Venor, Inc.

Case No. 813022601

INVENTION DISCLOSURE FORM

1. **TITLE OF INVENTION:** K-Grid for Data Visualization and Data Mining with Hill-Climbing Heuristic to Optimize Permutation Search

2. **INVENTOR(S)**

A. Name of first inventor: Rudi Cilibrasi

Work Number: 510-735-6504 Extension

Fax Number

Home Mailing Address: 244 W. Duane Ave. / Sunnyvale, CA 94085

Home Number: 408-738-9254

Citizenship: USA

B. Name of second inventor

Work Number Extension

Fax Number

Home Mailing Address

Home Number

Citizenship

C. Name of third inventor

Work Number Extension

Fax Number

Home Mailing Address

Home Number

Citizenship

(PLEASE ADD ADDITIONAL INVENTORS ON SEPARATE SHEET)

Venor, Inc.

Case No. 813022601

3. CONCEPTION OF INVENTION:

A. Was the invention made in the U.S.? Yes X No

B. Date and location of first drawing(s) 02/26/2013

C. Date and location of first written description 12/06/2012

D. Are there witness signatures? Yes; invoice #201212061 was paid via check and mentions K-Grid

4. CONSTRUCTION OF DEVICE:

A. Was prototype made? Yes

B. If so, where can prototype be found? Source code transferred to Venor, Inc. via email and repository.

C. If so, made by whom? Rudi Cilibrasi

D. If so, on what date were tests completed? 12/06/2012

E. Are there witnesses? Yes; invoice #201212061 was paid via check and mentions K-Grid

5. DISCLOSURE OF INVENTION TO OTHERS

A. Was invention disclosed to others? Yes :X No

B. If so, to whom and on what date? Only Venor employees under NDA: Ali Golshan, Wenfeng Wang

C. If so, were Non-Disclosure Agreements signed? Yes

D. If not disclosed to others, do you have any plans to disclose the invention?

6. OFFER FOR SALE OR SALE OF PRODUCT OR SERVICE INCORPORATING THE INVENTION:

A. Was a product/service offered for sale, advertised or sold? Yes No: X

B. If so, to whom and on what date?

C. If not, any plans to market/sell the product/service?

7. **USE OF A PRODUCT OR SERVICE INCORPORATING THE INVENTION:**

A. Is the product/service presently being used? Yes ________________ No: X______________________

B. If so, how and where? ____

____

C. If not, any plans to use the product/service? Yes; currently research prototype. It may eventually enable advanced data analysis.____

8. **PUBLICATION OF THE INVENTION:**

A. Are there any publications related to the invention? Yes _______________ No: X______________________

B. If so, published where and when? ____

C. If so, please provide a copy of each publication.

D. If not, any plans to publish the invention? : Not yet but in consideration.______________________________

____

9. **OTHER ISSUED PATENTS OR APPLICATIONS ON INVENTION:**

A. Was an application filed, in U.S. or other country, on the invention? Yes _____________ No: X____________

B. If so, in what country? ____

C. If so, please provide a copy of each application or issued patent.

10. **LIST ALL KNOWN RELATED PRINTED PUBLICATIONS, ISSUED PATENTS, PATENT APPLICATIONS:**

A. ____

B. ____

C. ____

D. Please provide a copy of each listed reference.

Venor, Inc.

Case No. 813022601

11. **WAS INVENTION:**

A. Conceived during performance of a Government contract? Yes ________ No: X________

B. Constructed during performance of a Government contract? Yes ________ No: X________

C. Tested during performance of a Government contract? Yes ________ No: X________

D. Contract Number ____________________
(Give Full Contract Number)

12. **ON A SEPARATE SHEET, DESCRIBE WHAT YOU KNOW EXISTS IN THE PRIOR ART.**

13. **ON SEPARATE SHEETS, PROVIDE DRAWINGS THAT ILLUSTRATE YOUR INVENTION, AND PROVIDE ANY FURTHER TEXT THAT MAY BE HELPFUL. PLEASE PROVIDE SUFFICIENT DRAWINGS AND TEXT TO ENABLE SOMEONE TO MAKE AND USE YOUR INVENTION IN THE BEST WAY YOU KNOW. IT IS BETTER TO BE SPECIFIC AND DETAILED IN YOUR EXPLANATION.**

Please be reminded that the invention(s) described herein are the intellectual property of Venor, Inc. Accordingly, we remind you to maintain these invention(s) in confidence outside of Venor, Inc.

Rudi Cilibrasi
Inventor

2-26-2013
Date

Rudi Cilibrasi [CTO]
Manager (Rudi Cilibrasi)

2-26-2013
Date

K-Grid for Data Visualization and Data Mining

Rudi Cilibrasi

March 9, 2014

Abstract

We introduce a new algorithm in data visualization called *K-Grid* as an assignment optimization procedure with wide applicability. K-Grid takes as input a list of object labels and an associated distance matrix. K-Grid searches to find an optimal objective score for a specific assignment of objects to placements over a lattice of points with integer coordinates. We describe an implementation using a hill-climbing heuristic to optimize the permutation search for information theoretic objective functions to allow for larger-scale data-mining applications.

