Algorithms for Permutation Groups and Cayley Networks

Kenneth D. Blaha

CIS-TR-89-14
September 6, 1989

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
UNIVERSITY OF OREGON

APPROVED: ______________________

Dr. Eugene M. Luks

We use subgroup towers and SGSs to construct Cayley networks with "failsoft" routing algorithms, and we adapt Valiant's permutation routing algorithm to run on the directed Cayley networks. Normal towers are used to define Cayley graphs and routing algorithms that perform well, as long no more than $d-1$ processors fail (d the degree of the graph). For several Cayley network families we exhibit a universal broadcast algorithm that runs in optimal time.

These same techniques are used to analyze nonsymmetric networks. In particular, we prove that Leland and Solomon's moebius graph is isomorphic to a quotient Cayley graph. This information is used to efficiently compute minimum routes and determine the diameter of the moebius graph.

AWARDS AND HONORS:

Tektronix Fellowship Award, 1988-89

PUBLICATIONS:

Blaha, K. Minimum bases for permutation groups: The greedy approximation. Tech. Rep. CIS-TR-86-16, University of Oregon, 1986.

Blaha, K. Finding a minimum base for permutation groups is NP-hard. In *Congressus Numerantium* (1987), vol. 58, pp. 141–150.

DEDICATION

To my family

CHAPTER I

INTRODUCTION

Motivation and Overview

Over the last thirty years there has been considerable effort focused on the development of algorithms for permutation groups. Since the size of a permutation group G on n points can be exponential in n, it is usually necessary to specify the group by a set of generators. Fortunately, a generating set of size $O(n)$ exists for every permutation group $G \leq Sym(n)$. Given such a succinct representation for G the question arises as to whether we can find efficient solutions to basic queries about the group. Using a base and strong generating set (SGS) Sims devised a method of storing the group that satisfies the following properties:

(a) it uses no more than a poly(n) (polynomial in n) space

(b) given any $g \in Sym(n)$ we can determine in poly(n) time if $g \in G$

(c) we can run through the elements of G one at a time without repetitions

The base is used to define a subgroup tower $G = G^0 \geq G^1 \geq G^2 \geq \cdots \geq G^k = \{id\}$, and the subgroup tower is used to define an SGS [41]. Once an SGS for the group is known membership can be determined easily by "sifting" through the coset representatives (a complete description of the sifting process is given in the next section). Although Sims described a practical algorithm for computing an SGS, it was not until 1980 [20] that a version of Sims' algorithm was proved to run in polynomial time $O(n^6)$.

Shortly after that, Knuth suggested a clever implementation of Sims' algorithm that he analyzed to run in $O(n^5 \log\log n)$ time [26]. Using Babai's result [3], which gives a linear limit on the length of subgroup chains in $Sym(n)$, this bound was later improved to $O(n^5)$. Subsequently, Jerrum described an algorithm for computing a base and SGS with in polynomial time, but not by the Greedy algorithm.

Under the assumption that P≠NP we know that there is no polynomial time solution to the minimum base problem. It is natural to ask if the Greedy algorithm is a good approximation heuristic. In Chapter III we answer this question by comparing the size of a Greedy base to the size of a largest nonredundant base, and to the size of a minimum base. We also compare the Greedy algorithm to another heuristic that has been suggested for finding small bases.

So far our discussion has focused on sequential algorithms for permutation groups. McKenzie and Cook were among the first to study parallel algorithms for permutation groups [37]. They showed that for abelian groups, the fundamental problem of group membership (Is $g \in G$?) is in NC. They also conjectured that the membership problem is P-complete (complete for P with respect to logspace reductions) for arbitrary groups, and hence not likely to be in NC.

This conjecture was based on the following observations. First, the sifting process used in membership testing seems inherently sequential. Second, the tower of subgroups generated by a base may have length linear in n. It was later shown through a series of papers, [35], [34] and [4], that membership is, in fact, in NC. However, the tower of subgroups used to perform the sifting was quite different from the tower produced by a base.

As a consequence of this work it was shown that a base and SGS could be constructed in NC, but the parallel sifting problem remained open [4]. In Chapter IV we address this problem. We show that there exists a base and SGS for which the sifting problem is P-hard. In contrast to this result, we show that for solvable groups and polynomial subgroup towers one can always find an SGS for which sifting is in NC. In this chapter we also answer the following question, " Is there a parallel algorithm for computing Greedy bases?" We show that it is highly unlikely that such an algorithm exists by proving that the deterministic Greedy base problem is P-complete.

In the second part of this dissertation we show how towers of subgroups and SGSs can be used to study parallel processing networks. The interconnection topology of a

Since then it has been noted that many of the families of regular graphs mentioned above may be viewed as Cayley graphs [11]. Thus there was a common thread that linked these networks together. Carlsson, Cruthirds, Sexton and Wright used the algebraic structure of the Cayley graph to generalize the CCC and produce new graphs that were more dense then any known to date. Not only were these graphs dense, but since they were Cayley graphs, they were symmetric and the authors believed they would have the same desirable properties that the other networks possessed. Although the authors believed that a good routing algorithm for these graphs should exist, none was presented. The question is, how does one take advantage of the underlying group structure to find good routing algorithms?

The problem of finding a minimum route on a Cayley graph is related to the problem of finding a minimal length generating sequence (MLGS) (smallest word in the generators) of an arbitrary group element. Goldreich and Even showed that MLGS is NP-hard [17]. Since then Jerrum has shown that it is PSPACE-complete [24]. Consequently, it would be unreasonable to attempt to solve the routing problem for an arbitrary group and an arbitrary set of generators.

Thus not only is the choice of the group important when constructing Cayley graphs, but the choice of generators is a critical consideration. In Chapter V we show how SGSs can be used to construct Cayley graphs with failsoft routing algorithms, and how Valiant's permutation routing algorithm [44] can be adapted to run on the directed Cayley networks. We also show how normal towers can be used to define Cayley graphs and routing algorithms that perform well, as long as no more than $d-1$ processors fail. We conclude this section with several examples of Cayley networks. In one of the examples the underlying group is a Sylow-2 subgroup of the symmetric group on n elements (n a power of two). In this case the generators are chosen with great care so that sifting could be applied. Also, techniques from Jerrum [25] are used to reduce the size of the generating set without significantly increasing the diameter of the Cayley graph.

In Chapter VI we extend Faber's work on universal broadcast schemes [18]. In particular, we show how certain Cayley networks constructed from SGSs can perform uni-

If $H \leq G$, then we define an equivalence relation on G in which two elements $g, h \in G$ are equivalent if and only if $gh^{-1} \in H$. The equivalence classes of G under this relation are called the cosets of H in G. We say that g is a coset representative of the (right) coset $Hg = \{hg | h \in H\}$. A set $U \subseteq G$ of size $|G : H|$ is a complete set of coset representatives of H in G if every element in G is equivalent to a (unique) element in the set. We say that H is normal in G, $H \trianglelefteq G$, if $Hg = gH$ for all $g \in G$.

