Broadcasting in Hypercycles*

Vassilios V. Dimakopoulos (dimako@ece.uvic.ca)
Nikitas J. Dimopoulos (nikitas@ece.uvic.ca)

Department of Electrical and Computer Engineering,
University of Victoria, P.O. Box 3055,
Victoria, B.C., CANADA, V8W 3P6

Abstract

Hypercycles is a class of multidimensional multiprocessor interconnection networks which includes hypercubes, toruses, rings and other related topologies. In this paper we consider the one-to-all communication problem for the general class of hypercycles. We present a nonredundant broadcasting algorithm which is completed in the minimum number of steps. The algorithm is given in distributed form, directly suggesting an efficient hardware implementation.

*This research was supported in part through grants by NSERC, IRIS and the University of Victoria.

1 Introduction

In a multiprocessor, one-to-all communication involves sending a message from a root node to all the other nodes through the underlying interconnection network. Such a communication primitive is necessary for system-level maintenance as well as for a big number of parallel applications, such as linear algebra algorithms [1].

A number of network topologies have been suggested for the interconnection between the processors. Hypercycles [8, 6] is a class of multidimensional graphs that includes such widely studied networks as the hypercubes, k-ary n-cubes, generalized hypercubes [3], toruses, and rings. Many properties and algorithms used for example in routing can be extended to the entire class of hypercycles, making it possible to choose a topology that best suits system requirements.

In this work, we study the broadcasting problem as applied to the entire class of hypercycles. We develop an algorithm that is nonredundant (i.e. no node receives the message more than once) and time-optimal (i.e. terminates in the minimum possible number of steps). Moreover, the algorithm is given in distributed form, suitable for direct implementation.

Similar algorithms have been given for the hypercubes [11], the generalized hypercubes [3] and other networks. Broadcasting in hypercycles differs from the ones given in [11, 3] because in the corresponding networks each dimension is a complete graph so that a message can be transmitted to all nodes in one step and with no redundancy. Hypercycles may not be fully connected in every dimension, making repetitions, because of their circular nature, a problem.

The paper is organized as follows: section 2 introduces the hypercycles and some of their graph theoretic properties; section 3 develops the backbone of our broadcasting scheme: broadcasting in a single dimension. In section 4 we give the multidimensional broadcasting algorithm and section 5 concludes the work.

2 Hypercycles

2.1 Mixed-radix system

The *mixed-radix system* [2] is a generalization of base b representation, where each digit of a number may have a different base. Given a number N, factored into r factors $m_1, m_2, ..., m_r$ as

$$N = m_r \times m_{r-1} \times \cdots \times m_1,$$

any number x, $0 \leq x \leq N-1$, can be written as an r-tuple

$$(x_r x_{r-1} \ldots x_1)_{m_r m_{r-1} \ldots m_1}$$

where m_i is the base of x_i and $0 \leq x_i \leq m_i - 1$ $(i = 1, 2, ..., r)$. The corresponding decimal number can be found as follows:

$$x = \sum_{i=1}^{r} x_i w_i$$

where $w_i = m_{i-1} \times m_{i-2} \times \cdots \times m_1$, the *weight* of the ith digit (w_1 is always 1). As an example, let $N = 24 = 3 \times 4 \times 2 = m_3 \times m_2 \times m_1$. Then $w_1 = 1, w_2 = 2, w_3 = 8$, and

$$(231)_{3,4,2} = 2 \times w_3 + 3 \times w_2 + 1 \times w_1 = 1 + 6 + 16 = 23.$$

If the bases of the mixed radix system are clear from context, we are going to omit them in order to simplify the notation, i.e.

$$(x_r x_{r-1} \ldots x_1) \equiv (x_r x_{r-1} \ldots x_1)_{m_r m_{r-1} \ldots m_1}.$$

Finally, we note that if $m_1 = m_2 = \cdots = m_r = b$, then the corresponding weights become $w_i = m_{i-1} \times m_{i-2} \times \cdots \times m_1 = b^{i-1}$, i.e. we obtain the standard base b representation.

