New Horizons in Sphere-Packing Theory, Part III: Regular Noncartesian Directed Graphs for FPGA Interconnects

Joseph Cessna and Thomas Bewley

***Abstract*—In directed graph topologies such as those used in field-programmable gate arrays (FPGAs), there are two competing design objectives: (i) spread information rapidly across the graph, and (ii) incorporate path redundancy and significant local (nearest neighbor) communication. At one end of this spectrum, for the most rapid information spread possible, butterfly graphs are optimal, though they completely lack path redundancy. At the other end of this spectrum, Cartesian-based graphs incorporate extensive local structure and path redundancy, though they essentially sacrifice the ability to spread information rapidly across the graph. Previous efforts to accelerate the spread of information in Cartesian-based graphs have been somewhat ad hoc from a graph-theoretic perspective. The present work generalizes these two classes of directed graphs into a unified, well-structured framework, with the Cartesian and butterfly graphs as special cases of a more general class of interconnects; this clearly-defined and well-structured new framework facilitates the optimal (application-specific) balance to be struck somewhere in-between the butterfly and Cartesian extremes.**

***Index Terms*—Structured computational interconnect, FPGA, directed graph, butterfly graph, 2D & 3D cartesian graph**

I. Background & Terminology

REGULAR directed graphs are built from a set of identical nodes containing an equal number of incoming and outgoing directed links. Such graphs are useful for the propogation and processing of information within high-performance massively parallel computer chips such as field-programmable gate arrays (FPGAs); comprehensive reviews of this active field of research include Dally & Towles (2003) and Duato, Yalmanchili, & Ni (1997), and the many references contained therein. In such chips, a large amount of information is fed in parallel into and across the graph, one stage at a time, in order to solve rapidly a large a complex problem via many relatively simple pieces processed in a massively parallel setting. A better understanding of the properties and tradeoffs inherent to the directed graphs used at the heart of these designs might facilitate the more effective design of such chips.

The present paper is a part of a larger study on the utility of noncartesian graphs and packings; Part I of this study (Bewley, Belitz, & Cessna 2011) describes the fundamental concepts and constructions, and Part II (Belitz & Bewley 2011) highlights the application of a closely-related body of theory to the problem of efficient derivative-free optimization.

The directional graphs we will consider in this work, which are discussed further in Bewley, Belitz, & Cessna (2011), are derived from more general $(n+1)$-dimensional topologies that contain a flow direction $\mathbf{x}_0$, along which we can locate discrete sets of nodes lying in the sequence of orthogonal n-dimensional hyperplanes. We think of information flowing forward, along this direction, passing in turn through each orthogonal hyperplane (or *stage*) of the graph. For finite graphs, each stage contains the same number of nodes. We then define the *total cardinality* of the graph as $(N_0 \times N_1 \times \cdots \times N_n)$, where N_0 is the number of stages in the graph, $M = (N_1 \times \cdots \times N_n)$ is the number of nodes in each stage, and N_k is the k^{th} *cardinality* (i.e. the number of nodes in the k^{th} dimension of each stage). Thus, the *size* (total number of nodes) of the graph is $N_0 \times M$.

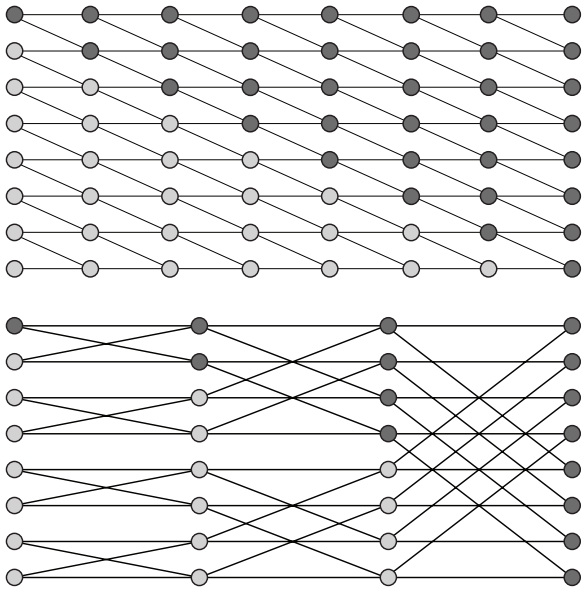

Fig. 1. Two traditional directed graphs are shown; information is assumed to flow from left to right. The 2D cartesian graph (top, 8×8) and the *2-ary 4-fly* (bottom, $4 \times 2 \times 2 \times 2$) represent the opposite extremes with regards to performance characteristics. The cartesian graph contains exclusively local links, and consequently has a lot of path diversity. In contrast, the butterfly graph spreads information across the nodes much more efficiently, but lacks any sense of locality from stage to stage and contains no path diversity.

