Optimal Routing in 2-jump Circulant Networks

Borut Robič *

University of Cambridge Computer Laboratory
Systems Research Group
New Museums Site, Pembroke Street, Cambridge CB2 3QG, UK

May 1996

Abstract

An algorithm for routing a message along the shortest path between a pair of processors in 2-jump circulant (undirected double fixed step) networks is given. The algorithm requires $\Theta(d)$ time for preprocessing, and $\ell = \mathcal{O}(d)$ routing steps, where ℓ is the distance between the processors and d is the diameter of the network.

*Supported by the Partridge Visiting Fellowship and the Slovenian Science Foundation. On leave from Jožef Stefan Institute and Faculty of Computer and Information Science, University Ljubljana, Slovenia.

1 Introduction

Let Z denote the set of integers, and let $h_1, h_2, n \in Z$ such that $0 < h_1 < h_2 < n$. The set of integers modulo n is denoted by Z_n. Finally, $gcd(x, y)$ denotes the greatest common divisor of x and y.

Definition. An undirected graph $G(V, E)$, where V and E are the sets of vertices and edges respectively, is *2-jump circulant*, in symbols $G(n; \pm h_1, \pm h_2)$, if

- $V = Z_n$ and
- $E = \{\{u, v\} \in V^2 | v = u \pm h_1 (mod\, n) \; or \; v = u \pm h_2 (mod\, n)\}$.

Thus, vertices of $G(n; \pm h_1, \pm h_2)$ represent integers modulo n, and there are four edges incident to each vertex u: two *short* edges connecting it to vertices $u \pm h_1 (mod\, n)$, and two *long* edges connecting it to vertices $u \pm h_2 (mod\, n)$.

2-jump circulant graphs are special cases of *circulant* graphs which are in turn abstractions of *undirected multiple fixed step computer networks*[1] (with processors and communication links represented by vertices and edges, resp.). We will therefore assume that $G(n; \pm h_1, \pm h_2)$ is *connected*, i.e. there is a path between each pair of vertices, which is exactly when $gcd(h_1, h_2, n) = 1$ [4].

An algorithm is said to be *routing algorithm* (also referred to as a *two-terminal routing algorithm*) if, given a pair of vertices (i.e. processors), it routes a message from the source to the destination vertex. A routing algorithm is called *oblivious* if the route depends only on the source and the destination vertex. Recall that the *diameter* of a graph is the length of the longest path among the shortest paths between all pairs of vertices. Thus, a shortest routing path may be as long as the diameter d implying that the worst-case time complexity of any routing algorithm is bounded below by $\Omega(d)$. We call a routing algorithm *optimal* if its worst-case time complexity is $\mathcal{O}(d)$. In view of this, we state the following problem.

Problem. *Find an optimal routing algorithm for 2-jump circulant graphs.*

Recently, much attention has been payed to *double fixed step digraphs* (known as *double loop graphs*) [1, 2, 3, 4], which are *directed* 2-jump circulant graphs, i.e. each vertex u has two *outgoing* arcs $u \to u + h_1 (mod\, n)$ and $u \to u + h_2 (mod\, n)$, and two *incoming* arcs $u - h_1 (mod n) \to u$ and $u - h_2 (mod n) \to u$. It is clear, however, that any optimal routing algorithm for double fixed step digraphs cannot, in general, produce shortest routes when applied to 2-jump circulant graphs as the latter offer forward *and* backward moves. We are not aware of any optimal routing algorithm for 2-jump circulant graphs though several remarkable results on circulant graphs have recently appeared [1]. In particular, we will make use of the $\mathcal{O}(\log n)$ time algorithm for computing the diameter of a 2-jump circulant graph [7].

In what follows we describe an optimal oblivious routing algorithm for 2-jump circulant graphs.