1 Introduction

Data visualization and data mining seek to address the question of how we might find patterns in data without humans first forming specific hypotheses for testing. In data mining, we use automatic machine learning and high-speed computers to make machines do the hard work of generating testable hypotheses as well as the statistical calculations required to test each of them. By iterating over very many interesting hypotheses, some may be found that are accurate and useful via automatic statistical significance testing given a suitable probability model. When these hypotheses range over topological spaces and utilize information theory as the basis for the model the approach is called information topology.

2 Technical Preliminaries and Notation

We write $\mathbb{Z}$ to represent the set of all integers, and $\mathbb{Z}_+$ to represent the positive integers. Similarly, we understand $\mathbb{Z}_s$ to represent the integers taken modulo s. Vectors are written in bold typeface so that $\boldsymbol{b}$ is understood to be a vector with individual components written as b_i with $i \in \mathbb{Z}_+$ and $i \leq d$ when $\boldsymbol{b}$ has dimension d. We write $\|\boldsymbol{b}\|$ to mean the Euclidean norm of $\boldsymbol{b}$ so that

$$\|\boldsymbol{b}\| \quad = \quad \sqrt{\sum_{i=1}^{d} b_i^2}$$

When we write $|x|$ it means that x is a real number or a set. If x is a set then $|x|$ refers to the cardinality of the set. If x is a real number then $|x|$ means the absolute value of x.

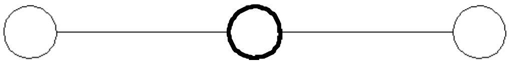

Figure 1: One dimensional local neighborhood showing 2 neighbors, 1 unit distant from a central node.

Figure 2: One dimensional *K-Grid* with 7 nodes.

2.1 K-Grid Shape

A K-Grid takes as input a set of parameters that define its shape. The most important of these is $k \in \mathbb{Z}_+$. This is called the *dimension* of the K-Grid. k is also the dimension of the next parameter $\boldsymbol{s}$ called the *size* of the K-Grid. We assume that $s_i \in \mathbb{Z}$ and $s_i > 3$. Each *location* (or *point*) in the K-Grid is addressed via a k-dimensional coordinate of the form:

$$\prod_{i=1}^{k} \mathbb{Z}_{s_i}$$

We write U to represent the union of all possible locations in the K-Grid.

2.2 K-Grid Topology

We define a *local neighborhood* of a point $\boldsymbol{p}$ (written $L(\boldsymbol{p})$) in the K-Grid to be any point $\boldsymbol{q}$ in the K-Grid such that

$$0 < \|\boldsymbol{p} - \boldsymbol{q}\| \leq \sqrt{k}$$

We distinguish two categories of K-Grid called *wrapped* K-Grid and *non-wrapped* K-Grid. The *non-wrapped* K-Grid is equivalent to normal Euclidean spatial distance. The *wrapped* K-Grid is slightly different because the coordinates are taken modulo s_i, the edges are considered to “wrap around” from one side to another with distance one as in Figures 4,9. This means that all points have an equal number of neighbors at each distance in the wrapped K-Grid. This is in contrast to the non-wrapped K-Grid whose corner, edge, and face locations have fewer neighbors than more central points. This means that non-wrapped K-Grids have a variety of different local neighborhood shapes whereas wrapped K-Grids have only one common shape for all local neighborhoods centered around each point. In the wrapped K-Grid $|L(p)| = 3^k - 1$. For the non-wrapped K-Grid we have only $k \leq |L(p)| \leq 3^k - 1$. See Figures 1,2,3

In either case, the total number of places (distinct coordinate locations) or discrete volume V in the K-Grid is the product of the components of the size vector $\boldsymbol{s}$:

$$V = \prod_{i=1}^{k} s_i$$

k	$\sqrt{k}$	$3^k - 1$	enumeration [distance, count]
1	1	2	[1,2]
2	$\sqrt{2}$	8	[1,4], [$\sqrt{2}$,4]
3	$\sqrt{3}$	26	[1,6], [$\sqrt{2}$,12],[$\sqrt{3}$,8]

Table 1: Distance for Neighborhood node, for k-dimensional space ($k = 1, 2, 3$)

3 K-Grid Data Input

We assume that we have a group F of files, strings, or in general objects given as input to the K-Grid and that $|F| \leq V$. We define an invertible function A called an *assignment* of the group F to the K-Grid so that each object in $f \in F$ has a specific location $A(f)$. We also have $A^{-1}(\boldsymbol{p})$ that takes as input any location and returns the object stored there or the special constant $\emptyset$ meaning empty. We assume a domain-specific (not Euclidean!) distance function given as a matrix $d(x, y)$ whose arguments index objects in F that returns a nonnegative real number in all cases. Although any distance function may be used, two are given as examples in this manuscript. The simplest is a distance function based on an example height attribute given later. The other distance function used in practice in a preferred embodiment uses Normalized Compression Distance (NCD) via data compression. NCD, an information distance, has a number of different modes and can utilize any normal data compression program. However, in one of our preferred embodiments we use *zlib* as the data compressor which is similar to *gzip*. Another specific case of NCD that is of interest to us is the Normalized Web Distance, or NWD, for natural language processing. Because there is only a finite number of objects in F, there are only finitely many sample points possible for the distance function and these may be understood as the upper- or lower-triangular half of a distance matrix. We assume also that each object in F has a *label* that can be used to represent the object when rendering the K-Grid. This label might be a word, number, or image that in some way represents the contents of the file or string at a high level.