By the degree of $G \neq \{id\}$ we mean the number of points moved by G. Let $\omega \in \Omega$, then we call $\{\omega^g | g \in G\}$ the G-orbit of ω. We say that $\Delta \subseteq \Omega$ is fixed by G if $\Delta^g = \Delta$ for all $g \in G$. Each $g \in G$ induces a permutation on Δ, which we denote by g^Δ. We call the totality of the g^Δ's formed for all $g \in G$ the constituent, G^Δ, of G on Δ.

For $A \subseteq \Omega$, we define the set $G_A = \{g \in G | \forall a \in A,\ a^g = a\}$, called the point-wise set stabilizer of A. If A consists of a single point, a, then we write $G_A = G_a$.

For any abstract group H, we define the homomorphism $\mathcal{R} : H \rightarrow \mathrm{Sym}(H)$ such that $\mathcal{R}(h)$ acts on H via right multiplication. That is, for each "point" $x \in H$, $x^{\mathcal{R}(h)} = xh$. We call $\mathcal{R}(H)$ the right regular representation of H. For additional background information on permutation groups see either [36], or [47].

The following definitions are due to Sims and may be found in [42]. A base for G is a sequence of points $B = b_1, b_2, \ldots, b_k$, $b_i \in \Omega$, such that the only element in G fixing all of the b_i is the identity. We say that base B has size k, and we denote the size of a smallest base for G by $\mathcal{M}(G)$. The tower of subgroups

$$G = G^0 \geq G^1 \geq \cdots \geq G^k = \{id\},$$

where $G^i = G_{\{b_1,\ldots,b_i\}}$, $1 \leq i \leq k$ is called the chain of stabilizers of G relative to B. We call a base nonredundant if each of the inclusions $G^{i-1} \geq G^i$ is proper.

This tower has three characteristics that are essential for computational purposes. First, the tower has length no more than $n - 1$. Second, $|G^{i-1} : G^i|$ is polynomial in n for $1 \leq i \leq k$ (infact, $|G^{i-1} : G^i| \leq n - i + 1$). Any subgroup tower satisfying this second condition is called polynomial. Third, given $g \in G$ we can determine in linear time if

Fact 1.3 If $G \leq Sym(\Omega)$, and $B = b_1, b_2, \ldots, b_k$ is a base for G, then (by Lagrange's Theorem) $|G| = |G^0 : G^1||G^1 : G^2| \cdots |G^{k-1} : G^k|$.

Lemma 1.4 Given $G = \langle\sigma\rangle \leq \mathrm{Sym}(n)$ and base $B = b_1, b_2, \ldots, b_k$, define r_i to be the size of the G-orbit (cycle of σ) containing b_i. Then $G^m = \langle\sigma^r\rangle$ where $r = lcm\{r_1, r_2, \ldots, r_m\}$, $1 \leq m \leq k$.

Proof: $G^m = \langle\sigma^j\rangle \Leftrightarrow j$ is the smallest positive integer such that σ^j fixes $b_1, \ldots, b_m$. But σ^j fixes $b_i \Leftrightarrow r_i$ divides j. □

We shall use the following construction in Chapters II, III and IV. Let X be a fixed finite set, and $\{\sigma_x | x \in X\}$ a fixed set of generators for the group $(Z_2)^{|X|}$. For $Y \subseteq X$ define $G(Y) = \langle\sigma_x | x \in Y\rangle$.

The right regular action $\mathcal{R} : G(Y) \rightarrow \mathrm{Sym}(G(Y))$ is extended to an action $\mathcal{R}_Y : G(X) \rightarrow \mathrm{Sym}(G(Y))$ in which $\{\sigma_x | x \in X \setminus Y\}$ act trivially. Suppose now that $\mathcal{C}$ is a collection of subsets of X. Let $\Omega_{\mathcal{C}} = \dot{\bigcup}_{Y \in \mathcal{C}} G(Y)$ (the disjoint union of the sets $G(Y)$, $Y \in \mathcal{C}$). Then the $\mathcal{R}_Y$ for $Y \in \mathcal{C}$, induce an action $\mathcal{R}_{\mathcal{C}} : G(X) \rightarrow \mathrm{Sym}(\Omega_{\mathcal{C}})$, where $\omega^{\mathcal{R}_{\mathcal{C}}(\sigma)} = \omega^{\mathcal{R}_Y(\sigma)}$ if $\omega \in G(Y)$. Note that, if $X = \bigcup_{Y \in \mathcal{C}} Y$, then $\mathcal{R}_{\mathcal{C}}$ is faithful. The reader may wish to examine the example given at the end of this chapter.

Now, given $Z \subseteq X$ we let $G_{\mathcal{C}}(Z)$ denote the permutation group $\mathcal{R}_{\mathcal{C}}(G(Z))$. We use the next two lemmas to analyze bases of $G_{\mathcal{C}}(Z)$.

Lemma 1.5 Let $Z \subseteq X, Y \in \mathcal{C}$ and $\omega \in G(Y)$, then $G_{\mathcal{C}}(Z)_\omega = G_{\mathcal{C}}(Z \setminus Y)$.

Proof: $G_{\mathcal{C}}(Z)_\omega = \mathcal{R}(\{\sigma \in G(Z) | \omega^{\mathcal{R}_{\mathcal{C}}(\sigma)} = \omega\} = \mathcal{R}(\langle\sigma_x | x \in Z \setminus Y\rangle)$.

The first equality follows from the definition of point stabilizer and the second from Fact 1.2. □

Lemma 1.6 Let $W = Y \cap Z$, where $Z \subseteq X$, and $Y \in \mathcal{C}$. Then the set $G(Y)$ has $|G(Y) : G(W)|$ $G_{\mathcal{C}}(Z)$-orbits each of size $|G(W)|$.

Proof: Let $H \leq L$, and $R : H \rightarrow \mathrm{Sym}(L)$ be a homomorphism such that H acts on L via right multiplication (i.e., $l^{R(h)} = lh$). Then the $R(H)$-orbits are the left cosets (lH)

```
(1) $i = 1$
(2) While $G^{i-1} \neq id$ do begin
    (2.1) Pick a point $b_i$ from a largest $G^{i-1}$-orbit
    (2.2) Using generators for $G^{i-1}$ compute a set of coset representatives
          for $G^i$ in $G^{i-1}$
    (2.3) Update the data structure
    (2.4) Compute a set of $O(n^2)$ Schreier generators for $G^i$
    (2.5) Reduce the Schreier generators to a set of $O(n)$ generators
          for $G^i$
    (2.6) $i = i + 1$
```

The running time of this algorithm is dominated by step (2.5). Thus, Jerrum's algorithm may be modified to include the Greedy heuristic without increasing the asymptotic running time of the algorithm.