2.2 Hypercycles and their properties

A hypercycle is a graph $G^{\rho_r,\rho_{r-1},\ldots,\rho_1}_{m_r,m_{r-1},\ldots,m_1} = (V,E)$ with N nodes where N is factored as $N = m_r \times m_{r-1} \times \cdots \times m_1$, generating a mixed radix system. The addresses of the nodes range from 0 to $N-1$ hence $V = \{0,1,\ldots,N-1\}$. After representing the address of each node in the mixed-radix system generated by the factors of N, the edge set is determined by $\rho_1,\rho_2,\ldots,\rho_r$, where

$$\rho_i \leq m_i/2,\ i = 1,2,\ldots,r$$

as follows: if $a = (a_r a_{r-1} \ldots a_1) \in V$ and $b = (b_r b_{r-1} \ldots b_1) \in V$ then $(a,b) \in E$ if and only if:

$$\begin{array}{ll} \exists \xi_j, 1 \leq \xi_j \leq \rho_j,\ 1 \leq j \leq r: & b_j = (a_j \pm \xi_j) \bmod m_j \\ \text{and } \forall i,\ 1 \leq i \leq r,\ i \neq j & a_i = b_i \end{array} \tag{1}$$

Hypercycles consist of r dimensions with m_i nodes in the ith dimension. Dimension i constitutes a circulant graph [4] $C_{m_i}\langle 1,2,\ldots,\rho_i\rangle$. The circulant graph $C_{m_i}\langle 1,2,\ldots,\rho_i\rangle$, where $\rho_i \leq m_i/2$, is defined as an m_i-node graph where node j is adjacent to nodes

$$j \pm 1, j \pm 2, \ldots, j \pm \rho_i \ (\mathrm{mod} m_i).$$

Note that the above definition is in accordance with (1) assuming that the hypercycle has only one dimension ($r = 1$). Examples of hypercycles are given in Figure 1. In Figures 1(a) and (b) the hypercycles are one-dimensional, i.e. they are identical to the circulants $C_6\langle 1\rangle$ and $C_6\langle 1,2\rangle$, correspondingly.

For the above type of graph we know [4, 8] that it is regular with degree d_i and diameter D_i equal to

$$d_i = \begin{cases} 2\rho_i & \text{if } \rho_i < m_i/2 \\ 2\rho_i - 1 & \text{if } \rho_i = m_i/2 \end{cases} \tag{2}$$

$$D_i = \left\lceil \frac{\lfloor m_i/2 \rfloor}{\rho_i} \right\rceil \tag{3}$$

Every node in an r-dimensional hypercycle belongs to r such circulants thus making hypercycles regular graphs with degree [8]

$$d(G^{\rho_r,\rho_{r-1},\ldots,\rho_1}_{m_r,m_{r-1},\ldots,m_1}) = \sum_{i=1}^{r} d_i$$

and diameter

$$D = \sum_{i=1}^{r} D_i \tag{4}$$

where d_i and D_i are given by (2) and (3) respectively.

If $\rho_i = \lfloor m_i/2 \rfloor$, for all $i = 1, 2, \ldots, r$, then we obtain the generalized hypercube that uses the same factoring of N. For $m_1 = m_2 = \cdots = m_r = 2$ we get the r-dimensional hypercube. The k-ary cubes are also a subset of hypercycles, the k-ary r-cube obtained for $m_1 = m_2 = \cdots = m_r = k$ and $\rho_1 = \rho_2 = \cdots = \rho_r = 1$.

Hypercycles have routing properties that are similar to those of the hypercubes. The routing algorithm has been described and analyzed elsewhere [7, 5] and is not needed for our purposes here.

3 Broadcasting in a single dimension

We first consider a single dimension of a hypercycle; it consists of the circulant graph $C_m\langle 1, 2, \ldots, \rho\rangle$. We recall that the diameter of this graph is equal to

$$D = \left\lceil \frac{\lfloor m/2 \rfloor}{\rho} \right\rceil \tag{5}$$

An optimal broadcasting algorithm should take exactly D steps to complete.

Assume that in the circulant $C_m\langle 1, 2, \ldots, \rho\rangle$ the nodes are numbered 0 to $(m-1)$ in a clockwise manner and node 0 wants to broadcast a message to all the other nodes — see for example Figures 1(a), (b) and Figure 3. In the message we attach a *weight* field which is used as follows: when an intermediate node receives a broadcast message it checks its weight; if it is unity then the node stops the message transmission, otherwise it decrements the weight by 1 and sends the message to the node in distance ρ away, continuing in the direction the message had when it was received.

Begin by giving a weight of D and sending the message from node 0 to nodes $1, 2, ..., \rho$ in the clockwise direction; this is accomplished in one step. After exactly D steps, nodes $(D-1)\rho+1, (D-1)\rho+2, \ldots, (D-1)\rho+(\rho-1), D\rho$ will have received the message with weight 1 and the transmission stops in this direction. Assume now that node 0 also sent the message with weight a to nodes $m-1, m-2, \ldots, m-\rho$ in the counterclockwise direction. Then after exactly a steps, nodes $m-(a-1)\rho-1, m-(a-1)\rho-2, \ldots, m-a\rho$ will have received the message with weight 1.