4 K-Grid Global Cost Objective Function

A K-Grid can be used to induce a global cost or benefit function over all assignments of the group F via an aggregate weighted sum of costs over all local neighborhoods. Assume that we have an assignment A for the set of objects F in the K-Grid. Then we define the local cost for $\gamma > 0$

$$c(\boldsymbol{p}) = \left(\sum_{\boldsymbol{q} \in L(\boldsymbol{p})} \frac{d(A^{-1}(\boldsymbol{q}), A^{-1}(\boldsymbol{p}))^{\gamma}}{\|\boldsymbol{q} - \boldsymbol{p}\|} \right)^{\frac{1}{\gamma}} \tag{1}$$

Next we sum this location-specific cost function over all locations in the K-Grid:

$$C = \sum_{\boldsymbol{r} \in U} c(\boldsymbol{r}) w(\boldsymbol{r}) \tag{2}$$

Note that in order to complete this sum we need to extend the distance function to include an arbitrary distance from each object to an empty space as well as an arbitrary distance between empty spaces. These structural constants determine the phase behavior of the system as explained later.

We write w to represent an arbitrary location-dependent weighting function. It is possible to use any positive constant for this number. However, doing so causes inconvenient symmetries in certain important use-cases. Therefore it is also considered useful to use a non-constant non-isotropic weighting function to normalize orientation as described later.

5 K-Grid Mutation Operations

We define a *block* in one dimension as a range of locations described using a lower and upper bound. In two dimensions a *block* is a rectangle, and in higher dimensions we continue to select a set of locations that form a complete non-empty rectilinear grid within a potentially larger K-Grid. We define a non-overlapping pair of blocks within a K-Grid as two blocks with equal dimensions and orientation that do not have any locations in common. We define a block-swap operation on an assignment A as a transposition of all points from one block to the other for a non-overlapping pair of blocks. We optionally allow a single bit of freedom per dimension to indicate if items are copied in forward or reversed (reflected) direction.

We use a pseudo-random number generator to select a non-overlapping pair of blocks to create a *simple mutation* operation that transforms assignment A into a different assignment A' as in Figure 5

We define a *complex mutation* as follows:

```
begin loop:
  apply simple mutation
  toss fair coin
  if heads,
    repeat loop.
  if tails,
    end loop.
```

6 K-Grid Search Heuristic

We search the space of K-Grid assignments for a specific group of objects and a specific distance function by using a greedy algorithm as follows:

```
let FAILCOUNT = 0
initialize K-Grid CURRENT to random or arbitrary starting assignment.
calculate objective function cost and store in BESTCOST
begin loop:
  let CAND = copy of CURRENT
  apply complex mutation to CAND
  calculate objective function cost of CAND and store in CANDCOST
  if CANDCOST < BESTCOST then:
    let BESTCOST = CANDCOST
    let CURRENT = CAND
    let FAILCOUNT = 0
  else
    let FAILCOUNT = FAILCOUNT + 1

  if FAILCOUNT < MAX_ITERATIONS,
    repeat loop.
  else
    end loop

output CURRENT and BESTCOST
```

See Figures 6,7,8

6.1 Phase Structure

The values used to fill in the information distance function d when objects interact with an empty neighbor or when two neighbors interact determine the *phase behavior* of the system. Different values for these parameters can cause objects to clump together like a solid or to fly apart like a gas.

6.2 Orientation Normalization

Since the local neighborhood is defined symmetrically with respect to reflection and orientation and because the global cost is a function of the local neighborhoods summed, the symmetry of the w weighting function is significant. If it is symmetric, then all possible reflections and potentially orientations can have minimum cost. This creates instability and difficulty in reading results that are only slightly changed between successive runs. To solve this problem, we suggest an uneven weighting function such as:

$$w(\boldsymbol{p}) = \sum_{i=1}^{k} \frac{(100 + p_i)(i + 100)(i + 100)}{1000000} \tag{3}$$

This causes the objects to fall to one corner and empty spaces to float to the opposite corner in the solid phase parameter regime. This stabilizes orientation to allow for iterative refinement or successive comparison.

7 Hierarchical Clustering

It is possible to use the K-Grid for hierarchical clustering. We may consider an entire K-Grid to be enclosed within a single cell of a larger K-Grid. It is necessary to establish a method for converting the assignment calculated within a smaller K-Grid into an object suitable for analysis at a greater level. In the case of files or strings using NCD as a distance function concatenation is a natural choice. By iterating through all the points in a K-Grid in a natural order we may convert a low-cost assignment into a low-entropy sequence ordering for the entire group of objects. This then allows K-Grid to be used to reorder smaller objects into more meaningful sequences that can then be analyzed at greater levels in enclosing K-Grids. In Figure 12 we show an example of a one dimensional hierarchical K-Grid.