2 Routing algorithm

A move along a short edge will be referred to as a *short hop*. Likewise, a *long hop* is a move along a long edge. A *positive* short hop is a move from u to $u+h_1(mod n)$, while a *negative* short hop is a move from u to $u-h_1(mod\, n)$, for some u. We define positive and negative long hops similarly. Due to vertex symetry of $G(n;\pm h_1,\pm h_2)$ a routing path between two vertices is determined up to a permutation of its hops. We may therefore assume that short hops precede long ones. This shows that a *shortest* routing path between two vertices consists of either positive or negative short hops, and similarly, of either positive or negative long hops (otherwise the path would not be the shortest one). Let the shortest path from u to v consist of $|a|$ short hops and $|b|$ long hops, where $a,b\in Z$. Here, we take $a>0$ *iff* short hops are positive, and $b>0$ *iff* long hops are positive. The routing algorithm is given below (recall that $sign(x)=1$ if $x\geq 0$ and $sign(x)=-1$ if $x<0$).

1.	$Compute(a,b)$;	... preprocessing
2.	**for** $i:=0$ **to** $\lvert a\rvert$ **do** $u:=u+sign(a)*h_1(mod\, n)$;	... short hops
3.	**for** $i:=0$ **to** $\lvert b\rvert$ **do** $u:=u+sign(b)*h_2(mod\, n)$;	... long hops

Algorithm $Route(u,v)$

Steps 2 and 3 take $|a|+|b|\leq d$ hops, where d is the diameter of G. The algorithm's worst-case time complexity is $\textit{Worst-case-Time}(Compute(a,b))+\mathcal{O}(d)$. We will show that $\textit{Worst-case-Time}(Compute(a,b))=\Theta(d)$ and thus prove $\Theta(d)$ to be worst-case time complexity of the algorithm $Route(u,v)$.

3 Preprocessing

The pair (a,b) is the integer solution of the congruence $u+xh_1+yh_2=v(mod\, n)$ which minimizes the sum $|x|+|y|$. Note however that, due to the vertex simetry of G, the shortest path from u to v is isomorphic to the shortest path from 0 to $w=v-u(mod\, n)$, $w\in Z_n$. Hence, (a,b) is the integer solution of the congruence

$$xh_1+yh_2=w(mod\, n) \tag{1}$$

that minimizes the sum

$$|x|+|y|. \tag{2}$$

Problem 1. *Find the solution $(x,y)\in Z^2$ of (1) that minimizes (2).*

A pair $(x,y)\in Z^2$ is a solution of (1) *iff* there exists a $k\in Z$ such that (x,y) is a solution of the Diophantine equation

$$xh_1+yh_2=w+kn. \tag{DE_k}$$

Since G is connected, there is a path between each pair of vertices u and v, implying that equation (1) admits a solution in Z^2 for every w. Consequently, for every w there is a $k\in Z$ such that (DE_k) has a solution in Z^2. Actually, such a (DE_k) has infinitely many integer solutions but we are interested in the one, denoted by (x_k,y_k), that minimizes (2). Since equation (DE_k) may be solvable in Z^2 for several $k\in Z$, several pairs (x_k,y_k) may exist—among these the one minimizing (2) is our final solution (a, b). Thus, we restate the Problem 1 as follows.

Problem 1′. *Among the solutions (x_k, y_k) of (DE_k) find the one that minimizes (2).*

Questions. In order to solve Problem 1′, we must answer to the following questions:
A. When is (DE_k) solvable in Z^2 ?
B. How do we compute solution (x_k, y_k) of a solvable (DE_k)?
C. How many solutions (x_k, y_k) must be computed in order to find the one that minimizes (2)?

We denote the fact that x divides y by $x\backslash y$. Let $h = gcd(h_1, h_2)$, $h_1^{'} = \frac{h_1}{h}$ and $h_2^{'} = \frac{h_2}{h}$. Since $0 < h_1 < h_2 < n$, we also have $0 < h_1^{'} < h_2^{'}$.

