Can the external directed edges of a complete graph form a radially symmetric field at long distance?

S. Halayka*

March 31, 2010

Abstract

Using a simple numerical method, the external directed edges of a complete graph are tested for their level of fitness in terms of how well they form a radially symmetric field at long distance (e.g., a test for the inverse square law in 3D space). It is found that the external directed edges of a complete graph can very nearly form a radially symmetric field at long distance if the number of graph vertices is great enough.

1 Method

Complete graphs have been used to construct a model of quantum gravity [1].

It is considered here that a complete graph G_1 consists of:

1. $n(G_1)$ vertices $V(G_1)$ that are uniformly distributed along a shell $S(G_1)$ of radius $r(G_1)$.
2. $(n(G_1)^2 - n(G_1))/2$ internal non-directed edges $I(G_1)$ (e.g., line segments) that join the pairs of vertices together.
3. $n(G_1)^2 - n(G_1)$ external directed edges $E(G_1)$ (e.g., rays) that are extensions of $I(G_1)$.

See Figure 1 for a diagram of a complete graph where $n(G_1) = 3$.

It seems fundamentally important to question whether or not the external directed edges $E(G_1)$ can form a radially symmetric field at long distance. If the field is to be considered radially symmetric, then the following two fitness criteria must be met:

1. With regard to a second shell $S(G_2)$ of larger radius $r(G_2) > r(G_1)$, the $n(G_2) = n(G_1)^2 - n(G_1)$ vertices $V(G_2)$ corresponding to where the external directed edges $E(G_1)$ intersect with $S(G_2)$ should be uniformly distributed along $S(G_2)$.

*shalayka@gmail.com

2. The external directed edges $E(G_1)$ should be normal to $S(G_2)$ at their respective intersection vertices.

With regard to the first criterion (e.g., uniform distribution fitness), the vertices $V(G_2)$ should be compared to an equal number $n(G_3) = n(G_2)$ of vertices $V(G_3)$ that are known to be uniformly distributed along a third and final shell $S(G_3)$ of radius $r(G_3) = r(G_2)$.

The generation of $n(G_3)$ uniformly distributed vertices along a 1D shell (e.g., a circle) is algorithmically simple: divide the circle's 2π radians into $n(G_3)$ equal portions and then use the polar coordinate equations to generate the $n(G_3)$ corresponding vertex positions. The generation of $n(G_3)$ uniformly distributed vertices along a 2D shell (e.g., a thin spherical shell) is not algorithmically simple: an iterative vertex repulsion code [3] was used here to generate $n(G_3)$ roughly uniformly distributed vertices.

The uniform distribution fitness test used here compares G_2, G_3 by analyzing the lengths of the internal non-directed edges $I(G_2)_{ij}$, $I(G_3)_{ij}$ (e.g., where $i = \{1, 2, \ldots, n(G_2)\}$, $j = \{1, 2, \ldots, n(G_2) - 1\}$, and so each unique internal directed edge is used for exactly two comparisons). Some kind of order must be established so that a reasonable correlation between $I(G_2)_{ij}$, $I(G_3)_{ij}$ can be produced. As such, the lengths of the internal non-directed edges corresponding to each and every vertex $V(G_2)$, $V(G_3)$ are placed into sorted bins $L(I(G_2))$, $L(I(G_3))$ before the comparison begins

$$L(I(G_2))_i = \text{sort}[\text{length}(I(G_2)_{i1}), \ldots, \text{length}(I(G_2)_{i(n(G_2)-1)})], \tag{1}$$

$$L(I(G_3))_i = \text{sort}[\text{length}(I(G_3)_{i1}), \ldots, \text{length}(I(G_3)_{i(n(G_3)-1)})]. \tag{2}$$

Since $V(G_3)$ are known to be uniformly distributed along $S(G_3)$, the sorted bins $L(I(G_3))$ should all contain identical length distributions (e.g., the reference distribution $L(G_3)_{\text{ref}}$). If $V(G_2)$ are also uniformly distributed along $S(G_2)$, then the sorted bins $L(I(G_2))$ should also all contain length distributions identical to $L(G_3)_{\text{ref}}$.

The uniform distribution fitness test is

$$F_D(G_1) = [0, 1], \tag{3}$$

$$F_D(G_1) = \frac{\displaystyle\sum_{i=1}^{n(G_2)} \sum_{j=1}^{(n(G_2)-1)} \frac{\min[L(I(G_2))_{ij}, L(I(G_3))_{ij}]}{\max[L(I(G_2))_{ij}, L(I(G_3))_{ij}]}}{n(G_2)^2 - n(G_2)}. \tag{4}$$

Again, each unique internal directed edge is used for exactly two comparisons, which is why equation (4) is normalized using $n(G_2)^2 - n(G_2)$, not $(n(G_2)^2 - n(G_2))/2$.

