environment. Frequently, the problem is to find compact allocations, such that points from the same set are closely together. One natural measure is to consider the average distance between points. A practical example occurs in the context of grid computing, where one needs to assign a sequence of jobs i, each requiring n_i processors, to a subset C_i of nodes of a square grid, such that the average communication delay between nodes of the same job is minimized; as this delay corresponds to the number of grid hops [6], this motivates the geometric task of finding subsets with a small average Manhattan distance.

Even in an offline setting without any occupied nodes, finding an optimal allocation for a single set of size n_i is not an easy task; the results are typically "round" shapes. If a whole sequence of sets have to be allocated, packing such shapes onto the grid will produce gaps, causing later sets to become scattered, and thus lead to extremely bad average distances. Even packing rectangular shapes is not a remedy, as the resulting packing problem is NP-hard, and scattered allocations may still occur.

In this paper, we give a first algorithmic analysis for the *online* problem. Using an allocation scheme based on a space-filling curve, we establish competitive factors of 1.8092 and 1.7849 for minimizing the maximum average Manhattan distance within an allocated set.

*Algorithms Group, Braunschweig Institute of Technology, Germany, {s.fekete,n.schweer,j-m.reinhardt}@tu-bs.de

Related Work The problem of finding the "optimal shape of a city", i.e., a shape of unit area that minimizes the average distance, was first considered by Karp, McKellar, and Wong [4]; independently, Bender, Bender, Demaine, and Fekete [1] showed that this shape can be characterized by a differential equation for which no closed form is known. For the case of a finite set of n points that needs to be allocated to a grid, Demaine et al. [3] showed that there is an $O(n^{7.5})$ dynamic-programming algorithm. In an offline setting in the presence of occupied points, Bender et al. [2] gave some simple 1.75-approximation algorithms, and a polynomial-time approximation scheme. Space-filling curves for processor allocation have been used before [6]; in particular, Wattenberg [10] has proposed an allocation scheme similar to ours, for purposes of minimizing the maximum *diameter* of an allocated shape. However, neither paper has yielded algorithmic results for our problem, and no constant competitive factor was proven.

2 Preliminaries

We examine the problem of selecting shapes from a square, such that the maximum average L_1-distance of the shapes is minimized. We first formulate the problem more precisely.

Definition 1 *A* city *is a shape in the plane with fixed size. For a city C of size n we call*

$$c(C) = \frac{1}{2} \iiiint_{(x,y),(u,v)\in C} |x-u|+|y-v| \,\mathrm{d}v \,\mathrm{d}u \,\mathrm{d}y \,\mathrm{d}x$$

the Manhattan distance between all pairs of points in C *and*

$$\phi(C) = \frac{2\,c(C)}{n^{5/2}}$$

the ϕ-value *or* average distance *of* C.

We divide by $n^{2.5}$ to get a dimensionless measure for the average distance, see [1].

The problem For a sequence $n_1, n_2, \ldots, n_k \in \mathbb{R}^+$ with $\sum_{i=1}^{k} n_i \leq 1$, cities $C_1, C_2, \ldots, C_k$ are to be chosen from the unit square, such that $\max_{1\leq i\leq k} \phi(C_i)$ is minimized.

Although it has not been proven, the problem is assumed to be NP-hard, see [8]; if we restrict city shapes to be rectangles, there is an immediate reduction from

This is an extended abstract of a presentation given at EuroCG 2011. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear in a conference with formal proceedings and/or in a journal.

deciding whether a set of squares can be packed into a larger square [5]. (A special case arises from considering integers, which corresponds to choosing grid locations.) On the one hand, to make things a little easier, we only consider sizes n_i which are larger than a known lower bound ε. On the other hand, our approximation works online, i.e., we choose the cities in a specified order, and no changes are made to previously allocated cities.

There are lower bounds for $max_{1\leq i\leq k}\phi(C_i)$ that generally cannot be achieved by any algorithm. One important result is the following theorem.

Theorem 1 *Let C be any city. Then $\phi(C) \geq 0.650245$.*

A proof can be found in [1]. For $n_1 = 1$ any algorithm must select the whole unit square, thus 2/3, the ϕ-value of a square, is a lower bound. We will discuss approaches for a better bound in the end.