We now want to make sure that no redundant transmissions are made, in other words, the counterclockwise paths should not meet the clockwise paths. From the above description, after D steps the furthest node reached in the clockwise direction is node $D\rho$. At the same time, after a steps, the furthest node reached visiting nodes in the counterclockwise direction from node 0, is node $m-a\rho$; $m-a\rho$ should be greater than $D\rho$. Solving for the *maximum* integer value of a, we obtain:

$$m-a\rho > D\rho \Rightarrow a < \frac{m}{\rho} - D \Rightarrow a = \begin{cases} \frac{m}{\rho} - 1 - D & \text{if } m/\rho \text{ is integer} \\ \lfloor \frac{m}{\rho} \rfloor - D & \text{if } m/\rho \text{ is not an integer} \end{cases}$$

or equivalently,

$$a = \left\lfloor \frac{m-1}{\rho} \right\rfloor - D.$$

Notice now that the transmission in the clockwise direction stopped at node $D\rho$ while in the opposite direction, it stopped at node $m-a\rho$. Between nodes $D\rho$ and $m-a\rho$ there are exactly $k = m-a\rho-D\rho-1$ nodes. Hence, after the ath step, k nodes should continue the transmission for one more step in the counterclockwise direction to let the above nodes receive the message. This can be accomplished by giving, in the beginning, weights of $(a+1)$ to the k closest to the origin nodes in the counterclockwise direction. The remaining nodes receive a weight of a.

Based on the above analysis, we give a nonredundant broadcasting algorithm for circulant graphs shown in Figure 2, which is completed in exactly D steps, as shown below. The node addresses in the algorithm are all $\bmod m$ and, a and k have the following values:

$$\begin{aligned} a &= \left\lfloor \frac{m-1}{\rho} \right\rfloor - D \\ k &= m - a\rho - D\rho - 1. \end{aligned} \tag{6}$$

Note that Algorithm 1 is clockwise preferential because of Case 1(a). Algorithm 1 can be written to be counterclockwise preferential by simply interchanging the occurrence of the terms 'clockwise' and 'counterclockwise'. The properties of the algorithms are identical because of the rotational invariance of the circulant graphs.

Theorem 1 *Algorithm 1 is a nonredundant broadcasting algorithm for the circulant graph $C_m\langle 1, 2, \ldots, \rho\rangle$.*

Proof. Without loss of generality (due to the symmetry of the graph) we can assume that the originating node is node 0. We consider broadcasting clockwise from node 0 (broadcasting in the counterclockwise direction is treated similarly). Case 1(a) of Algorithm 1 generates ρ paths by following the routes:

$$0 \rightarrow \begin{cases} 1 & \rightarrow & \rho+1 & \rightarrow & \cdots & \rightarrow & (D-1)\rho+1 \\ 2 & \rightarrow & \rho+2 & \rightarrow & \cdots & \rightarrow & (D-1)\rho+2 \\ \vdots & & & & & & \\ \rho-1 & \rightarrow & \rho+(\rho-1) & \rightarrow & \cdots & \rightarrow & (D-1)\rho+(\rho-1) \\ \rho & \rightarrow & 2\rho & \rightarrow & \cdots & \rightarrow & D\rho. \end{cases}$$

Node $D\rho$ is reached following the last path. *Any* number j between $(\rho+1)$ and $D\rho$ can be uniquely represented as $j = q\rho + r$, where $0 \le r < \rho$ is the remainder of the division by ρ. Since node j receives the message from node $j - \rho = (q-1)\rho + r$ and sends the message to node $j + \rho = (q+1)\rho + r$ (Case 2 of Algorithm 1), we see that the node addresses on any path have the same remainder. Hence, any node up to and including node $D\rho$ belongs to a path and the paths have no node in common.

Exactly the same holds for the paths formed in the counterclockwise direction from node 0.

To complete the proof we have to show that while the clockwise paths reach node $D\rho$, the counterclockwise paths reach node $D\rho + 1$. We distinguish two cases:

Case 1: $k = 0$

In this case the counterclockwise paths reach node $m - a\rho$:

$$0 \to m - \rho \to m - 2\rho \to \cdots \to m - a\rho$$

Since $k = 0$, from (6) we get

$$k = m - a\rho - D\rho - 1 = 0 \Rightarrow m - a\rho = D\rho + 1.$$

Case 2: $k \neq 0$

In this case the counterclockwise paths reach node $m - k - a\rho$ following the route:

$$0 \to m - k \to m - k - \rho \to \cdots \to m - k - a\rho$$

starting with weight $(a + 1)$. But

$$m - k - a\rho = m - (m - a\rho - D\rho - 1) - a\rho = D\rho + 1. \qquad \square$$

Theorem 2 *The broadcasting described in Algorithm 1 is completed in exactly D steps (i.e. minimum number of steps).*

Proof. Since we start with weights D, a, and possibly $(a+1)$, and after each step the weights are decreased by one, the algorithm is completed in $\max\{D, a\}$ or $\max\{D, a+1\}$ steps.