With regard to the second criterion (e.g., normal fitness), each external directed edge $E(G_1)_{ij}$ corresponds to one intersection vertex $v(G_2)_k$ (e.g., $i = \{1, 2, \ldots, n(G_1)\}$, $j = \{1, 2, \ldots, n(G_1) - 1\}$, $k = \{1, 2, \ldots, n(G_2)\}$. Where both

$S(G_1)$ and $S(G_2)$ are centred at the coordinate system origin, the normal fitness test is

$$F_N(G_1) = [0,1], \tag{5}$$

$$F_N(G_1) = \frac{\displaystyle\sum_{i=1}^{n(G_1)} \sum_{j=1}^{(n(G_1)-1)} \frac{\widehat{E}(G_1)_{ij} \cdot \widehat{V}(G_2)_k + 1}{2}}{n(G_2)}. \tag{6}$$

2 Results

The 1D and 2D shell fitness test results for various $n(G_1)$, $r(G_1)$, and $r(G_2)$ are listed in the following tables

Uniform distribution fitness $F_D(G_1)$ for a 1D shell of radius $r(G_1) = n(G_1)$

$r(G_2)$ \ $n(G_1)$	2	4	8	16	32	64	128
10^3	1	0.829572	0.893257	0.946621	0.993124	0.997482	0.998891
10^{10}	1	0.827916	0.886982	0.93004	0.95885	0.976516	0.986852
10^{17}	1	0.827916	0.886982	0.93004	0.95885	0.976516	0.986852

Normal fitness $F_N(G_1)$ for a 1D shell of radius $r(G_1) = n(G_1)$

$r(G_2)$ \ $n(G_1)$	2	4	8	16	32	64	128
10^3	1	0.999999	0.999993	0.99997	0.999876	0.999496	0.997962
10^{10}	1	1	1	1	1	1	1
10^{17}	1	1	1	1	1	1	1

Uniform distribution fitness $F_D(G_1)$ for a 2D shell of radius $r(G_1) = n(G_1)$

$r(G_2)$ \ $n(G_1)$	2	4	8	16	32	64	128
10^3	1	0.937087	0.931829	0.974859	0.97905	0.995469	0.998372
10^{10}	1	0.937088	0.930686	0.973607	0.974738	0.994824	0.997366
10^{17}	1	0.937088	0.930686	0.973607	0.974738	0.994824	0.997366

Normal fitness $F_N(G_1)$ for a 2D shell of radius $r(G_1) = n(G_1)$

$r(G_2)$ \ $n(G_1)$	2	4	8	16	32	64	128
10^3	1	0.999999	0.999993	0.99997	0.999876	0.999496	0.997963
10^{10}	1	1	1	1	1	1	1
10^{17}	1	1	1	1	1	1	1

As these fitness test results show, the external directed edges of a complete graph can very nearly form a radially symmetric field at long distance if the number of vertices is great enough. For instance, a 2D shell in 3D space can very nearly reproduce the inverse square law (e.g., field strength proportional to $1/r$).

See [2] for the full code and expanded table data. In the full code, the iterative vertex repulsion code [3] has been modified to use the Mersenne Twister pseudorandom number generator (PRNG) code [4]. The full code also uses a modified version of the ray-shell intersection code given in [5].

References

[1] Konopka T, Markopoulou F, Smolin L. Quantum Graphity. (2006) arxiv:hep-th/0611197

[2] Halayka S. Fitness C++ code. http://completegraph-rays.googlecode.com/files/rays_1.0.zip

[3] Bulatov V. Vertex repulsion code. http://www.math.niu.edu/~rusin/known-math/96/repulsion

[4] Saito M, Matsumoto M, Hiroshima University. SIMD-oriented Fast Mersenne Twister (SFMT). http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT/

[5] Dunn F, Parberry I. 3D Math Primer for Graphics and Game Development, section 13.12. ISBN: 978-1556229114

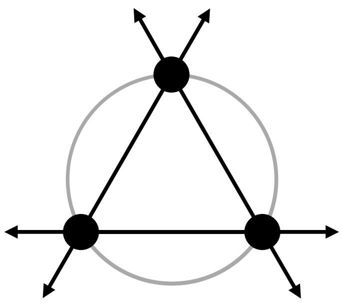

Figure 1: A complete graph G_1, where $n(G_1) = 3$ vertices (e.g., black disks) are uniformly distributed along a 1D shell $S(G_1)$ (e.g., a gray circle). There are $(n(G_1)^2 - n(G_1))/2 = 3$ internal non-directed edges $I(G_1)$ (e.g., black line segments), and $(n(G_1)^2 - n(G_1)) = 6$ external directed edges $E(G_1)$ (e.g., outward pointing black rays).