3.1 Answer to A.

Theorem 1. *Equation (DE_k) is solvable in Z^2 iff $h\backslash w + kn$.*

Proof. See [6, p.96] **qed.**

3.2 Answer to B.

Assume that (DE_k) is solvable in Z^2. According to [6, p.96], integer solutions of (DE_k) are exactly the pairs $(x_k(t), y_k(t))$, $t \in Z$, where

$$x_k(t) = X_k + h_2^{'}t \qquad and \qquad y_k(t) = Y_k - h_1^{'}t. \tag{3}$$

We find a *basic* solution (X_k, Y_k) as follows. Since it is a solution, it must be that $X_k h_1 + Y_k h_2 = w + kn$. Division by h yields $X_k h_1^{'} + Y_k h_2^{'} = W_k$, where $W_k = \frac{w+kn}{h}$. But $gcd(h_1^{'}, h_2^{'}) = 1$, so the equation

$$ph_1^{'} + qh_2^{'} = 1 \tag{4}$$

admits an integer solution [6, p.37], say $(p, q) \in Z^2$. Multiplying (4) by W_k yields $pW_k h_1^{'} + qW_k h_2^{'} = W_k$, which in turn reveals that $X_k = pW_k$ and $Y_k = qW_k$, i.e.

$$X_k = p\frac{w + kn}{h} \qquad and \qquad Y_k = q\frac{w + kn}{h}. \tag{5}$$

Note that the solution (p, q) of (4) is *independent* of k, so it may be computed only once. (Extended Eucliud algorithm [5, p.325] finds (p, q) in $\mathcal{O}(\log n)$ time.)

Knowing (X_k, Y_k) we could systematically search the solutions (3) at $t = 0, \pm 1, \pm 2, \ldots$ for the one minimizing (2), i.e. for the solution (x_k, y_k) (though the termination condition of the search is not clear yet). However, the following theorem shows that (x_k, y_k) can be found directly.

Theorem 2. *(x_k, y_k) is obtained from (3) either at $t = t_L = \lfloor -\frac{X_k}{h_2^{'}} \rfloor$ or $t = t_R = \lceil -\frac{X_k}{h_2^{'}} \rceil$, depending on which of the two minimizes the sum (2), which is in this case*

$$|x_k| + |y_k| = \min\{|X_k + h_2^{'}t_L| + |Y_k - h_1^{'}t_L|,\ |X_k + h_2^{'}t_R| + |Y_k - h_1^{'}t_R|\}. \tag{6}$$

Proof. Function $f(z) = |X_k + h_2' z| + |Y_k - h_1' z|$, $z \in R$, has the minimum at $z_0 = -\frac{X_k}{h_2'}$. To see this we show that $f(z_0 \pm \varepsilon) > f(z_0)$, $\forall \varepsilon > 0$. Now $f(z_0 \pm \varepsilon) = |X_k + h_2'(z_0 \pm \varepsilon)| + |Y_k - h_1'(z_0 \pm \varepsilon)| = |\pm \varepsilon h_2'| + |Y_k + \frac{h_1'}{h_2'} X_k \mp \varepsilon h_1'| \geq |\pm \varepsilon h_2'| + |Y_k + \frac{h_1'}{h_2'} X_k| - |\mp \varepsilon h_1'| = \varepsilon h_2' + f(z_0) - \varepsilon h_1' = f(z_0) + \varepsilon(h_2' - h_1') > f(z_0)$, since $h_2' - h_1' > 0$. Since z_0 is a real number, we choose between its integer neighbors $t_L = \lfloor z_0 \rfloor$ and $t_R = \lceil z_0 \rceil$, depending on the value of f at t_L and t_R. **qed.**

However, (x_k, y_k) is *admissible* only if $|x_k| + |y_k| \leq d$ where d is the diameter of G; otherwise, (DE_k) admits no adequate solution (x_k, y_k).

3.3 Answer to C.

Theorem 3. *At most $4d$ solutions (x_k, y_k) must be computed.*

Proof. We are interested only in admissible solutions (x_k, y_k), i.e. those satisfying the relation $|x_k| + |y_k| \leq d$. Let us find an upper bound on $|k|$. If a solution is admissible then $|x_k| \leq d$ and $|y_k| \leq d$. From (DE_k) we find that $|k|n = |kn| = |x_k h_1 + y_k h_2 - w| \leq h_1|x_k| + h_2|y_k| + w \leq (h_1 + h_2)d + n - 1 \leq 2nd + n - 1 < 2nd + n$. Hence $|k| < 2d + 1$, that is, $|k| \leq 2d$. **qed.**

4 Algorithm $Compute(a, b)$

The following algorithm finds (a, b) that describes the shortest routing path from u to v.