We conclude this section with an example that serves two functions. First, it gives the reader a concrete example of how the set X and the collection C of subsets of X are used to construct the permutation group $G_C(X)$. Second, it shows that the Greedy algorithm fails to find a minimum base for the group $G_C(X)$.

Example 1.1 Let $X = \{a,b,c,d,e,f\}$, $Y_1 = \{a,b\}$, $Y_2 = \{c,d\}$, $Y_3 = \{e,f\}$, and $Y_4 = \{a,c,e\}$. Let $C = \{Y_1, Y_2, Y_3, Y_4\}$. Following the construction outlined above we define the following elementary abelian 2-groups: $G(X) = \langle \sigma_x | x \in X \rangle$, $G(Y_1) = \langle \sigma_a, \sigma_b \rangle$, $G(Y_2) = \langle \sigma_c, \sigma_d \rangle$, $G(Y_3) = \langle \sigma_e, \sigma_f \rangle$, $G(Y_4) = \langle \sigma_a, \sigma_c, \sigma_e \rangle$.

Recall that Ω_C is the disjoint union of the $G(Y_i)$ $i = 1,2,3,4$, and

$$G_C(X) \leq Sym(G(Y_1)) \times Sym(G(Y_2)) \times Sym(G(Y_3)) \times Sym(G(Y_4)).$$

The monomorphism $\mathcal{R}_C : G(X) \rightarrow Sym(\Omega_C)$ maps the generators of $G(X)$ to generators of $G_C(X)$: $\mathcal{R}_C(\sigma_a) = (\sigma_a, id, id, \sigma_a)$, $\mathcal{R}_C(\sigma_b) = (\sigma_b, id, id, id)$, $\mathcal{R}_C(\sigma_c) = (id, \sigma_c, id, \sigma_c)$, $\mathcal{R}_C(\sigma_d) = (id, \sigma_d, id, id)$, $\mathcal{R}_C(\sigma_e) = (id, id, \sigma_e, \sigma_e)$, $\mathcal{R}_C(\sigma_f) = (id, id, \sigma_f, id)$.

CHAPTER II

MINIMUM BASES FOR PERMUTATION GROUPS

We demonstrated, in Example 1.1, that the Greedy algorithm fails to find a minimum base. In the first section of this chapter we prove that the minimum base problem is, in fact, NP-hard. The corresponding decision problem of determining whether there exists a base of size at most N (for a given positive integer N) is NP-complete. Moreover, the problem remains NP-complete even if we restrict G to be either a cyclic group or an elementary abelian group with orbits size no more than 8.

We prove, in the next section, that for abelian groups this bound on the size of the orbits is sharp. That is, if G is an abelian group with orbits of size less than 8, then we can find a minimum base for G in polynomial time. Our algorithm uses Lovász's result [31], in which a maximum matching of a linear 2-polymatroid is found, in polynomial time, to handle abelian semisimple groups with orbits of size 4 and 6.

Finding Minimum Bases is NP-hard

We prove that the minimum base problem is NP-hard by showing that the corresponding decision problem is NP-complete. The decision problem small base (SB) is defined as follows:

SB Input: $G \leq \mathrm{Sym}(n)$ given by generators and a positive integer $N \leq n$.
Question: Does there exist a base for G of size no more than N?

We show that even when the group G is restricted to cyclic groups or elementary abelian groups, the SB problem is NP-complete. It is not difficult to show that the SB problem is in NP. Guess a base for G, $B = b_1, b_2, \ldots, b_k$, and check that $k \leq N$. Then use your favorite algorithm (Sims, Knuth, or Jerrum) to verify that B is a base for G.

Example 2.2 Let (Y, M) be an instance of Exact Two Cover where $Y = \{a, b, c, d\}$ and M contains the 2-sets: $\{b, d\}, \{b, c\}$ and $\{a, d\}$. If we mimic the construction in the proof of Theorem 2.1 and map a, b, c, d to the primes 2,3,5,7 respectively, then the permutation σ will have cycle decomposition consisting of 3 disjoint cycles of sizes 21, 15 and 14.

Using Lemma 1.4 one checks that a minimum base for $G = \langle\sigma\rangle$ is comprised of one point from the cycle of size 15 and one point from the cycle of size 14. The Greedy algorithm starts by fixing a point b in the G-orbit of size 21. The group $G_b = \langle\sigma^{21}\rangle$ has 10 nontrivial orbits, 3 of size 5 and 7 of size 2. The Greedy algorithm selects two more points; first, a point in an orbit of size 5 is fixed, and then a point in an orbit of size 2. Thus, the Greedy algorithm will always produces a base of size 3.

Of course it is quite easy to construct cyclic groups of smaller order and degree for which the Greedy algorithm fails. In fact, we can find examples that involve only two primes.

Example 2.3 Let G be the cyclic group generated by

$$\sigma = (1, 2, \ldots, 8)(9, 10, \ldots, 17)(18, 19, \ldots, 29).$$

The Greedy algorithm first fixes a point, say $b = 18$, in the G-orbit of size 12. By Lemma 1.4 $G_b = \langle\sigma^{12}\rangle$. Now G_b has 4 orbits of size 2 and 3 orbits of size 3. Next the Greedy algorithm fixes a point in a 3-orbit, and then a point in an orbit of size 2 resulting in a Greedy base of size 3. A minimum base for G has size 2 (e.g., $B = 1, 9$).

In the above reduction of X3C to SB the size of the orbits of the cyclic group increased, as the problem size of X3C increased. One might wonder if it is possible to solve the SB problem efficiently for groups that are restricted to have bounded orbits. Theorem 2.2 suggests that this is not the case.

Theorem 2.2 SB is NP-complete even if G is constrained to be an elementary abelian 2-group with orbits of size 8.

Algorithm AMB_7:

```
(1) Find a minimum base $B_1$ for the subgroup of $G$ that
    fixes all the points in orbits of size 5 and 7
(2) Find a minimum base $B_2$ for the Frattini subgroup of $G_{B_1}$
(3) Find a minimum base $B_3$ for $(G_{B_1B_2})^{\Delta}$, where $\Delta$
    is the union of all $G_{B_1B_2}$-orbits of size 4 and 6
(4) Find a minimum base for $G_{B_1B_2B_3}$
```

The following proposition describes how step (1) of the algorithm is accomplished, and proves that the base we find can be extended to a minimum base for G.

Proposition 2.4 In polynomial time we may reduce any instance of AMB_7 to the problem of finding a minimum base for an abelian permutation group G', where all of the G'-orbits have size 2,3,4 or 6.