Necessarily, a node in a given stage must only connect to nodes in the immediately preceding and following stages; it may not connect to any nodes in its same stage. The connection between stages is governed by the overall topology of the graph and will vary from stage to stage in a repeating pattern. The number of distinct connections between stages is a function of the dimension n of the stages themselves. For the directed graphs of interest, a node will connect to exactly s nodes in both the forward and backward direction. Thus, s

The authors are with the Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093 USA (e-mail: jcessna@ucsd.edu; bewley@ucsd.edu).

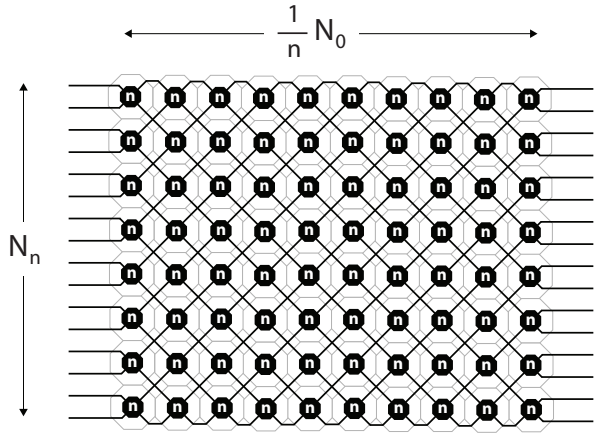

Fig. 2. The generalized degree-2 directed graph. Each stage is n-dimensional, and the total cardinality is $(N_0 \times N_1 \times \cdots \times N_n)$. When $n = 1$ this graph reduces to the 2D cartesian graph and when $N_k = 2, \quad \forall k > 0$, this graph reduces to the well-known *2-ary* butterfly. The full connectivity is defined recursively in Figure 3.

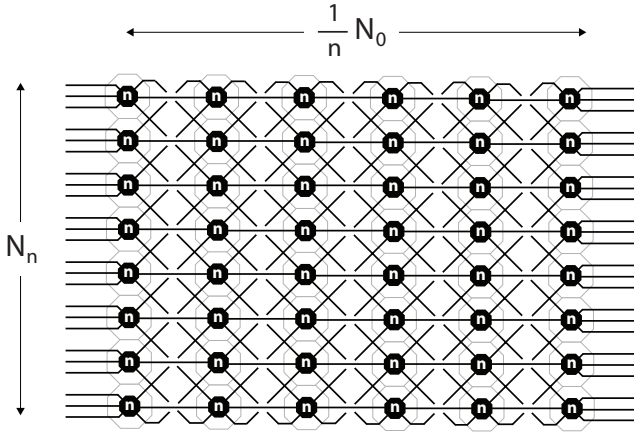

Fig. 4. The generalized degree-3 directed graph. Each stage is n-dimensional, and the total cardinality is $(N_0 \times N_1 \times \cdots \times N_n)$. When $N_k = 3, \quad \forall k > 0$, this graph reduces to the well-known *3-ary* butterfly. The full connectivity is defined recursively in Figure 5.

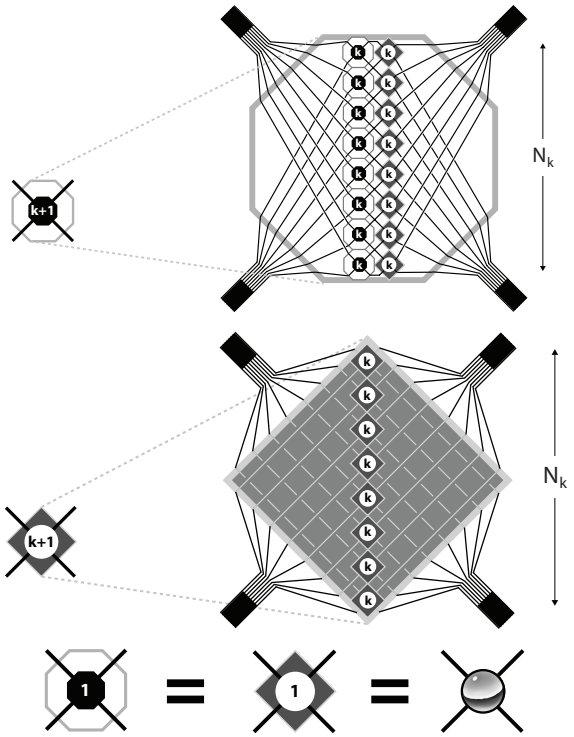

Fig. 3. The generalized degree-2 directed graph is best graphically defined recursively through the combination of the two basic building blocks shown above. At the lowest level, these building blocks reduce to a single node.