Case 1: $k = 0$

In this case we do not have weights of $(a + 1)$ so we have to show that $a \leq D$; we get

$$k = m - a\rho - D\rho - 1 = 0 \Rightarrow a - D = \frac{m - 1}{\rho} - 2D. \tag{7}$$

From (5) we obtain:

$$D = \left\lceil \frac{\lfloor m/2 \rfloor}{\rho} \right\rceil \geq \frac{\lfloor m/2 \rfloor}{\rho} \geq \frac{m-1}{2\rho} \tag{8}$$

and (7) gives $a - D \leq 0$.

Case 2: $k > 0$

In this case we need to show that $a < D$; working as in Case 1 we get

$$k = m - a\rho - D\rho - 1 > 0 \Rightarrow a - D < \frac{m-1}{\rho} - 2D$$

and from (8), $a - D < 0$. □

Four examples are shown in Figure 3.

A comment should be made now about the proposed broadcasting algorithm with respect to the implementation cost. The hardware requirements are minimal since all the intermediate nodes need only compare the received weight to 1 and possibly decrement it by 1. All the computations for the different weights have to be performed only once, and only in the originating node. This can be done easily in software or by hardwiring the values of a, k and D to the router subsystem. We describe our prototype implementation in detail in section 4.1.

4 The complete algorithm

The multidimensional broadcasting of a message to every node in a hypercycle is similar to the one used for hypercubes [11] and generalized hypercubes [3]. The complete procedure is formulated in Algorithm 2, in Figure 4, and is completed in the minimum number of steps, equal to the diameter of the hypercycle.

Assuming that the r dimensions of the hypercycle $G^{\rho_r,\rho_{r-1},\ldots,\rho_1}_{m_r,m_{r-1},\ldots,m_1}$ are ordered (in any fixed ordering), the originating node sends the message with a pair of weights attached to it, (wd, wc), where wd is used for propagating through the dimensions and wc is used for broadcasting within a single dimension (circulant graph). The nodes that receive the message with weights (wd, wc) send it to dimensions $i = 1, 2, \ldots, wd-1$ with weights (i, wc_i), where wc_i is computed according to case 1 of Algorithm 1, because the message has to be broadcast along the whole circulant graph that corresponds to dimension i. They also send it along dimension wd with weights $(wd, wc-1)$ according to case 2 of Algorithm 1 since the node is an intermediate node in the broadcasting along dimension wd. This way the message is received by every node with no redundancy. An example is shown in Figure 5.

An informal proof of the no-redundancy property can be stated as follows: assume without loss of generality that node $(00\cdots0)$ is the origin. Because Algorithm 1 is nonredundant (as used in the first condition of Case 2 in Algorithm 2), we can assume that each dimension behaves like a fully connected graph and all its nodes receive the message in one step; thus nodes $(a_r 00\cdots0)$, for all $a_r = 0, 1, \ldots, m_r - 1$, receive the message at the same time. The remaining transmissions originate from these nodes and route through strictly lower dimensions (second condition in Case 2 of Algorithm 2). Hence all nodes that will receive the message from node $(a_r 00\cdots0)$ *must* have their first address digit equal to a_r. Consequently, they could never receive the message from a node $(a'_r 00\cdots0)$, $a'_r \neq a_r$. A simple induction shows that nodes receiving the message from a node $(a_r a_{r-1}\cdots a_k 00\cdots0)$ cannot receive it from any node $(a'_r a'_{r-1}\cdots a'_k 00\cdots0)$ where for at least one i in $\{k, k+1, \ldots, r\}$, $a'_i \neq a_i$.

The only thing to note is that since the maximum value for wd is r and the maximum value for wc, when in dimension i, is the diameter D_i of the corresponding circulant graph (Theorem 2), the number of steps needed to complete the broadcasting for the whole hypercycle is

$$\sum_{i=1}^{\max\{wd\}} \max\{wc \text{ in dimension } i\} = \sum_{i=1}^{r} D_i$$

which according to (4) is equal to the diameter of the hypercycle. Hence the algorithm requires the minimum number of steps to complete.