8 K-Grid Examples

In Figure 10, we show a 2-d K-Grid based on mitochondrial gene sequences from certain mammals. In Figure 11, we show the resulting assignment after several iterations of the cost reducing optimization algorithm. By using Normalized Compression Distance for our distance function in this case we avoid specifying any biological information outside of the uninterpretted gene sequences.

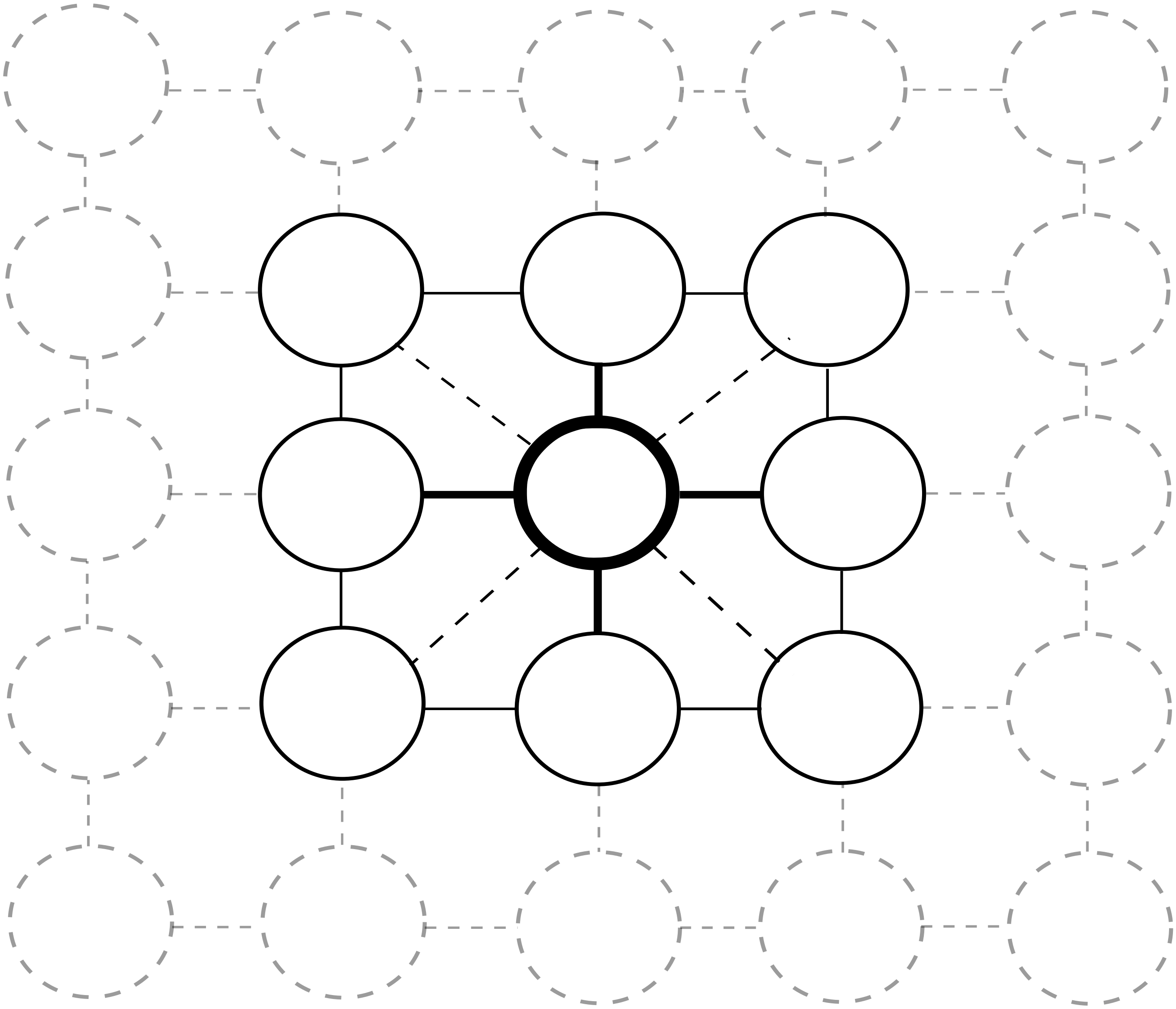

Figure 3: Two dimensional local neighborhood showing 8 neighbors, 1 or ($\sqrt{2}$)unit distant from a central node.