Proof: Let $G = \langle \Phi \rangle$ be an instance of AMB_7. Let Δ_1 be the union of all the G-orbits of size 7, let Δ_2 be the union of all the G-orbits of size 5, and let Δ_3 be the union of the remaining G-orbits. Then by the fundamental theorem of abelian groups, $G = G^{\Delta_1} \times G^{\Delta_2} \times G^{\Delta_3}$. By raising the generators of G to the appropriate power we find generators for the groups G^{Δ_i}, $i = 1,2,3$.

$$G^{\Delta_1} = \langle \phi^{30} | \phi \in \Phi \rangle$$

$$G^{\Delta_2} = \langle \phi^{42} | \phi \in \Phi \rangle$$

$$G^{\Delta_3} = \langle \phi^{35} | \phi \in \Phi \rangle$$

Observe that B is a minimum base for G if and only if $B \cap \Delta_i$ is a minimum base for G^{Δ_i} $1 \leq i \leq 3$. Thus, to compute a minimum base for G it will suffice to compute a minimum base for each G^{Δ_i}, $i = 1,2,3$.

The groups G^{Δ_1} and G^{Δ_2} are elementary abelian p-groups (p a prime) with orbits of size p. By Facts 1.1 and 1.3 any nonredundant base for these groups is a minimum all the remaining G-orbits. By the fundamental theorem of abelian groups we know that there exists integers r, s and t such that $G \simeq (Z_4)^r \times (Z_2)^s \times (Z_3)^t$.

The function $F : G \rightarrow G$ defined by $F(h) = h^6$ is a homomorphism since G is abelian. The group $F(G)$ is called the Frattini subgroup of G. Note that $F(G) \leq G^{\Delta_1}$, and $F(G) \simeq (Z_2)^r$.

Any minimum base for G must contain r points from Δ_1 that constitute a base for $F(G)$. The first step of the reduction is to find r points $B = b_1, b_2, \ldots b_r$ such that B is a base for $F(G)$. This can be accomplished by running the Greedy algorithm on $F(G)$, or by running the Greedy algorithm on the group G^{Δ_1} and selecting the first r points that are fixed by the algorithm. The base B that we obtain for $F(G)$ is by no means unique. To finish the proof we must prove that any minimum base for $F(G)$ can be extended to a minimum base for G.

It will suffice to show that if B and B' are two minimum bases for $F(G)$, then $\mathcal{M}(G_B) = \mathcal{M}(G_{B'})$. The groups G_B and $G_{B'}$ are abelian semisimple (since they contain no elements of order 4), and both groups have order $2^s 3^t$. Hence, each group is isomorphic to $(Z_2)^s \times (Z_3)^t$. If we could prove that $G_B = G_{B'}$ we would be done. Unfortunately this statement is not true. Consider, for example, the group $\langle(1,2,3,4)(5,6,7,8)\ ,\ (1,2,3,4)(5,8,7,6)\rangle$ and let $B = 1$ and $B' = 5$. Instead we prove a weaker statement that is sufficient to imply that $\mathcal{M}(G_B) = \mathcal{M}(G_{B'})$.

By the proof of Proposition 2.5 we know that $\mathcal{M}(G_B)$ is completely determined by the action of the group on the G_B-orbits of size 4 and 6. All of the G_B-orbits in Δ_1 have size 2 or less. Thus $\mathcal{M}(G_B)$ is determined by the action of G_B on Δ_2. A similar argument holds for $G_{B'}$. To finish the proof we need only show that $G_B^{\Delta_2} = G_{B'}^{\Delta_2}$

Note that both $F(G) \cap G_B$ and $F(G) \cap G_{B'}$ are trivial. Thus, we may conclude that both $K = F(G) \times G_B$ and $K' = F(G) \times G_{B'}$ are subgroups of G. Both groups are elementary abelian and $|K| = |K'| = 2^{r+s}3^t$. Thus it follows that $K = ker(F) = K'$. Finally $F(G) \leq G^{\Delta_1}$ implies that $G_B^{\Delta_2} = ker(F)^{\Delta_2} = G_{B'}^{\Delta_2}$, as desired. □

Proposition 2.4 and proposition 2.6 describe the first two steps of the algorithm. Moreover, they prove that any partial base constructed by the execution of the first two circuit in (S, r), then there is a vector $v \in L$, $v \neq 0$ which is contained in the linear span of each circuit $X - X_i$. Projecting everything onto a hyperplane complementary to p, we get a projection compressing X. Lovász points out that the only step of the maximum matching algorithm for linear 2-polymatroids that does not generalize to all 2-polymatroids is the construction of the projection described above [32, page 212].

To perform step (3) of the minimum base algorithm we construct a 2-polymatroid (S, f), where S is a collection of subsets taken from a direct product of two linear spaces. The following lemma proves that we can find a maximum matching for (S, f) in polynomial time using Lovász's algorithm.

Lemma 2.7 Let $L = (Z_2)^r \times (Z_3)^s$. Let f be the integer valued function defined on the subsets of L such that, $f(X) = r_1 + s_1$ if and only if $\langle X \rangle \simeq (Z_2)^{r_1} \times (Z_3)^{s_1}$.

Let S be a collection of two element subsets of L such that $f(x) = 2$, for each $x \in S$. Then (S, f) is a 2-polymatroid, and a maximum matching for (S, f) can be found using Lovász's algorithm.

Proof: It may be convenient to think of L as the direct product of two vector spaces and to view the elements of L as $(s + r)$-tuples. It is a simple exercise to check that (S, f) is a 2-polymatroid. Let $D \subset S$ be a nontrivial double circuit with principal partition $\{D_1, D_2, \ldots, D_m\}$ and define $K_i = D - D_i$ for $1 \leq i \leq m$.

To verify that a maximum matching can be found using Lovász's algorithm we must prove that we can find a nontrivial element of $K_1 \cap K_2 \cap \cdots \cap K_m$ in polynomial time. Using the principal partition for D, we can compute in polynomial time a set of generators for $K_1 \cap K_2 \cap \cdots \cap K_m$.

To complete the proof we must guarantee that the intersection is nontrivial whenever the double circuit is nontrivial. Lovász uses an induction argument to prove that $K_1 \cap K_2 \cap \cdots \cap K_m$ is nonempty when the 2-polymatroid is linear [33, Lemma 11.3.3]. This same proof may be used for (S, f) by simply replacing each occurrence of the word, "dim" with the letter "f". □

Remark 2.9 Let G be an abelian permutation group for which all the orbits have size a prime or a product of two primes; then we can find a minimum base for G in polynomial time.

Proof: For any G-orbit O, we know that the constituent G^O is abelian and transitive. Thus, G^O is regular and $|G^O| = |O|$. We can modify Propositions 2.4 and 2.5 so that the orbits of prime order can be ignored. Proposition 2.6 is essentially the same except that $F(G)$ is abelian semisimple (the $F(G)$-orbits have prime order). We generalize Lemma 2.7 to handle polymatroids (S, f), where S is a direct product of (possibly more than two) linear spaces. □

Remark 2.10 The problem of finding a maximum matching of a linear 2-polymatroid over the field $GF(p)$ is polynomial time equivalent to the problem of finding a minimum base for an abelian semisimple group G with orbits of size p^2.