4.1 An FPGA prototype

We have developed CoDeL [9], a novel hardware description language suitable for rapid prototyping of algorithmic machines. The compiler generates synthesizeable VHDL code which can be mapped to technologies like FPGAs and ASICs. Algorithms 1 and 2 were expressed as a CoDeL source program; The module was made completely programmable: during startup it inputs the machine's configuration, i.e. the number of dimensions, the m_i's, the ρ_i's and the D_i's. Up to four dimensions are supported, each dimension having up to 15 nodes, allowing for a maximum of $15^4 \approx 50$K nodes in total. The maximum allowable degree is 16. The constants a and k are calculated for each dimension according to (6) and stored in registers. The module then waits until a broadcast message is received whereby it generates the appropriate weights based on the the presented algorithms.

The resultant VHDL code was simulated for functional correctness and targeted to a XILINX 4013Q208 FPGA chip [12]. The statistics of the implementation were as follows: 2106 primitive XILINX cells, 62 used pins and 98% utilized area. With a clock of 25MHz the worst-case delay for the source and intermediate nodes (corresponding to a 4-dimensional graph of degree 16) is 5μs and 4μs, respectively. The delay drops to 0.6μs for an intermmediate node in a single-dimension graph (Case 2 of Algorithm 1). We expect the delays to be reduced significantly if the implementation technology is full custom VLSI, instead of XILINX gate arrays.

5 Conclusion

We presented a nonredundant broadcasting algorithm for the entire class of hypercycles. The algorithm was proven to be time-optimal and also quite efficient in terms of hardware requirements. Also, the algorithm was presented in fully distributed form, which leads to easy implementation.

In the course of our research we have already designed and manufactured a VLSI chip that implements parts of the router engine for hypercycles in Northern Telecom's CMOS 4S 1.2μ technology [10]. The broadcasting algorithm presented here has been implemented in FPGA technology using CoDeL, a novel hardware description language we have developed. We are currently using CoDeL to design the rest of the communication subsystem with our ultimate goal being a fully functional hypercycle network.

References

[1] **Bertsekas, D P and Tsitsiklis, J N** *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs (1989)

[2] **Bhuyan, L N and Agrawal, D P** 'Design and Performance of Generalized Interconnection Networks', *IEEE Trans. on Computers*, Vol. 32 (1983) pp. 1081 – 1090

[3] **Bhuyan, L N and Agrawal, D P** 'Generalized Hypercube and Hyperbus Structures for a Computer Network', *IEEE Trans. on Computers*, Vol. 33 (1984) pp. 323 – 333

[4] **Boesch, F and Tindell, R** 'Circulants and Their Connectivities', *Journal of Graph Theory*, Vol. 8 (1984) pp. 487 – 499

[5] **Dimopoulos, N J, Chowdhury, M, Sivakumar, R and Dimakopoulos, V** 'Routing in Hypercycles. Deadlock Free and Backtracking Strategies', *PARLE '92 Parallel Architectures and Languages Europe*, Paris, France (1992) pp. 973 – 974

[6] **Dimopoulos, N J, Radhakrishnan, S and Radvan, D** 'Routing and Processor Allocation on a Hypercycle-Based Multiprocessor', *Proc. International Conference on Supercomputing*, Cologne, Germany (1991) pp. 105 – 114

[7] **Dimopoulos, N J, Radvan, D and Li, K F** 'Performance Evaluation of the Backtrack to the Origin and Retry Routing for Hypercycle Based Interconnection Networks', *Proc. 10th International Conference on Distributed Systems*, Paris, France (1990) pp. 279 – 284

[8] **Dimopoulos, N J, Rasmussen, R D, Bolotin, G S, Lewis, B F and Manning, R M** 'Hypercycles, Interconnection Networks With Simple Routing Strategies', *Proc. Canadian Conference on Electrical and Computer Engineering*, Vancouver, B.C., Canada (1988) pp. 577 – 580

[9] **Sivakumar, R, Dimakopoulos V V and Dimopoulos N J** 'Design of a Programmable Controller for Hypercycle Based Interconnection Networks', *FPD'95, the 3rd Canadian Workshop on Field-Programmable Devices*, Montreal, Quebec, Canada, (May 1995) (*to appear*)

[10] **Sivakumar, R, Dimopoulos, N J, Dimakopoulos, V and Chowdhury, M** 'Implementation of the Routing Engine for Hypercycle Based Interconnection Networks', *Proc. Canadian Conference on VLSI*, Kingston, Ont., Canada (1991) pp. 6.4.1 – 6.4.7