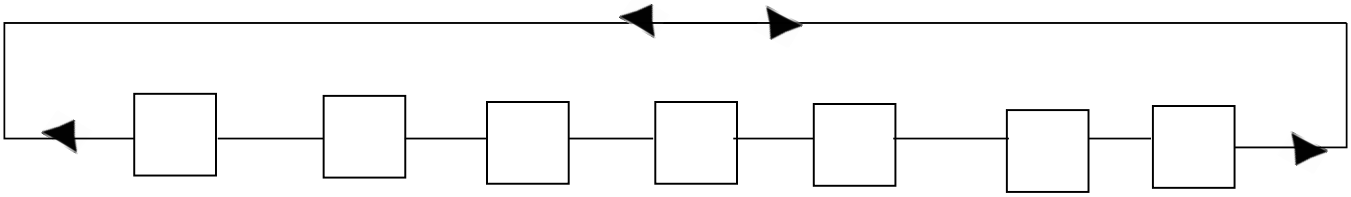

Figure 4: We illustrate a *wrapped K-Grid* with a simple one-dimensional example. The first and the last items are connected with one unit distance.

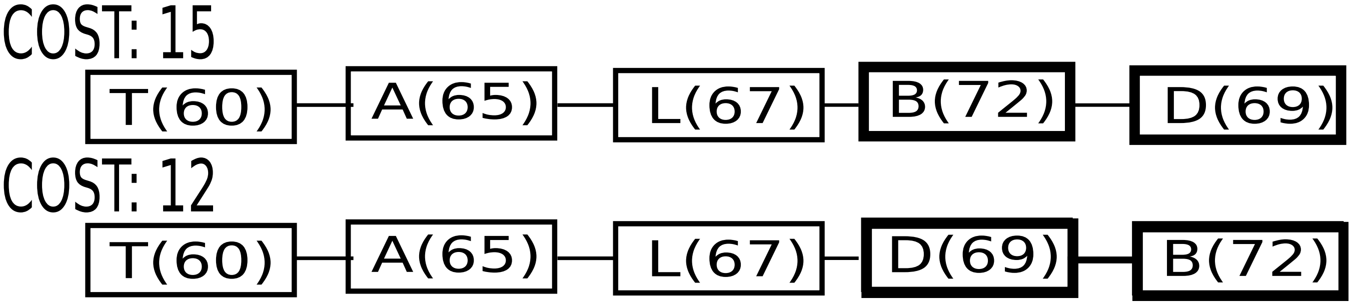

Figure 5: Simple unit (minimum) length swap in 1 dimensional *K-Grid*.

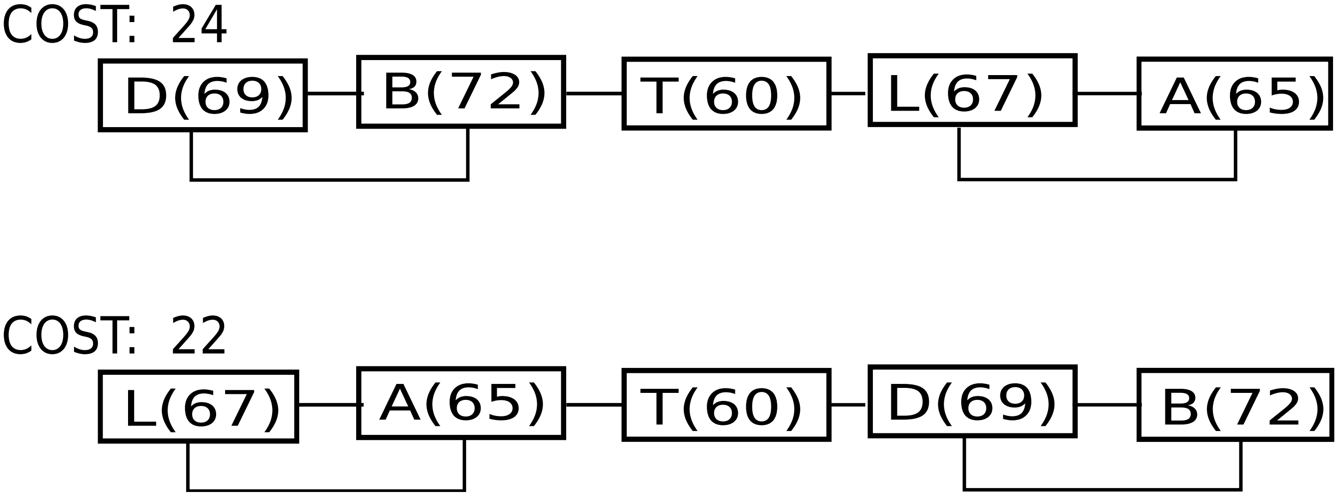

Figure 6: One dimensional *K-Grid* with a specific assignment in the upper row. The next row shows the assignment after a *two-length-segment* swap mutation.

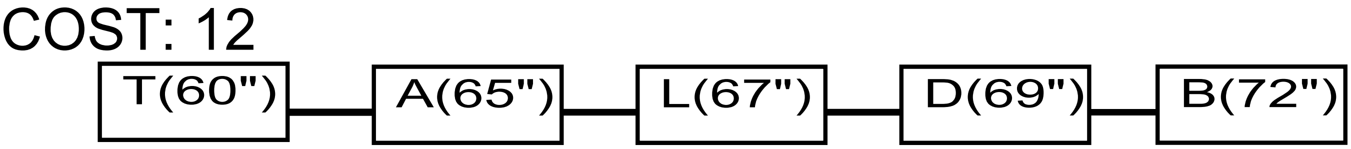

Figure 7: After many iterations, the optimal (lowest cost) arrangement is a sorted list.