A $w := v - u (mod\, n)$, where $0 \leq w < n$;
B $h := gcd(h_1, h_2)$; $h_1' := h_1/h$; $h_2' := h_2/h$;
C   Find (p, q) from eq. (4);
D   Find the diameter d of G;
E $(a, b) := (d + 1, d + 1)$;
F   **for** $k := -2d$ **to** $2d$ **do**
G   **if** $h \backslash w + kn$ **then**
H   **begin**
I   Find X_k and Y_k according to eq. (5);
J   Find $|x_k| + |y_k|$ according to eq. (6);
K   **if** $|x_k| + |y_k| \leq d$ **then**
L   **if** $|x_k| + |y_k| < |a| + |b|$ **then**
M $(a, b) := (x_k, y_k)$
            end.

Algorithm $Compute(a, b)$

Computational complexity. Steps A..C can be performed in $\mathcal{O}(\log n)$ time [5]. Step D also takes $\mathcal{O}(\log n)$ time [7]. Loop F is entered $4d = \Theta(d)$ times (Theorem 3) while the body G..M takes $\mathcal{O}(1)$ time (Theorem 2). Hence, the overall time complexity is $\mathcal{O}(\log n) + \Theta(d)$. Since $d = \Omega(\sqrt{n})$ [1, 7], the first term is absorbed by the second one. $Compute(a, b)$ has (worst-case) time complexity $\Theta(d)$.

5 Improved algorithm $Compute(a, b)$

In this section we describe an improvement of the above algorithm that is based on two observations. First, we show that not every equation (DE_k) has to be checked for the solvability since only every h-th equation is solvable in Z^2 (Observation 1). Second, we improve Theorem 3 by showing that at most $2d$ (instead of $4d$) equations (DE_k) have to be considered (Observation 2). In spite of that, the *asimpthotic* time complexity of the algorithm $Compute(a, b)$ remains $\Theta(d)$.

Observation 1. *Only every h-th (DE_k) is solvable in Z^2.*

Proof. Let $h \backslash w + kn$ and let $\Delta k \geq 1$ be the smallest positive integer such that $h \backslash w + (k + \Delta k)n$. We show that $\Delta k = h$. From $h \backslash w + kn$ and $h \backslash w + (k + \Delta k)n$ it follows that $h \backslash \Delta k \cdot n$. For Δk to be the *smallest* such integer, we must have $\Delta k \cdot n = lcm(h, n)$, where $lcm(h, n)$ is the least common multiplier of h and n. Since $lcm(h, n) = \frac{hn}{gcd(h,n)}$, we have $\Delta k = \frac{h}{gcd(h,n)}$. But $gcd(h, n) = gcd(gcd(h_1, h_2), n) = gcd(h_1, h_2, n) = 1$ since G is connected. Hence $\Delta k = h$. **qed.**

Observation 2. *At most $2d$ solutions (x_k, y_k) must be computed.*

Proof. If $h_2 > \lfloor \frac{n}{2} \rfloor$, then we may view each pair of long edges $\pm h_2$ as a pair of edges $\mp(n - h_2)$. More precisely, a pair of edges $(u, u \pm h_2 (mod\, n))$ can be viewed as a pair of edges $(u, u \mp (n - h_2)(mod\, n))$, for each vertex u. We can do similarly with short edges if $h_1 > \lfloor \frac{n}{2} \rfloor$. As a result, we may assume w.l.g. that $0 < h_1 < h_2 \leq \lfloor \frac{n}{2} \rfloor$. But then $h_1 + h_2 < n$ and the proof of Theorem 3 ends with $|k| \leq d$. **qed.**