Proof: One direction of the remark is proved by Theorem 2.8. To prove the other direction we assume, without loss of generality, that the linear 2-polymatroid (S, f) is specified by the columns of a $m \times 2r$ matrix M over the field $GF(p)$. Mimicking the proof of Theorem 2.8 we use the rows of matrix M to generate a group G with r orbits of size p^2. The remark follows from equation II.1. □

- G has a nonredundant base of size at least $\frac{1}{3}\mathcal{M}(G)\log n$.

Proof: Suppose that $n \geq 8k^2$, and that r is the integer maximal with respect to $k2^r + 2rk \leq n$. Define $X = \{1, 2, \ldots, rk\}$, $X_i = \{r(i-1)+1, \ldots, ir\}$, and $Y_j = \{j\}$ for $1 \leq i \leq k$, $1 \leq j \leq rk$. Let $C = \{X_1, \ldots, X_k, Y_1, \ldots, Y_{rk}\}$.

Using the notation from Chapter I we define $G = G_C(X) \leq \mathrm{Sym}(\Omega_C)$. Then $G \simeq (Z_2)^{rk}$, and $|\Omega_C| = k2^r + 2rk$.

Since a largest G-orbit has size 2^r and $|G| = 2^{kr}$ it follows from Facts 1.1 and 1.3 that a minimum base for G must have size at least k. Let $A = a_1, a_2, \ldots a_k$ where $a_i \in G(X_i)$, then by Lemma 1.5 it follows that A is a minimum base for G.

Now we show that the group G has a nonredundant base of size at least $\frac{1}{3}k\log n$. Let $B = b_1, b_2, \ldots, b_{rk}$ where $b_j \in G(Y_j)$, and let $G = G^0 \geq G^1 \geq \cdots \geq G^{rk}$ be the chain of stabilizers of G relative to B. By Lemma 1.5 we have $G^i = G_C(X \setminus \{1, 2, \ldots, i\})$, and it follows that B is a nonredundant base for G. The size of B is $rk \geq \frac{1}{3}k\log n$, since $k2^{r+2} \geq n \geq 8k^2$. □

If B is any nonredundant base $G \leq Sym(n)$, then $\mathcal{M}(G) \leq |B| \leq \mathcal{M}(G)\log n$. We know that there exist groups that have nonredundant bases as large as $\mathcal{M}(G)\log n$. In the next section we show that if B is a greedy base for G, then $|B|$ is closer to $\mathcal{M}(G)$ than to $\mathcal{M}(G)\log n$.

A Sharp Bound for Greedy Bases

In contrast to the $\log n$ indeterminacy of an arbitrary nonredundant base we show that a greedy base is within a $\log\log n$ factor of optimal.

Lemma 3.3 If $G \leq\mathrm{Sym}(n)$ has a base of size k, then there exists a G-orbit of size at least $|G|^{\frac{1}{k}}$.

Proof: Follows from Fact 1.1 and Fact 1.3. □

Theorem 3.4 If $G \leq\mathrm{Sym}(n)$, then any greedy base for G has size no more than $\lceil \mathcal{M}(G)\log\log n \rceil + \mathcal{M}(G)$.

Proof: Define $s_0 = r$ and $s_i = s_{i-1} - \frac{s_{i-1}}{k} - 1$ for $i \geq 1$. By a straightforward inductive argument, we have $s_i \leq r_i$ and $s_i = (1 - \frac{1}{k})^i r - k(1 - (1 - \frac{1}{k})^i)$ for $i \geq 0$. Then

$$s_i \geq 1 \Leftrightarrow i \leq \frac{\log(r + k) - \log(k + 1)}{\log \frac{k}{k-1}}.$$

The result follows since $\frac{k}{2} \leq \frac{1}{\log \frac{k}{k-1}}$. □

<u>Theorem</u> 3.6 Fix $k \geq 2$, then for any $n \geq 2^{4k^2+7k+7}$ there exists $G \leq$Sym(n) such that

- $\mathcal{M}(G) = k$, and
- Every greedy base for G has size at least $\frac{1}{5}\mathcal{M}(G) \log \log n$.

Proof: Suppose that $n \geq 2^{4k^2+7k+7}$, and that r is the largest integer such that $k2^r + 2^{r+k+1} \leq n$. Let X be a set of order rk. The set X is partitioned into k sets of order r, $A_{1,0}, A_{2,0}, \ldots, A_{k,0}$. We now recursively define sets $A_{i,j}$ for $1 \leq i \leq k$, and $1 \leq j \leq \gamma$ (γ defined later) as follows: $A_{i,j}$ is a subset of $A_{i,j-1}$ created by removing $\lfloor \frac{|A_{i,j-1}|}{k} \rfloor + 1$ elements from $A_{i,j-1}$. The elements removed from the k sets $A_{1,j-1}, A_{2,j-1}, \ldots, A_{k,j-1}$ are placed in a set B_j. Note that $|B_j| > |A_{i,j-1}|$. Let $\gamma = \lfloor (k/2)(\log(r + k) - \log(k + 1)) \rfloor$. The value of γ was computed in Lemma 3.5 to insure that $|A_{i,j}| \geq 1$ for all values of i and j. Let $C = \{A_{1,0}, \ldots, A_{k,0}, B_1, \ldots, B_\gamma\}$.

Once again we use the notation from Chapter I to define $G = G_C(X) \leq \text{Sym}(\Omega_C)$. Recall that $G \simeq (Z_2)^{rk}$, and $|\Omega_C| = k2^r + 2^{|B_1|} + 2^{|B_2|} + \ldots + 2^{|B_\gamma|}$.

Consider the sequence of points $A = a_1, a_2, \ldots, a_k$ where $a_i \in G(A_{i,0})$. Since $X = \dot{\bigcup}_{i=1}^{k} A_{i,0}$, it follows from Lemma 1.5 that A is a base for G. Moreover, any subcollection C' of C that covers X (i.e., $X = \bigcup_{Y \in C'} Y$) must contain all the $A_{i,0}$, $1 \leq i \leq k$. It follows that A is a minimum base for G.

Next we show that the Greedy Algorithm must select $B = b_1, b_2, \ldots, b_\gamma$ as a partial base for G, where $b_j \in G(B_j)$. It suffices to show that $G(B_j)$ is the largest G^{j-1}-orbit. By Lemma 1.5 we have $G^{j-1} = G_C(X \backslash \dot{\bigcup}_{l=1}^{j-1} B_l) = G_C(\dot{\bigcup}_{i=1}^{k} A_{i,j-1})$.