[11] **Sullivan, H and Bashkow, T R** 'A Large Scale, Homogeneous, Fully Distributed Parallel Machine I', *Proc. 4th Symposium on Computer Architecture*, (1977) pp. 105 – 117

[12] **XILINX Corporation** *The Programmable Logic Data Book* (1994)

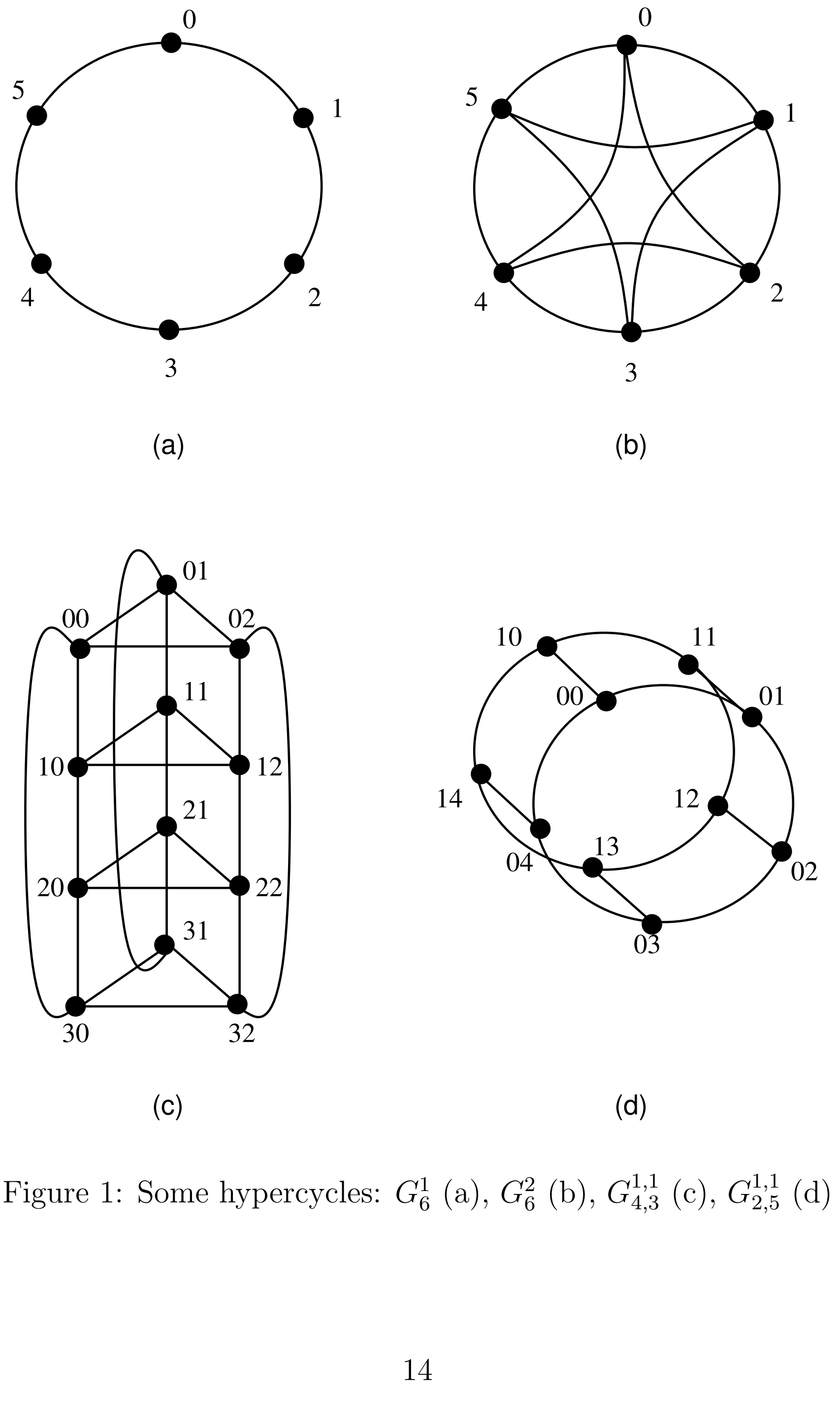

Figure 1: Some hypercycles: G_6^1 (a), G_6^2 (b), $G_{4,3}^{1,1}$ (c), $G_{2,5}^{1,1}$ (d)

Case 1: ORIGINATING NODE (node i)

- Send the message with weight D to all ρ nodes in the clockwise direction, i.e. nodes $i+1, i+2, \ldots, i+\rho$
- Send the message with weight $(a+1)$ to the first k nodes counterclockwise, i.e. nodes $i-1, i-2, \ldots, i-k$
- Send the message with weight a to the remaining $\rho - k$ nodes (counterclockwise), i.e. nodes $i-(k+1), i-(k+2), \ldots, i-\rho$