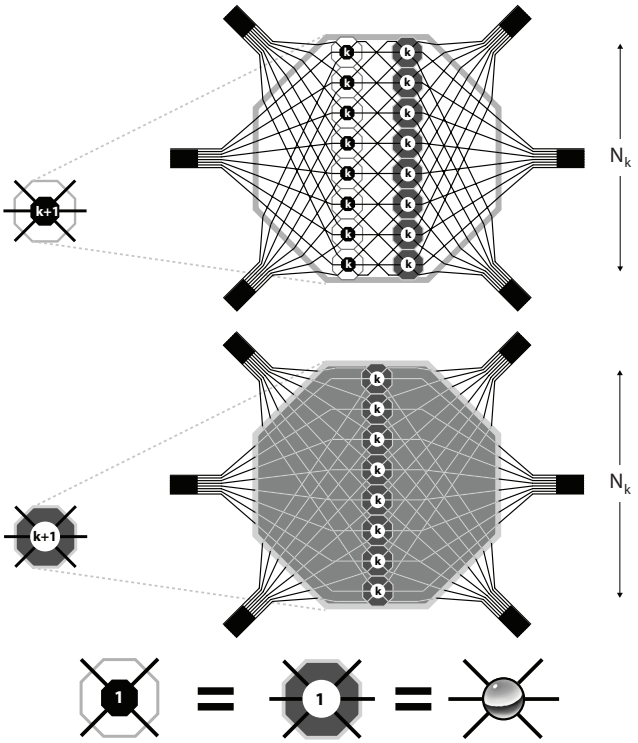

Fig. 5. The generalized degree-3 directed graph is best graphically defined recursively through the combination of the two basic building blocks shown above. At the lowest level, these building blocks reduce to a single node.

defines the degree of the graph. For this paper, we will focus exclusively on directed graphs of degree $s = 2$ and $s = 3$.

A node in a given stage connects to its s forward neighbors along a *channel*. Two nodes are said to be connected if there exists an ordered set of channels *in the flow direction only* (i.e. a *path*) that connects a node in one stage to another node in a subsequent stage. It is important to note that paths only move in the flow direction. Redundant paths occur when two nodes can be connected by more than one unique path. These redundant paths are important because they decrease bottlenecks in the graph and thus increase the graph's overall robustness. A graph with many redundant paths is usually efficient at performing local data operations such as simple comparisons or additions.

On the other hand, an increase in path redundancy necessarily is associated with a decrease in the *spread* of a graph. The spread is a measure of how quickly information from one node reaches the other nodes in subsequent stages. *Saturation* occurs when the information from one node spreads to every node in a future stage. The *saturation length* is the number of traversed stages required until saturation is achieved. A graph with a low saturation length has a high communication rate across all nodes and thus can efficiently perform global data operations such as data transposes.

II. General Classes

A 2D cartesian graph (see top of Figure 1) has nodal degree $s = 2$ and contains $N_0 = q$ stages with dimension $n = 1$. The total cardinality of the graph is $(q \times p)$, there are $M = p$ nodes per stage, and the size of the graph is qp. The cartesian graph has a high saturation length of $q-1$ when compared to a comparable butterfly graph due to the high number of redundant paths. These paths are a byproduct of the local nature of the cartesian interconnect.

In contrast, a traditional *p-ary q-fly* butterfly graph (see Dally & Towles 2003, and the bottom subfigure of Figure 1) has nodal degree $s = p$ and contains $N_0 = q$ stages with dimension $n = q-1$. The total cardinality of the graph is $(q \times p \times \cdots \times p)$, there are $M = p^{q-1}$ nodes per stage, each flow normal cardinality is equal to the nodal degree $N_k = p$, and the size of the graph is qM. As a result, it can be shown that the butterfly graph provides the minimum saturation length of any graph with stage size p^{q-1}. However, each node has only one unique path to the nodes of the saturated stage and thus lacks any path diversity (and consequently local nature).