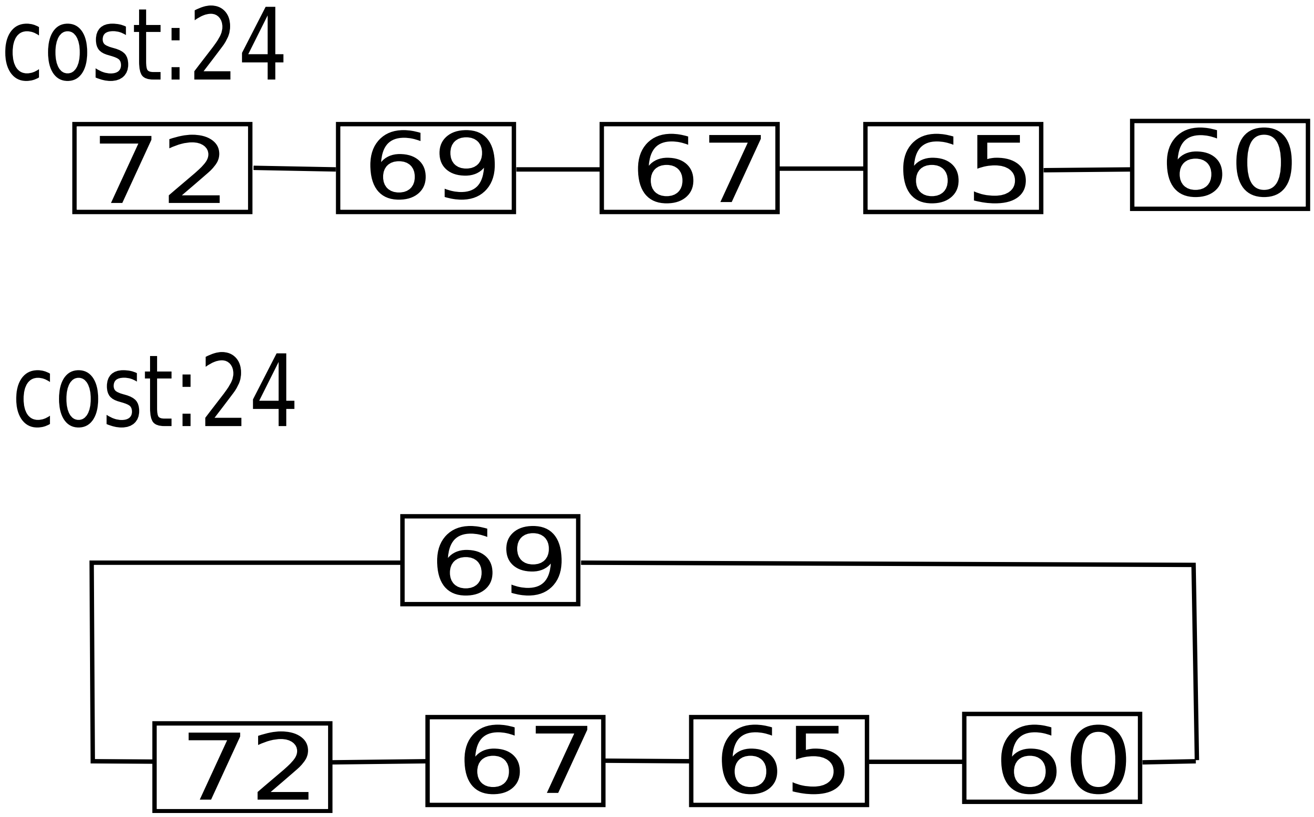

Figure 8: For wrapped *K-Grid*, the first and the last items are connected, so the total cost is 24 for both top and bottom arrangement in case of $\gamma = 1$ in 4. If $\gamma = 2$, the cost will be 17(top) and 15.3(bottom), the optimal one is the second.