3 An Algorithm

While long and narrow shapes tend to have large ϕ-values, shapes that fill large parts of an enclosing rectangle with similar width and height usually have better average distances. Our approach uses the space-filling Hilbert curve to yield a provably constant competitive factor.

First, we divide the unit square into a grid consisting of $2^r \times 2^r$ cells, such that the size of each cell of the grid is equal to or less than the lower bound of the n_i. The Hilbert curve is based on a recursive construction scheme and becomes space-filling for infinite repetition of said scheme [7]. For a finite number r of repetitions, the curve traverses the used grid. For $1 \leq r \leq 3$, the curve is shown in figure 1. Thus, the Hilbert curve provides an order for the cells of the grid.

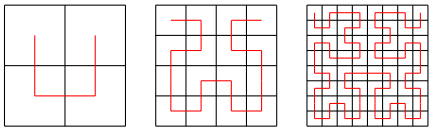

Figure 1: Hilbert curve with $1 \leq r \leq 3$

Furthermore, we denote the size of each cell by $c = 4^{-r}$, and the cell in row j and column k after r iterations by $E_r^2[j][k] = [2^{-r}(j-1), 2^{-r}j] \times [2^{-r}(k-1), 2^{-r}k]$. For the sake of concise presentation of our analysis within this short abstract, we assume that every input n_i is an integral multiple of c; in general, we can proceed to make the grid self-refining. The description of the algorithm is simple: for every input n_i we choose the next n_i/c cells traversed by the Hilbert curve as the city C_i, starting in the upper left corner of the grid.

A formal implementation of the curve can be done using text-rewriting rules, such as the ones in [9].

4 Analysis

Our analysis of the algorithm focuses on two points: describing a way to systematically find worst cases of a specified size and finding a competitive factor.

Definition 2 *We denote by W_i a city with i cells (or i whole $E_l^2[j][k]$ for $l \neq r$) that is a worst-case output of our strategy for cities of size i.*

As all cities generated by our strategy consist of an integral number of cells, the ϕ-value of a produced city can be calculated considering only the distances and count of the cells occupied in the grid. Thus, the W_i can be found by simply comparing the distances of all i-tuples of cells on the grid, which are successively traversed by the Hilbert curve. This leaves the question of how large the grid has to be chosen to contain a city W_i traversed by the curve.

Lemma 2 *For $r = \lceil \log_2(\lceil i/4 \rceil + 1) \rceil + 2$ the Hilbert curve traverses a city W_i.*

We do not give a proof here. It can be done using the text-rewriting rules from [9] to construct a formal grammar and show via induction that after a certain number of substitutions, every possible sequence of symbols of a particular length has been generated, i.e., the sequence corresponding to the city with the greatest ϕ-value must have been produced as well. We used this approach to determine the shapes and ϕ-values of the W_i for $i \leq 65$. The average distances for $16 \leq i \leq 63$ can be seen in the fourth column of table 1; the worst cases among the examined ones are W_{56} and W_{14}, which have the same shape, shown in figure 2.

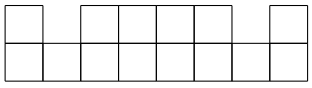

Figure 2: W_{14}

The worst cases can also be used to find an upper bound for the maximum average distance and ultimately give a competitive factor.

Lemma 3 *For every $0 \leq l \leq r$ the $E_l^2[j][k]$ are traversed by the Hilbert Curve in a specific order, i.e., the curve does not only pose an order on the cells of the grid, but on each set of the $E_l^2[j][k]$.*

Again, we do not give a formal proof here; the claim follows directly from the definition of the Hilbert curve, as its construction scheme is invoked recursively for sub-squares of the unit square.