To achieve the minimal saturation length, the traditional degree-2 butterfly graph places many unnecessary restrictions on the parameters of the graph (e.g. the dimension of each stage, the cardinality of each dimension). While keeping the nodal degree, overall size, and number of stages of the graph constant, we can relax the constraints on the dimension and allow freedom to pick each dimension's cardinality individually, subject to

$$\prod_{k=1}^{n} N_k = M. \tag{1}$$

This allows us to define an entire family of degree-2 graphs that include on one extreme the butterfly graph and on the other the simple 2D cartesian graph. This family is illustrated in Figures 2 and 3. As one might expect, the choices of both dimension and cardinality have a direct effect on the local versus global nature of the resulting graph, and therefore allow for much design flexibility.

In a similar manner, the degree-3 butterfly graph can be generalized by also removing the dimensionality and cardinality constraints, subject to the overall size constraint of (1). This produces another family of directed graphs that span the range from entirely local to fully global connectivity. However, unlike the degree-2 case, this family does not reduce to 3D cartesian, as one might expect. This family of directed graphs is shown in Figures 4 and 5.

In both the generalized degree-2 and degree-3 classes, the connectivity between stages can be thought of as spanning only one dimension at a time, in turn cycling through each dimension. Thus, the connectivity pattern repeats itself every n stages. An interesting third class of directed graphs can be found by allowing the channels between adjacent stages to span two dimensions at a time. This requires a degree-3 node and an even number of dimensions in the graph's stages. With this design, the interconnect pattern repeats itself every $n/2$ stages. As a result, we are able to obtain more local path redundancy with higher-dimensional stages that in turn lead to shorter saturation lengths. This family of graphs is illustrated

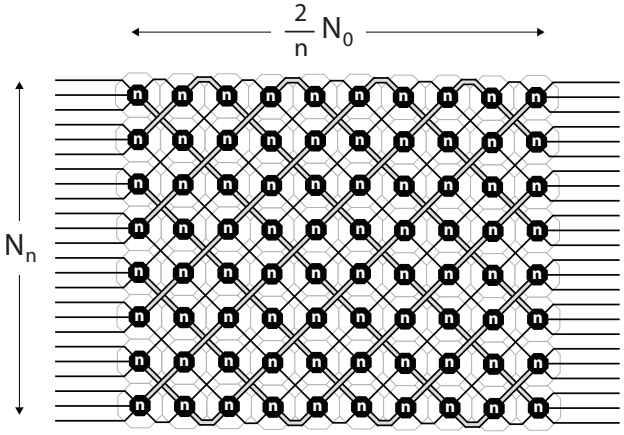

Fig. 6. The generalized degree-3 double-twist directed graph. Each stage is n-dimensional (for n even), and the total cardinality is $(N_0 \times N_1 \times \cdots \times N_n)$. When $n = 2$ this graph reduces to the well-known 3D cartesian graph. The full connectivity is defined recursively in Figure 7