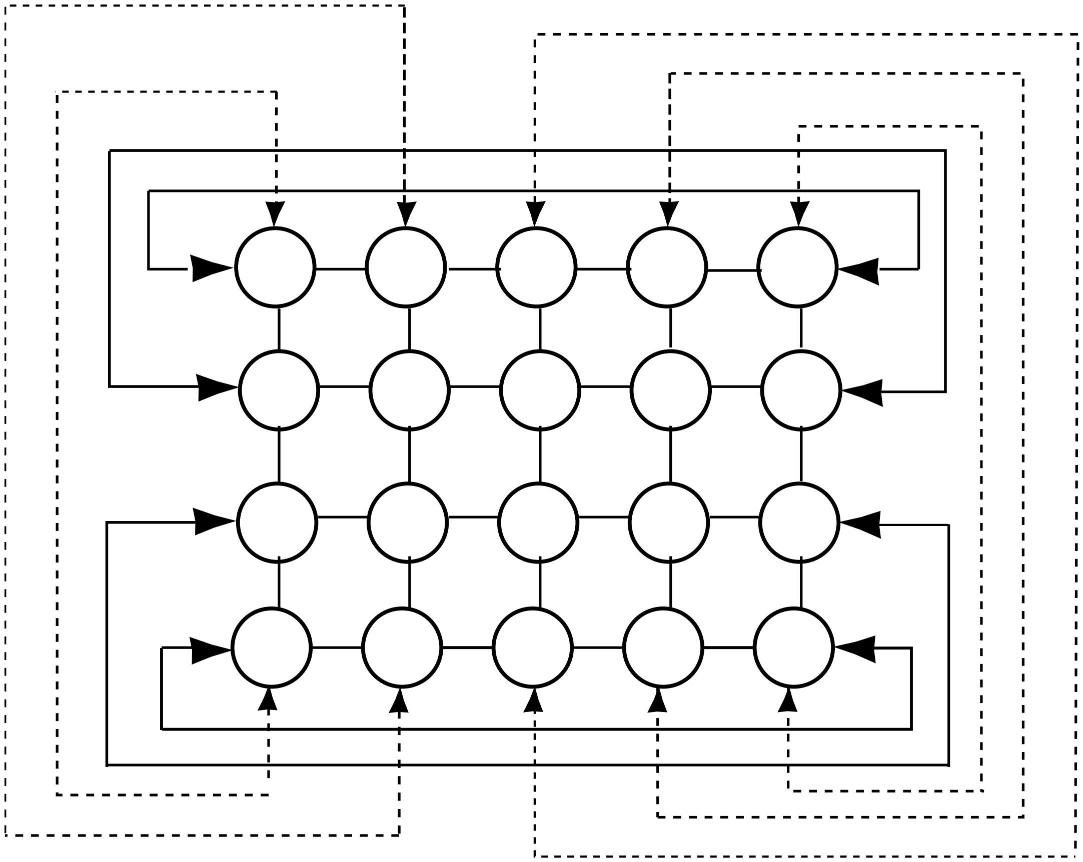

Figure 9: 4×5 2-dimensional *K-Grid* wrapping. (torroidal grid or torus shape)