Now using Lemma 1.6 we see that the points in $G(A_{i,0})$ are partitioned into G^{j-1}-orbits of size $2^{|A_{i,j-1}|}$ for $1 \leq i \leq k$. The action of G^{j-1} on points $G(B_i)$ is trivial if

Example 3.4 Let $X = \{x_1, x_2, x_3, y_1, y_2\}$, $Y_1 = \{x_1, x_2, x_3\}$, $Y_2 = \{y_1, y_2\}$, and $Y_3 = Y_4 = Y_5 = \{x_1, y_1\}$. Define $C = \{Y_1, Y_2, Y_3, Y_4, Y_5\}$, and let $G = G_C(X)$. Recall that $G \simeq (Z_2)^5$, and that $\Omega_C = \dot{\bigcup}_{i=1}^{5} G(Y_i)$.

Since each $G(Y_i)$ is a G-orbit it follows from Fact 1.1 that any base for G must have size at least 2. Using Lemma 1.5 and Lemma 1.6 we see that $B = b_1, b_2$, where $b_i \in G(Y_i)$ for $i = 1, 2$ is a Greedy1 base for G.

On the other hand the Greedy2 algorithm will always start by fixing a point in either $G(Y_3)$, $G(Y_4)$ or $G(Y_5)$, resulting in a base of size 3.

Example 3.5 Let $X = \{x_1, x_2, x_3, y_1, y_2, y_3\}$, $Y_1 = \{x_1, x_2, x_3\}$, and let $Y_2 = \cdots = Y_7 = \{x_1, y_1\}$. Define $Y_8 = Y_9 = Y_{10} = \{x_2, y_2\}$, and $Y_{11} = \{x_3, y_3\}$. Let $C = \{Y_1, Y_2, \ldots Y_{11}\}$ and let $G = G_C(X)$.

Then $\Omega_C = \dot{\bigcup}_{i=1}^{k} G(Y_i)$, and each $G(Y_i)$ is a G-orbit. Using Lemma 1.5 one checks that $B = b_1, b_2, b_3$ is a minimum base for G, where $b_1 \in G(Y_2)$, $b_2 \in G(Y_8)$ and $b_3 \in G(Y_{11})$. Moreover, the Greedy2 algorithm will always construct a base of size 3.

In contrast, the Greedy1 algorithm will start by fixing a point in the G-orbit $G(Y_1)$, and thus construct a base of size 4.

We know that specific examples cannot be used to compare the two greedy heuristics. Instead, we shall use the worst case performance as a means of comparison. We already have a sharp bound for the worst case performance of the Greedy1 algorithm. What we need now is a sharp bound for the worse case performance of the Greedy2 algorithm. Unfortunately, we are unable to find such a bound. In lieu of a sharp bound, we show that for n sufficiently large there exists $G \leq Sym(n)$ such that, any Greedy2 base for G is arbitrarily close to the upper bound $O(\mathcal{M}(G) \log n)$.

Theorem 3.8 Fix $k \geq 2$ and $0 < \epsilon < 1$. Let N be the smallest integer for which $(\log N)^{\epsilon} \geq \log\log N$. For any integer $n \geq max(N, 2^{2^{k+1}})$ there exists $G \leq Sym(n)$ such that

- $\mathcal{M}(G) = k$, and
- Every Greedy2 base for G has size at least $\frac{1}{6}\mathcal{M}(G)(\log n)^{1-\epsilon}$.

least $\lfloor \frac{r}{c} \rfloor$.

To finish the proof we must prove that $\lfloor \frac{r}{c} \rfloor \geq \frac{1}{6}\mathcal{M}(G) \log n^{1-\epsilon}$. Note that $r \leq \log n < 2r$ implies that $\frac{1}{2}(\log n)^{\epsilon}(\log n)^{1-\epsilon} < r$ and that $\log r \leq \log\log n$. Now the result follows from the fact that $N \leq n$ and $k \leq \log r$. □

Theorem 3.4 and Theorem 3.8 allow us to compare the worst case performance of algorithms Greedy1 and Greedy2.

where $j, k < i$.

Question: Does $g_n = 1$?

In this chapter we prove that the two algebraic problems, deterministic greedy base and factoring (both described later), are P-complete. This is done by reducing a restricted version of the P-complete problem, greedy independent set (GIS), to our algebraic problems. We sketch Cook's proof that GIS is P-complete, and point out why the restricted version of the problem remains P-complete. We conclude the chapter with a PRAM algorithm that proves that factoring is in NC for solvable groups.

Greedy Bases and Independent Sets

Let $\Gamma(V, E)$ be a graph with vertex set V and edge set E. A subset $W \subseteq V$ is called an independent set of vertices in $\Gamma(V, E)$, if for all $w_1, w_2 \in W$, $(w_1, w_2) \notin E$.

There is a natural greedy algorithm for constructing a maximal independent set of vertices in $\Gamma(V, E)$. Given a linear ordering of the vertex set V, the greedy algorithm repeatedly picks the smallest vertex from V that is not adjacent to a previously selected vertex. The corresponding decision problem, greedy independent set, is defined as follows:

GIS Input: Graph $\Gamma(V, E)$ where V is linearly ordered.

Question: Is the last vertex in the ordering part of the greedy maximal independent set?

Proposition 4.1 [Cook] The GIS problem is P-complete.

Proof: The GIS problem is clearly in P. To prove the problem is complete we sketch Cook's logspace reduction of MCVP to GIS.

Let $g_0, g_1, \ldots g_n$ be an instance of the MCVP. We construct a graph $\Gamma(V, E)$ with vertex set $V = \{v_0, v_1, \ldots, v_n\} \cup \{w_0, w_1, \ldots, w_n\}$. We order the vertices so that v_i and w_i precede v_j and w_j, whenever $i < j$. The ordering of v_i relative to w_i is determined by the gate g_i. Let w_0 precede v_0 and let v_1 precede w_1. For any i, $2 \leq i \leq n$, w_i precedes v_i if eligible point.

Let us call this the deterministic greedy (base) algorithm. With respect to this algorithm we can talk about "the" greedy base for a group. Note that with respect to the original Greedy Algorithm there could be an exponential number of greedy bases for a group. We define the deterministic greedy base (DGB) problem as follows:

DGB	Input:	A generating set for $G \leq \mathrm{Sym}(\Omega)$, a linear ordering of Ω and a fixed $\omega \in \Omega$.
	Question:	Is ω part of the greedy base for G?

Lemma 4.3 The DGB problem is P-complete.

Proof: Since the deterministic greedy base is unique we know that the DGB problem is in P. To prove that the problem is P-complete it will suffice to show that there is a logspace reduction of GIS to DGB.

Let $\Gamma(V, E)$ be an instance of GIS with linear ordering $v_0 < v_1 < \cdots < v_n$. By Remark 4.2 we may assume, without loss of generality, that $(v_0, v_1) \in E$ and that each v_i, $2 \leq i \leq n$, is connected to exactly two smaller vertices.