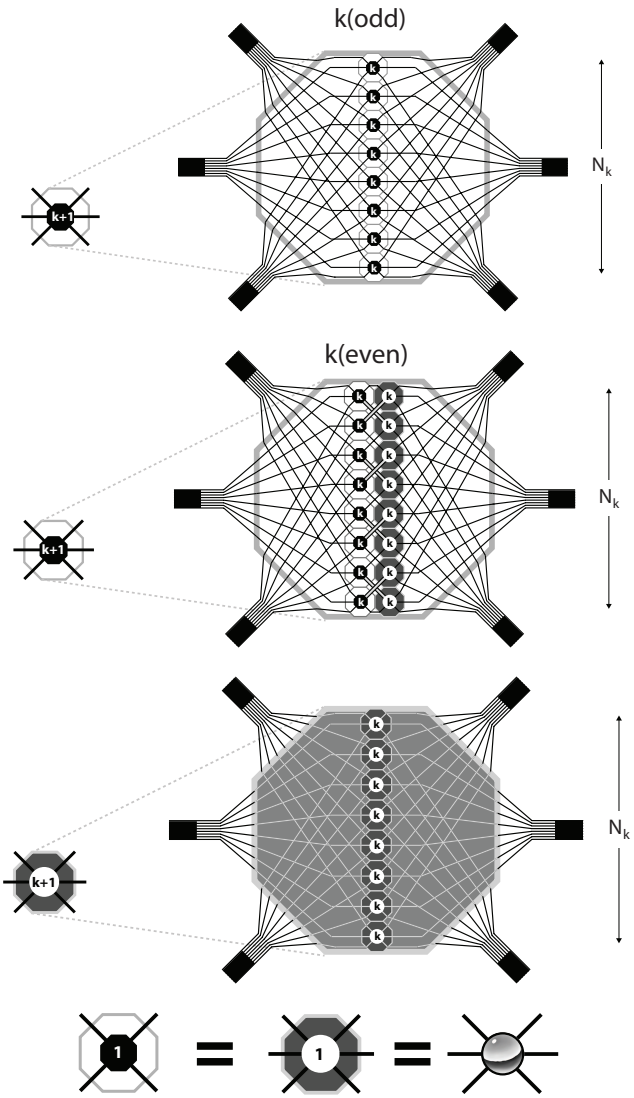

Fig. 7. The generalized degree-3 double-twist directed graph is best graphically defined recursively through the combination of the two basic building blocks shown above. For one of the building blocks, the definition varies at the odd and even levels. At the lowest level, though, these building blocks still reduce to the basic node.

TABLE I
INFORMATION SPREAD FOR SELECT DIRECTED GRAPHS (ASSUMING A LARGE NUMBER OF GATES PER STAGE).

Directed Graph		n	Gates reached at each stage from a particular input to the graph										g_{10}	g_{20}
Generalized degree-2	$\mathbb{Z}_2$, V_2^{90}	1	2	3	4	5	6	7	8	9	10	11	66	231
	A_3^+, V_3^{90}	2	2	4	6	9	12	16	20	25	30	36	161	946
	V_4^{90}	3	2	4	8	12	18	27	36	48	64	80	300	2,800
	V_5^{90}	4	2	4	8	16	24	36	54	81	108	144	478	6,797
	V_6^{90}	5	2	4	8	16	32	48	72	108	162	243	696	14,325
	V_7^{90}	6	2	4	8	16	32	64	96	144	216	324	907	27,110
	V_8^{90}	7	2	4	8	16	32	64	128	192	288	432	1,167	46,836
	V_9^{90}	8	2	4	8	16	32	64	128	256	384	576	1,471	76,126
	V_{10}^{90}	9	2	4	8	16	32	64	128	256	512	768	1,791	119,772
	V_{11}^{90}	10	2	4	8	16	32	64	128	256	512	1,024	2,047	176,122
Generalized degree-3	W_2^{90}	1	3	5	7	9	11	13	15	17	19	21	121	441
	W_3^{90}	2	3	9	15	25	35	49	63	81	99	121	501	3,301
	W_4^{90}	3	3	9	27	45	75	125	175	245	343	441	1,489	17,185
	W_5^{90}	4	3	9	27	81	135	225	375	625	875	1,225	3,581	70,857
	W_6^{90}	5	3	9	27	81	243	405	675	1,125	1,875	3,125	7,569	245,545
	W_7^{90}	6	3	9	27	81	243	729	1,215	2,025	3,375	5,625	13,333	741,161
	W_8^{90}	7	3	9	27	81	243	729	2,187	3,645	6,075	10,125	23,125	1,978,545
	W_9^{90}	8	3	9	27	81	243	729	2,187	6,561	10,935	18,225	39,001	4,855,001
	W_{10}^{90}	9	3	9	27	81	243	729	2,187	6,561	19,683	32,805	62,329	?
	W_{11}^{90}	10	3	9	27	81	243	729	2,187	6,561	19,683	59,049	88,573	?
$s=3$	$\mathbb{Z}_3$	2	3	6	10	15	21	28	36	45	55	66	286	1,771
	A_5^+	4	3	9	18	36	60	100	150	225	315	441	1,358	22,165