i	$c^*(W_i)$	$c_{city}(i)$	$\phi(W_i)$	$\Phi(W_{i+2})$
16	$434\,^2/_3$	$338\,^2/_3$	0.8490	1.1764
17	$512\,^2/_3$	$396\,^1/_3$	0.8605	1.1497
18	$602\,^1/_3$	$457\,^1/_3$	0.8764	1.1174
19	685	$522\,^1/_3$	0.8706	1.1058
20	768	$591\,^2/_3$	0.8587	1.1098
21	870	663	0.8610	1.0903
22	$992\,^2/_3$	$749\,^1/_3$	0.8745	1.0713
23	$1101\,^2/_3$	$839\,^1/_3$	0.8685	1.0424
24	1216	933	0.8619	1.0150
25	$1322\,^1/_3$	$1032\,^1/_3$	0.8463	0.9777
26	1432	1134	0.8309	0.9701
27	$1527\,^2/_3$	1249	0.8066	0.9785
28	1672	$1365\,^1/_3$	0.8061	0.9864
29	$1853\,^1/_3$	1492	0.8184	0.9773
30	2046	$1622\,^2/_3$	0.8301	0.9710
31	2213	$1759\,^2/_3$	0.8272	0.9726
32	$2393\,^1/_3$	$1898\,^2/_3$	0.8263	0.9791
33	2602	2057	0.8319	0.9735
34	$2835\,^2/_3$	$2216\,^2/_3$	0.8414	0.9691
35	3045	2384	0.8403	0.9712
36	3266	$2554\,^2/_3$	0.8400	0.9772
37	$3519\,^1/_3$	$2727\,^2/_3$	0.8453	0.9726
38	$3799\,^1/_3$	$2921\,^2/_3$	0.8536	0.9682
39	$4049\,^2/_3$	$3117\,^2/_3$	0.8527	0.9570
40	$4309\,^1/_3$	3322	0.8517	0.9463
41	4545	$3530\,^1/_3$	0.8445	0.9307
42	4788	3749	0.8376	0.9214
43	5009	3976	0.8262	0.9306
44	$5266\,^2/_3$	$4205\,^1/_3$	0.8202	0.9393
45	$5641\,^1/_3$	$4456\,^2/_3$	0.8306	0.9393
46	$6031\,^1/_3$	4712	0.8405	0.9395
47	$6379\,^2/_3$	$4970\,^1/_3$	0.8425	0.9439
48	$6741\,^1/_3$	5234	0.8446	0.9505
49	$7147\,^1/_3$	$5507\,^1/_3$	0.8505	0.9512
50	$7586\,^1/_3$	5788	0.8583	0.9516
51	7993	$6076\,^1/_3$	0.8606	0.9559
52	$8411\,^1/_3$	6368	0.8628	0.9620
53	8878	$6691\,^2/_3$	0.8683	0.9619
54	$9379\,^1/_3$	$7017\,^1/_3$	0.8754	0.9617
55	9835	$7352\,^1/_3$	0.8768	0.9569
56	10304	7690	0.8781	0.9522
57	10733	$8033\,^2/_3$	0.8751	0.9445
58	$11173\,^1/_3$	$8384\,^2/_3$	0.8723	0.9372
59	$11583\,^2/_3$	$8749\,^1/_3$	0.8665	0.9268
60	$12005\,^1/_3$	$9117\,^1/_3$	0.8610	0.9225
61	12391	$9506\,^1/_3$	0.8527	0.9232
62	12862	$9904\,^2/_3$	0.8499	0.9201
63	13415	$10305\,^1/_3$	0.8517	0.9175

Table 1: Total and average distances for cities W_n allocated according to our strategy, as well as the optimal values $c_{city}(n)$ according to [3].

Lemma 4 *Let C be a city generated by our strategy with size $n \leq k\,4^j\,c$ for $j \in \{0, 1, \ldots, r\}, k \in \mathbb{N}$. Furthermore, let W be a city W_{k+1} of size $(k+1)\,4^j\,c$. Then $c(C) \leq c(W)$ holds.*

Proof. C cannot contain parts of more than $k+1$ of the sub-squares $E^2_{r-j}[p][q]$. Otherwise we would have a contradiction to Lemma 3, as it would contain fewer than $k-1$ complete sub-squares, thus implying that they are not traversed in order.