Let $X = \{v_0, v_1, \ldots, v_n\} \cup \{w_1, w_2, \ldots w_n\}$ and let $W_i = \{w_i, v_i\}$ for $1 \leq i \leq n$. Define $Y_0 = Y_1 = \{v_0, w_1, v_1\}$ and for i, $2 \leq i \leq n$, let $Y_i = \{v_i, v_j, v_k\}$, where v_j, v_k are the two unique vertices less than v_i that are connected to v_i.

Let $C = \{W_i, Y_j | 1 \leq i \leq n,\ 0 \leq j \leq n\}$ and let $G = G_C(X) \leq Sym(\Omega_C)$. Recall that Ω_C is the disjoint union of the sets $G(W_i)$ and $G(Y_j)$, $1 \leq i \leq n, 0 \leq j \leq n$, and that G is generated by the permutations $\{\mathcal{R}_C(\sigma_x) | x \in X\}$.

Order the elements in each set $G(W_i)$ and $G(Y_j)$, $1 \leq i \leq n$, $0 \leq j \leq n$ arbitrarily. We will extend this to a linear ordering on Ω_C by ordering the sets so that,

$$G(Y_0) < G(W_1) < G(W_2) < \cdots < G(W_n) < G(Y_1) < G(Y_2) < \cdots < G(Y_n).$$

Let b_i be the smallest point in $G(Y_i)$. We define an instance of the DGB problem where $G = \langle \mathcal{R}_C(\sigma_x) | x \in X \rangle$, and the ordering of the set is defined as above. We wish to

Let $\Gamma(V,E)$ be an instance of GIS, with linear ordering $v_1 < v_2 < \cdots < v_n$. We may assume, without loss of generality, that $(v_1, v_2) \in E$ and that each v_i, $2 \leq i \leq n$, is connected to exactly two vertices less than itself.

Let $G < Sym(3n)$ generated by the 3-cycles $\{(3i-2, 3i-1, 3i) | 1 \leq i \leq n\}$. We view each element $g \in G$ an an n-tuple $g = (g_1, g_2, \ldots, g_n)$, where g_i is equal to either $\bar{0}, \bar{1}$ or $\bar{2}$ (i.e., g_i is equal to either id, $(3i-2, 3i-1, 3i)$ or $(3i-2, 3i, 3i-1)$).

Let $B = 3, 6, \ldots, 3n$ and $G_i = \{(0, \ldots, 0, g_{i+1}, \ldots, g_n) | g_j \in \{\bar{0}, \bar{1}, \bar{2}\}, i+1 \leq j \leq n\}$. Then $G = G^0 > G^1 > \cdots > G^n = \{id\}$ is the chain of stabilizers of G relative to B.

We define a set of coset representatives $U_i = \{\alpha_i, \beta_i, \gamma_i\}$ for G^i in G^{i-1}, $1 \leq i \leq n$, as follows. Set $\alpha_i = id$, $\beta_i = (0, \ldots, 0, g_i, 0, \ldots, 0)$ $g_i = \bar{1}$ and $\gamma_i = (h_1, h_2, \cdots, h_n)$, where

$$h_j = \begin{cases} \bar{2} & \text{if } j = i \\ \bar{1} & i < j \text{ and } (v_i, v_j) \in E \\ \bar{0} & \text{otherwise.} \end{cases}$$

Clearly $U = \bigcup_{i=1}^{n} U_i$ is an SGS for G relative to B, and the set U can be constructed from $\Gamma(V,E)$ by an algorithm that uses no more than $O(\log n)$ space. To complete the construction we define $g = (\bar{2}, \bar{2}, \ldots, \bar{2}) \in G$ and $u = (\bar{0}, \bar{0}, \ldots, \bar{0}, \bar{2}) \in U_n$.

Let V' be the greedy maximal independent set for $\Gamma(V,E)$. It suffices to show that $v_n \in V'$ if and only if $g = u_n u_{n-1} \cdots u_1$, where $u_i \in U_i$ and $u = u_n$.

Fix m, $2 < m \leq n$, and suppose that $g = u_n u_{n-1} \cdots u_1$ and v_j, v_k are the two unique vertices less than v_m that are connected to v_m. Let

$$g u_1^{-1} u_2^{-1} \cdots u_{m-1}^{-1} = (0, \ldots, 0, h_m, h_{m+1} \ldots, h_n),$$

then by the definition of the SGS, U, it follows that

$$h_m = \bar{2} \text{ if and only if } u_j \neq \gamma_j \text{ and } u_k \neq \gamma_k. \tag{IV.4}$$

the commutator $[b_j, b_i]$ satisfies $e_1 = e_2 = \cdots e_i = 0$

(c) for each integer i, $1 \leq i \leq m$, the canonical expression for the element $b_i^{p_1}$ also satisfies $e_1 = e_2 = \cdots = e_i = 0$

Theorem 4.7 Suppose $G \leq Sym(n)$ is solvable and we are given generators for each subgroup in the polynomial tower $G = G_0 \geq G_1 \geq \cdots \geq G_m = \{id\}$, then we can find an NC-efficient SGS for the tower.

Proof: Using machinery established in [35] we can compute, in NC, a PCB for a normal tower $G = N_0 > N_1 > \cdots > N_k = \{id\}$, such that k is $O((\log n)^2)$ and N_{j-1} modulo N_j is abelian semisimple, $1 \leq j \leq k$.

Throughout this proof we view N_{j-1}/N_j as a direct product of vector spaces, and we shall refer to the N_{j-1}/N_j as vector spaces.

Luks and McKenzie introduce the notion of a "structure forest" and develop linear algebra techniques needed to find, in NC, the following:

(a) a homomorphism $\Phi_j : N_{j-1} \rightarrow Sym(A)$, such that $|A|$ is polynomial in n and the $ker\Phi_j = N_j$, $1 \leq j \leq k$

(b) (PCB) elements in N_{j-1} that map to a basis of $\Phi_j(N_{j-1})$

Using the vector space representation, $\Phi_j(N_{j-1}) = N_{j-1}/N_j$, we solve, in NC, the factoring problem modulo N_j. The elements we find in $\Phi_j(N_{j-1})$ are then pulled back to inverse images in G. Next, we describe how the normal tower is used to "slice up" the groups in the tower $G = G_0 \geq G_1 \geq \cdots \geq G_m = \{id\}$.

Using results from [4] generators for $H_j^i = (G_i \cap N_{j-1})N_j$ can be found in NC. Note that this gives us a tower of subspaces for each j, $1 \leq j \leq k$,

$$\Phi_j(N_{j-1}) = \Phi_j(H_j^0) \geq \Phi_j(H_j^1) \geq \cdots \geq \Phi_j(H_j^m) = \{id\}. \qquad \text{(IV.5)}$$

Moreover, we can find in NC sets Γ_j^i, $1 \leq i \leq m$, $1 \leq j \leq k$, such that $\Phi_j(\Gamma_j^i \cup \cdots \cup \Gamma_j^m)$ is a basis for $\Phi_j(H_j^{i-1})$.