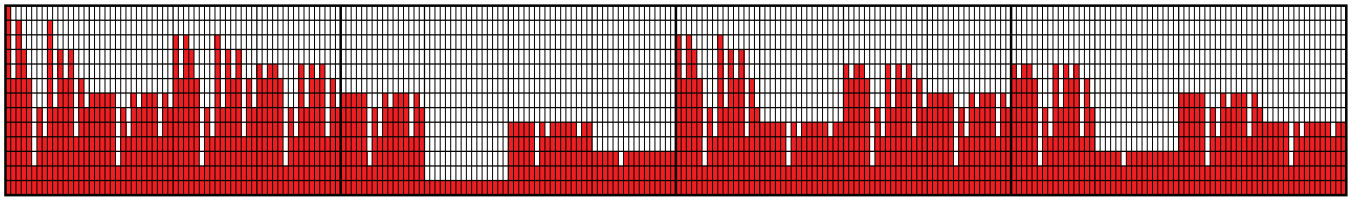

Fig. 9. The signal propagation through a directed graph with the A_5^+ topology. The illustrated graph has a total cardinality of $(12\times4\times4\times4\times4)$, thus each of the 12 stages of the graph contain $M=256$ nodes with periodic connections across each of the flow normal dimensions. Here, a signal is injected into the upper-left corner and travels down through the graph. The shaded blocks represent the gates that are reached by the signal at each subsequent stage. In this example, the original signal saturates all 256 gates at the 12^{th} stage. This illustrates how the periodic connections associated with finite cardinality serve to further reduce the predicted spread of Table I by increasing path redundancy.

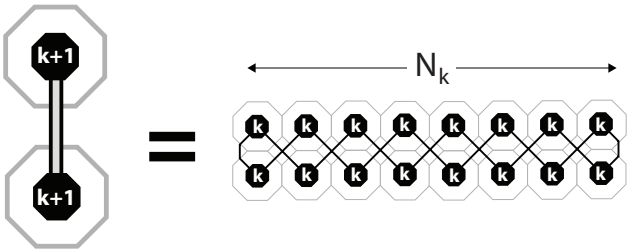

Fig. 8. The weave in the $(k+1)^{th}$ dimension is combined with an additional weave in the k^{th} dimension as defined above. This shortens the repeatative interconnect pattern between stages and allows for more path diversity while maintaining a sufficient global spread.

in Figures 6, 7 and 8. As it turns out, for $n=2$ this family of graphs reduces to the well known 3D cartesian topology.

III. DISCUSSION & SUMMARY

The properties of the three families of graphs are summarized in Table I. For a given graph, we look at the spread of information, starting from a single node and flowing forward through the graph. To separate the issue of periodic connections, an infinite cardinality was assumed during the calculations. In general, adding the necessary constraint of finite cardinality will increase overall path redundancy at the cost of reducing the expected spread shown in the table. The finite cardinality is addressed further in Figure 9. The shaded cells highlight the areas of optimal spread in each graph; in these regions, the information is spreading to the maximum number of nodes possible. As a result, there is no path redundancy. The unshaded cells illustrate where each graph diverges from the standard butterfly graph, introducing more locality and robustness to the graph at the expense of overall global spread. The final two columns of the table show the total number of nodes reached in stages 0–10 and 0–20, respectively. These numbers give a measure of the short versus long range spread of each particular graph; the difference between these two spreads is highlighted when comparing the W_4^{90} graph to the A_5^+ topology. Across the board, through the first ten stages, the W_4^{90} graph reaches more nodes, giving

it a larger short range spread. Thus, the A_5^+ graph contains more local structure and more path redundancy over this range. However, comparing g_{20} for each graph, one can see that past ten stages, the spread of A_5^+ quickly outpaces W_4^{90}. This means that the third family in some sense has the best of both worlds: a relatively high local path redundancy followed by a large long-range spread. While this may seem obviously superior, there still is no perfect graph. Rather, one must understand the needs of a particular application and select a topology appropriately. This research attempts simply to unify some of the existing design decision and to provide intermediate alternatives.

REFERENCES

[1] T. R. Bewley, P. Belitz, & J. B. Cessna (2011) New Horizons in Sphere-Packing Theory, Part I: Fundamental Concepts & Constructions, from Dense to Rare. *Submitted*.

[2] P. Belitz, & T. R. Bewley (2011) New Horizons in Sphere-Packing Theory, Part II: Lattice-Based Derivative-free Optimization via Global Surrogates. *Submitted*.

[3] Cessna, J, & Bewley, T (2009) Honeycomb-Structured Computational Interconnects and Their Scalable Extension to Spherical Domains *SLIP'09*, July 26-27, San Francisco.

[4] Dally, WJ, & Towles, B (2003) *Principles and Practices of Interconnection Networks*. Morgan Kaufmann.

[5] Duato, J, Yalmanchili, S, & Ni, L (1997) *Interconnection Networks; an Engineering Approach*. IEEE Computer Society Press.