Case 2: THE OTHER NODES (wc is the weight of the received message)

s If $(wc = 1)$

then Stop

else
- $wc \leftarrow wc - 1$
- Send the message with the new weight wc to the node at distance ρ in the direction the message was received

Figure 2: Algorithm 1 (all node addresses are mod m)

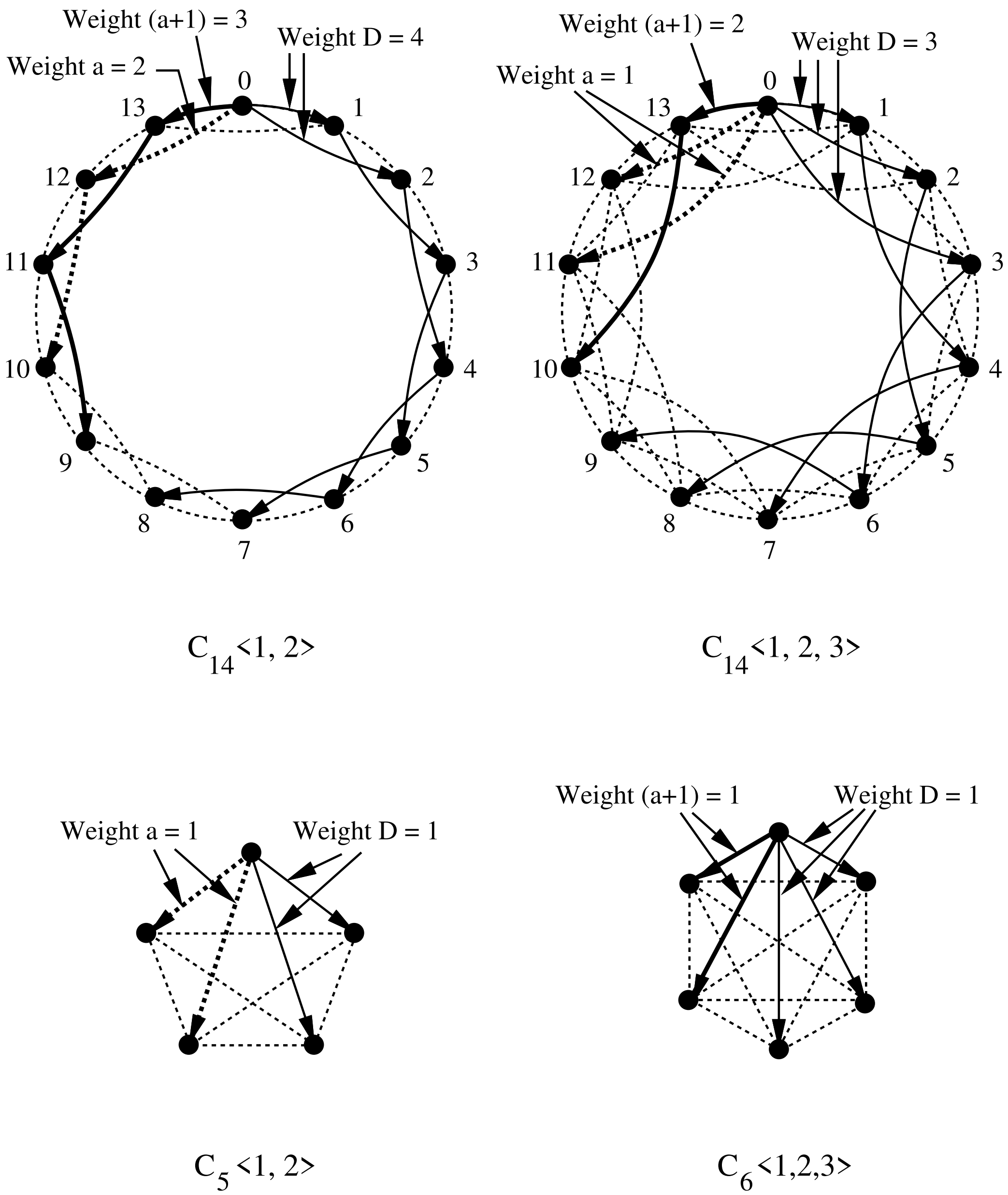

Figure 3: Broadcasting in circulant graphs

Case 1: ORIGINATING NODE

For every dimension $i = 1, 2, \ldots, r$ send the message with weight (i, wc_i) to all the adjacent nodes. The weight wc_i is to be calculated according to Case 1 of Algorithm 1, applied to the circulant graph of the ith dimension