Consequently, C can be bounded by a city X consisting of $k+1$ sub-squares of size $4^j\,c$, and $c(C) \leq c(X)$ holds. As there is no city consisting of $k+1$ sub-squares with a greater ϕ-value than W_{k+1}, $c(X) \leq c(W)$ holds as well. Combining both inequalities yields $c(C) \leq c(W)$. □

Theorem 5 *Our strategy guarantees* $\max_{1 \leq i \leq k} \phi(C_i) \leq 1.1764$.

Proof. For $r \leq 2$ the only possible inputs n are $c \leq n \leq 16c$. In this case, $\phi(C)$ can be bounded by $\max_{1 \leq i \leq 16} \phi(W_i) = \phi(W_{14}) \approx 0.8781$.

In the following consider $r > 2$ and $4^j c < n \leq 4^{j+1}c$ for $j \in \{2, \ldots, r-1\}$.

For each of the cases $l\,4^{j-2}c < n \leq (l+1)\,4^{j-2}c$ with $l \in \{16, 17, \ldots, 63\}$ we can use Lemma 4 to get a city W of size $(l+2)\,4^{j-2}\,c$ and shape W_{l+2}, such that $c(C)$ is less than or equal to $c(W)$.

Thus, we get the following inequality:

$$\phi(C) \leq \frac{2\,c(W)}{(l\,4^{j-2}c)^{5/2}}$$

Because of the definition of the ϕ-value, this yields

$$\frac{\phi(W_{l+2})((l+2)\,4^{j-2}c)^{5/2}}{(l\,4^{j-2}c)^{5/2}} = \phi(W_{l+2})\left(1+\frac{2}{l}\right)^{5/2}$$

We denote $\phi(W_{l+2})\left(1+\frac{2}{l}\right)^{5/2}$ by $\Phi(W_l)$ and list the values of $\Phi(W_l)$ for $16 \leq l \leq 63$ in the fifth column of table 1. Therefore, $\phi(C) \leq 1.1764$ holds. □

Corollary 6 *Our strategy achieves a competitive factor of 1.8092.*

Proof. According to Theorem 1, no algorithm can guarantee a better ϕ-value than 0.650245. Our strategy yields an upper bound of 1.1764. This results in a factor of $1.1764/0.650245 \approx 1.8092$. □

5 Discrete Point Sets

Our above analysis relies on continuous weight distributions, which imply the lower bound on ϕ-values stated in Theorem 1. This does include the case of integer values n_i. However, as discussed in [3], considering discrete weight distributions may allow lower average distances; e.g., a single point has an average distance of 0. As a consequence, *towns* (subsets of the integer grid) have lower average distances than cities of the same total weight. However, we still get a competitive ratio for the case of online towns.

Definition 3 ([3]) *An n-town T is a subset of n points in the integer grid. Its* normalized average Manhattan distance *is*

$$\phi(T) = \frac{2c(T)}{n^{5/2}} = \frac{\sum_{s\in T}\sum_{t\in T}\|s-t\|_1}{n^{5/2}}$$

Theorem 7 *For n-towns, our strategy guarantees a competitive factor of at most 1.7849 for the ϕ-value.*

Proof. Analogous to Theorem 5, we consider the values up to $n = 64$, and show that the worst case is attained for $i = 16$, which yields an upper bound of 1.123. (The analog to Table 1 is omitted for lack of space.) For a lower bound, the general value of 0.650245 for ϕ-values cannot be applied, as discrete point sets may have lower average distance. Instead, we verify that the ratio of achieved ϕ to optimal ϕ for $n \leq 64$ is less than 1.7849; for $65 \leq n \leq 80$, Table 1 in [3] allows us to verify that $\phi \geq 0.629171$. For $n \geq 81$, we make use of equation (5), p. 89 of [3] to show the same lower bound of 0.629171; details are slightly involved and omitted for lack of space. Overall, we get a competitive ratio bounded by upper divdided by lower bound, i.e., 1.7849. □

6 Lower Bounds

We demonstrate that there are non-trivial lower bounds for a competitive factor by considering the online scenario for towns.