We can now combine these observations to design a more efficient algorithm $Compute(a, b)$. Let k_0 be absolutely the smallest k such that (DE_k) is solvable in Z^2. (The value of k_0 can be found in $\mathcal{O}(h)$ time by checking consecutive values of $k = 0, 1, -1, 2, -2, \ldots$ until the first is encountered satisfying the condition $h \backslash w + kn$. Observation 1 guarantees that this process terminates and returns k_0 in at most h steps.) Knowing k_0 and combining observations 1 and 2 we find that only equations (DE_{k_i}), $i = 0, \pm 1, \pm 2, \ldots, \pm \lfloor \frac{d}{h} \rfloor$ must be solved, where $k_i = k_0 + ih$. In particular, the basic solution of (DE_{k_0}) is

$$X_{k_0} = p \frac{w + k_0 n}{h} \qquad and \qquad Y_{k_0} = q \frac{w + k_0 n}{h}, \tag{5a}$$

while the basic solutions of other (DE_{k_i}), $i = \pm 1, \pm 2, \ldots, \pm \lfloor \frac{d}{h} \rfloor$ are

$$X_{k_i} = X_{k_0} + ipn \qquad and \qquad Y_{k_i} = Y_{k_0} + iqn, \tag{5b}$$

since $X_{k_i} = p \frac{w + k_i n}{h} = p \frac{w + (k_0 + ih)n}{h} = p \frac{w + k_0 n}{h} + ipn$, and similarly for Y_{k_i}.

Using these basic solutions we compute solutions (x_{k_i}, y_{k_i}) and values $|x_{k_i}| + |y_{k_i}|$ as described in Theorem 2. The improved algorithm $Compute(a, b)$ is given below.

A $w := v - u(mod\, n)$, where $0 \le w < n$;
B $h := gcd(h_1, h_2)$; $h_1' := h_1/h$; $h_2' := h_2/h$;
C Find (p, q) from eq. (4);
D Find the diameter d of G;
E $(a, b) := (d+1, d+1)$;
F Find k_0;
G Find X_{k_0} and Y_{k_0} according to eq. $(5a)$;
F **for** $i := -\lfloor d/h \rfloor$ **to** $\lfloor d/h \rfloor$ **do**
H **begin**
I Find X_{k_i} and Y_{k_i} according to eq. $(5b)$;
J Find $|x_{k_i}| + |y_{k_i}|$ according to eq. (6);
K **if** $|x_{k_i}| + |y_{k_i}| \le d$ **then**
L **if** $|x_{k_i}| + |y_{k_i}| < |a| + |b|$ **then**
M $(a, b) := (x_{k_i}, y_{k_i})$
end.

Improved algorithm $Compute(a, b)$

Acknowledgement. I am much obliged to Janez Žerovnik who brought this problem to my attention. I am also grateful to my host fellow Dr.Simon Crosby, Kam H. Shum, other members of the Systems Research Group at the Computer Laboratory, as well as the fellows of the Fitzwilliam College, for making my stay as a visiting fellow so pleasant.

References

[1] J.-C. Bermond, F. Comellas, D.F.Hsu. Distributed loop computer networks: A survey, *J. Parallel Distr. Comput.* **24**, pp.2-10, 1995.

[2] Y. Cheng, F.K. Hwang. Diameters of weighted double looop networks. *J. Algorithms* **9**, pp.401-410, 1988.

[3] D.-Z. Du, D.F.Hsu (Eds.) *Combinatorial network theory*, Kluwer Academic Publ., 1996.

[4] M.A. Fiol,, J.L.A. Yebra, I. Alegre, M. Valero. A discrete optimization problem in local networks and data alignment, *IEEE Trans. Comput.* **C-36**, pp.702-713, 1987.

[5] D.E. Knuth. *The art of computer programming* Vol. 2: Seminumerical algorithms, Addison-Wesley, 1969.

[6] B.M. Stewart. *Theory of numbers.* The McMillan, 1964.

[7] J. Žerovnik, T. Pisanski. Computing the diameter in multiple-loop networks. *J. Algorithms* **14**, pp.226-243, 1993.