Case 2: THE OTHER NODES ((wd, wc) is the weight of the received message)

- If $(wc > 1)$ then

  Within dimension wd send the message with weight $(wd, wc - 1)$ following Case 2 of Algorithm 1
- If $(wd > 1)$ then

  For every dimension $i = 1, 2, \ldots, wd - 1$ send the message with weight (i, wc_i) to the corresponding adjacent nodes. The weight wc_i is to be calculated according to Case 1 of Algorithm 1, applied to the circulant graph of the ith dimension

Figure 4: Algorithm 2

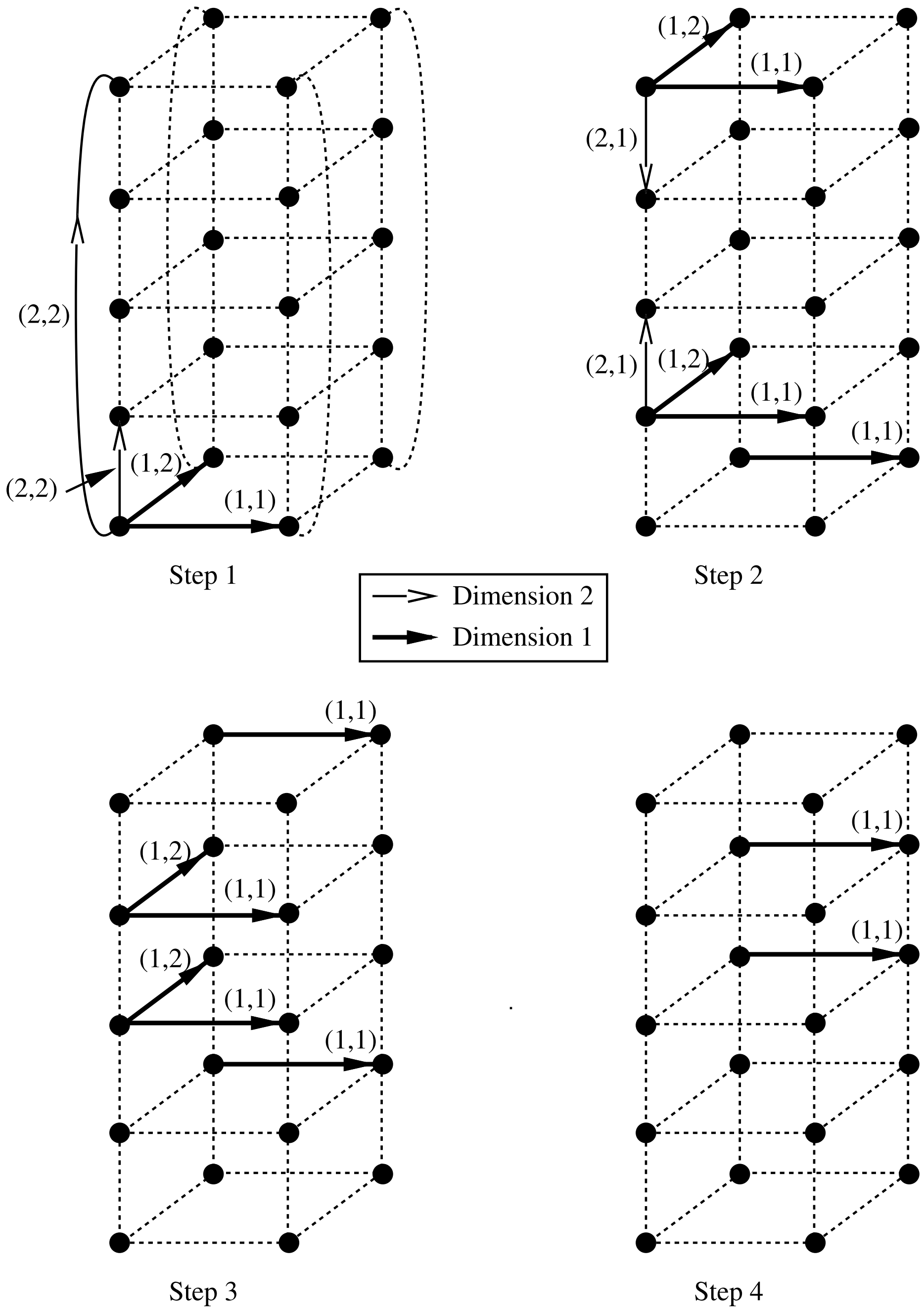

Figure 5: Broadcasting in the hypercycle $G^{1,1}_{5,4}$