Theorem 8 *No online strategy can guarantee a competitive factor below $\frac{64}{\sqrt{5}^5} = 1.144866....$*

Proof. Consider a 3x3 square, and let $n_1 = 4$. If the strategy allocates a 2x2 square (for a total distance of 8), then $n_2 = 5$, and the resulting L-shape has a total distance of 20 and a ϕ-value of $40/5^{2.5} = 0.715541...$ Allocating the first town with an L-shape of total distance 10 results in $\phi = 20/32 = 0.625$, and the second with a total distance of 16, or $\phi = 32/5^{2.5} = 0.572433...$

If instead, the first town is allocated different from a square, the total distance is at least 10, and $\phi \geq 20/32$; then $n_2 = n_3 = n_4 = n_5 = n_6 = 1$, and an optimal strategy can allocate the first town as a 2x2 square, with $\phi = 0.5$. This bounds the competitive ratio, as claimed. □

7 Conclusions

The offline problem (where all n_i are given in advance) is interesting in itself. A simple lower bound for a guaranteed maximum is 2/3, as that is the average distance of the whole square. For the case $n_1 = n_2 = 1/2$ we conjecture an optimum of $\sqrt{2}/2$, which we could prove for the special case of divisions using a straight line across the center. We believe the global worst case is attained for $n_1 = n_2 = n_3 = 1/3$.

Our description of the competitive factor of 1.8092 uses the assumption that there is a lower bound ε for the input and a smallest common denominator $c \leq \varepsilon$ for all input numbers. Discarding this assumption is possible, but requires some more tedious analysis.

In principle, further improvement of the established factors could be achieved by replacing the anlysis from $n = 16, \ldots, 64$ by $n = 65, \ldots, 256$. However, the highest known optimal ϕ-values are for $n = 80$ using the $O(n^{7.5})$ algorithm of [3], so the involved computational effort promises to be large.

Acknowledgments

We thank Bettina Speckmann for helpful comments and pointing out reference [10].

References

[1] C. M. Bender, M. A. Bender, E. D. Demaine, and S. P. Fekete. What is the optimal shape of a city? *J. Physics A: Math. Gen.*, 37(1):147–159, 2004.

[2] M. A. Bender, D. P. Bunde, E. D. Demaine, S. P. Fekete, V. J. Leung, H. Meijer, and C. A. Phillips. Communication-Aware Processor Allocation for Supercomputers: Finding Point Sets of Small Average Distance. *Algorithmica*, 50(2):279–298, 2008.

[3] E. D. Demaine, S. P. Fekete, G. Rote, N. Schweer, D. Schymura, and M. Zelke. Integer point sets minimizing average pairwise L1 distance: What is the optimal shape of a town? *Computational Geometry: Theory and Applications*, 40:82–94, 2011.

[4] R. M. Karp, A. C. McKellar, and C. K. Wong. Near-Optimal Solutions to a 2-Dimensional Placement Problem. *SIAM J. Comp.*, 4(3):271–286, 1975.

[5] J. Y.-T. Leung, T. W. Tam, C. S. Wing, G. H. Young, and F. Y. Chin. Packing squares into a square. *J. Parallel Distrib. Comput.*, 10(3):271–275, 1990.

[6] V. J. Leung, E. M. Arkin, M. A. Bender, D. P. Bunde, J. Johnston, A. Lal, J. S. B. Mitchell, C. A. Phillips, and S. S. Seiden. Processor Allocation on Cplant: Achieving General Processor Locality Using One-Dimensional Allocation Strategies. In *Proc. IEEE Int. Conf. Cluster Comp.*, pages 296–304. 2002.

[7] H. Sagan. *Space-Filling Curves*. Springer, 1994.

[8] N. Schweer. *Algorithms for Packing Problems*. PhD thesis, TU Braunschweig, 2010.

[9] R. Siromoney and K. Subramanian. Space-filling curves and infinite graphs. In *Graph-Grammars and Their Application to Computer Science*, volume 153 of *LNCS*, pages 380–391, Berlin, 1983. Springer.

[10] M. Wattenberg. A note on space-filling visualizations and space-filling curves. In *Proc. IEEE Symp. Information Visualization*, pages 181–186, 2005.