COST: 84.4041

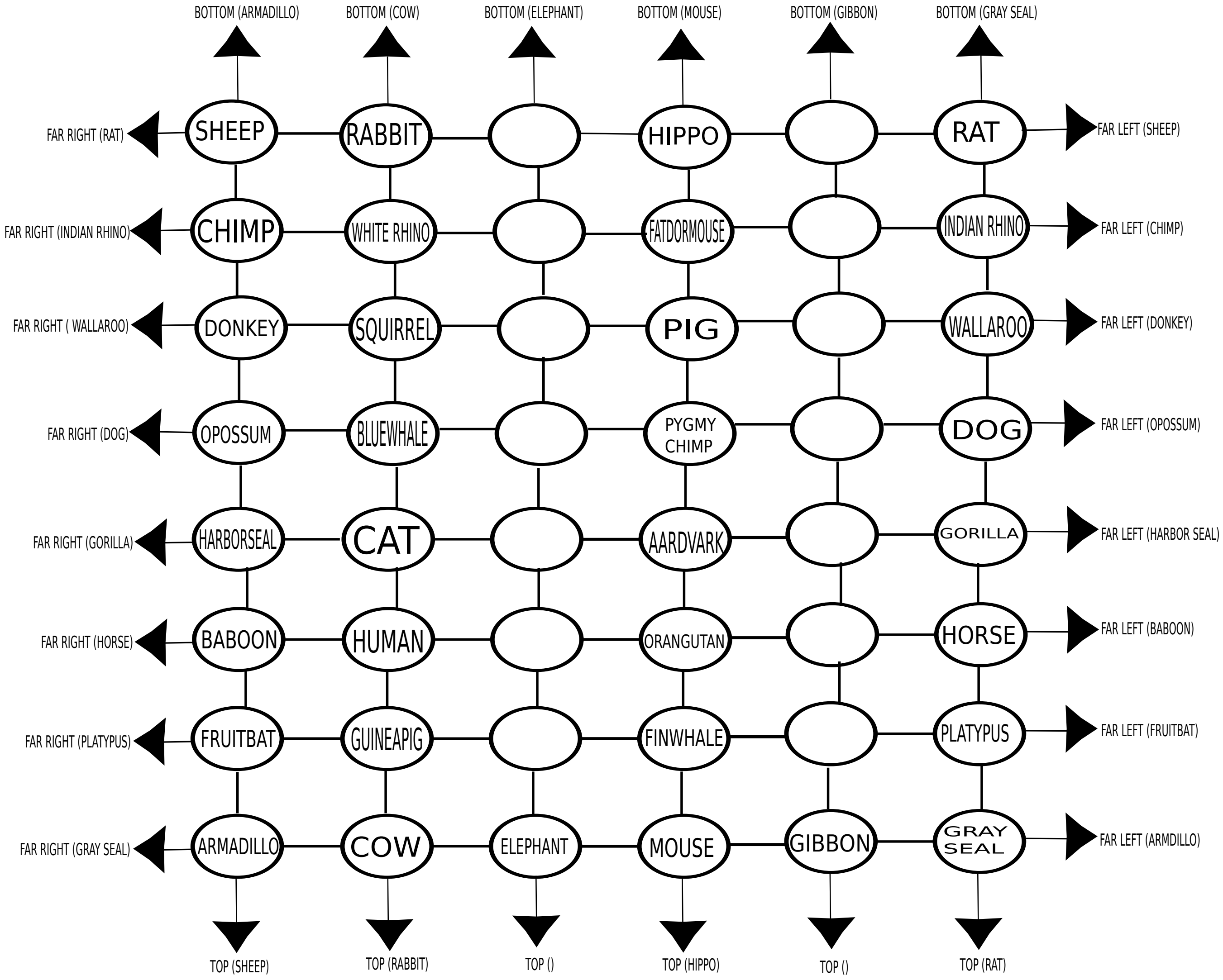

Figure 10: One example of 2D *K-Grid* with high cost, showing animals placed randomly.

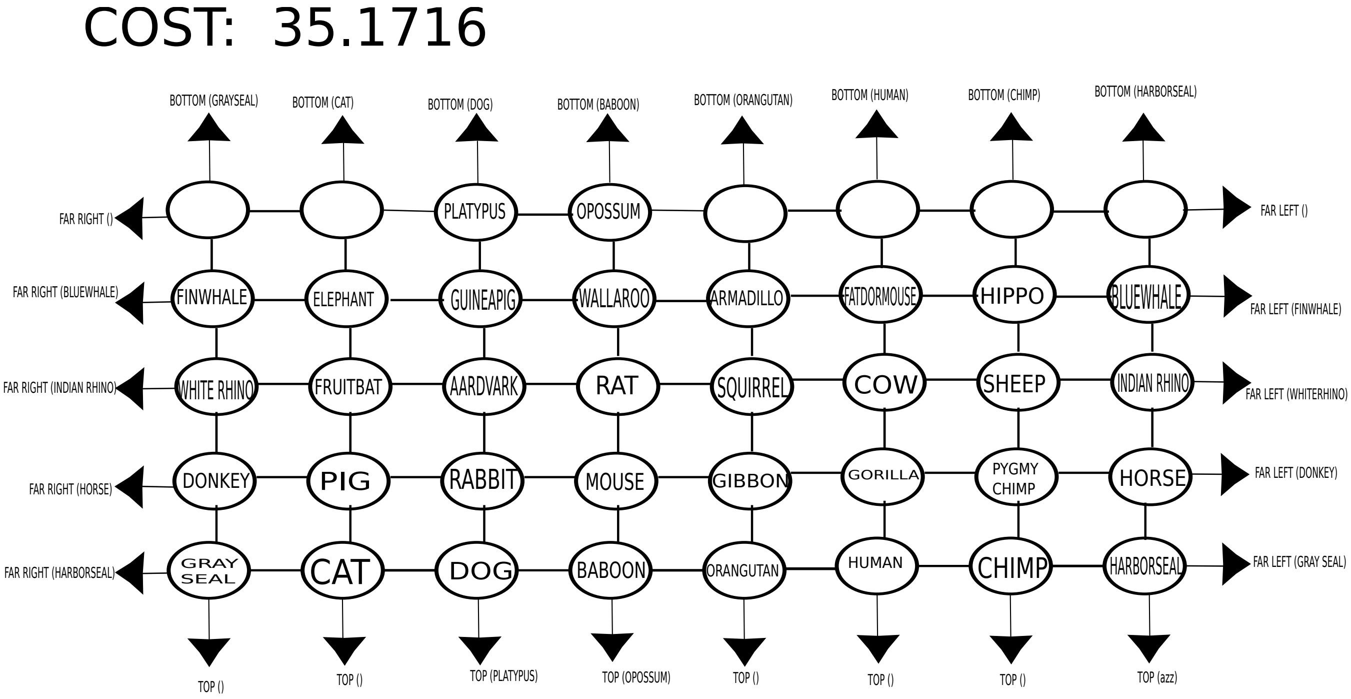

Figure 11: One example of 2d *K-Grid* (sorted).

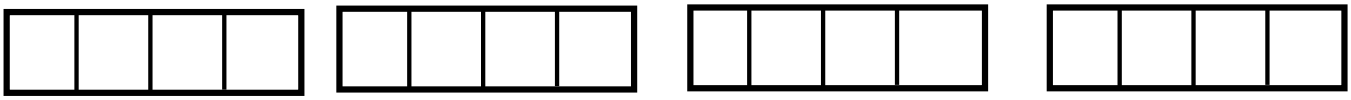

Figure 12: 4-unit 2-level 1-d hierarchical *K-Grid*