Proof: By [4] we can find, in NC, generators for each group in the chain of stabilizers of G with respect to B. □

$\Gamma(G, W')$, includes all the inverses of W (i.e., $W' = W \bigcup W^{-1}$).

To compute a route from the vertex g_1 to the vertex g_2 in $\Gamma(G, W)$ we sift $g_1^{-1}g_2$ through the SGS. This gives us the equation,

$$g_1 u_k u_{k-1} \cdots u_1 = g_2,\ u_i \in U_i.$$

From the equation it follows that there is a path from g_1 to g_2 of the form,

$$g_1 \xrightarrow{u_k} g_1 u_k \xrightarrow{u_{k-1}} g_1 u_k u_{k-1} \xrightarrow{u_{k-2}} \cdots \xrightarrow{u_1} g_2. \tag{V.6}$$

Note that some of the u_i, $1 \le i \le k$, could be the identity. In this case we ignore the edges $\xrightarrow{u_i}$ (i.e., these edges do not exist in the graph). So the actual length of the path is $k - |I|$, where $u_i = id$ if and only if $i \in I$.

If we sift $g_2^{-1}g_1$ we obtain the equation,

$$g_1 v_1^{-1} v_2^{-1} \cdots v_k^{-1} = g_2,\ v_i \in U_i.$$

For the undirected Cayley graph this gives us a second path from g_1 to g_2,

$$g_1 \xrightarrow{v_1^{-1}} g_1 v_1^{-1} \xrightarrow{v_2^{-1}} g_1 v_1^{-1} v_2^{-1} \xrightarrow{v_3^{-1}} \cdots \xrightarrow{v_k^{-1}} g_2. \tag{V.7}$$

Note that any sequence of edges labels $w_1 w_2 \cdots w_m$ may be interpreted as a path in $\Gamma(G, W)$, starting at vertex $g \in G$ and ending at vertex $g w_1 w_2 \cdots w_m$. Throughout the remainder of this chapter paths are described as sequences of edge labels.

Theorem 5.2 Let $\Gamma(G, W)$ be the undirected Cayley graph constructed from an SGS for the tower $G = G_0 \le G_1 \le \cdots \le G_k = \{id\}$. Then there is an efficient algorithm for computing two disjoint paths of length no more than k between any two vertices in the graph.

If $u \in U_i$, then $g_2^{-1}ue_1e_2\cdots e_{i-1} \in G_{i-1}$. We use this fact to define the one-to-one function $\psi_i : U_i \rightarrow U_i$. For $u_i \in U_i$ define $\psi_i(u)$ to be the unique element in U_i satisfying the equation,

$$ue_1e_2\cdots e_{i-1}G_i\psi_i(u) = g_2G_i.$$

Define $\gamma_i : U_i \rightarrow G_i$ so that $\gamma_i(u)$ is the unique element in G_i satisfying the equation,

$$u\gamma_i(u)e_1e_2\cdots e_{i-1}\psi_i(u) = g_2.$$

Consider the following $|W|$ equations, all of which are equal to g_2,

$$u\gamma_i(u)e_1e_2\cdots e_{i-1}\psi_i(u),\ u \in U_i \setminus \{id\}, \text{ for } 1 \leq i \leq k. \tag{V.8}$$

If we replace $\gamma_i(u)$ with the sift of $\gamma_i(u)$, then we can view the sequence of edges labels in (V.8) as a path from g_1 to g_2 ($g_1 = id$). To complete the proof it will suffice to prove that each path has length at most $k+1$, and that all of the paths are disjoint.

First, let us observe that all of the paths have length no more than $k+1$. Since $\gamma_i(u) \in G_i$ a sift of $\gamma_i(u)$ produces a word in W of length at most $k-i$. Thus, all of these paths described above have length no more than $k+1$.

Let P_1 be a prefix of the path $u\gamma_i(u)e_1e_2\cdots e_{i-1}\psi_i(u)$, and let P_2 be a prefix of the path $\gamma_i(w)e_1e_2\cdots e_{i-1}\psi_i(w)$, where $u, w \in W$. As in the proof of Theorem 5.2 it will suffice to show that $P_1 = P_2$ implies that $P_1 = g_2$.

We start by proving that the two paths $u\gamma_i(u)e_1e_2\cdots e_{i-1}\psi_i(u)$, and $\gamma_i(w)e_1e_2\cdots e_{i-1}\psi_i(w)$, are disjoint if $u, w \in U_i \setminus \{id\}$ and $u \neq w$. First note that it is impossible for a prefix of $u\gamma_i(u)$ to equal a prefix of $w\gamma_i(w)$, because these elements are in different G_i cosets. Let s, t be minimal such that,

$$u\gamma_i(u)e_1e_2\cdots e_s = w\gamma_i(u)e_1e_2\cdots e_t. \tag{V.9}$$

congestion as possible. A vertex may transmit more than one message at a time, but only one message may travel on an edge in a given time step. Collisions occur when two or more messages wish to traverse the same edge at the same time. When this happens one of the messages is sent, and the other messages are forced to wait on a queue.

An initialized scheme is a pair (Γ, IC), where Γ is a regular directed graph. The initial conditions, IC, specify how the packets and their destinations are distributed at time zero. We assume that each processor sends h messages and receives h messages.

A routing scheme for (Γ, IC) is oblivious if the route of each message depends only on the source and destination, and is not effected by the routes of the other messages. If e is an edge in Γ then traff(e) is the expected number of distinct messages that pass along e. We say that the routing scheme is symmetric if traff(e_1)=traff(e_2) for all edges e_1, e_2 in Γ.

We say that a scheme is nonrepeating if whenever two messages take paths $e_1e_2\cdots e_r$ and $e'_1e'_2\cdots e'_s$ in which $e_i = e'_j$ and $e_l = e'_m$, then it is the case that $l - i = m - j$ and for all p $(i \leq p \leq l)$ $e_p = e'_{p+j-i}$. In other words, once two routes diverge they remain separated.

Valiant described a simple two-phase routing scheme for the 2-ary m-cube that with high probability runs in $O(m)$ time [44]. A more general proof of the algorithm, together with a permutation routing scheme for shuffle graphs and grid graphs, was given by Valiant and Brebner in [45]. In the first phase of the algorithm messages are sent to random vertices in the graph. In the second phase the messages are routed to their correct destination. Their proof that the algorithm is successful, with overwhelming probability, relies on the fact that each phase of the algorithm is oblivious, nonrepeating, and symmetric.

We show that Valiant's routing scheme is an effective algorithm for solving the partial h-relation problem on any directed Cayley network constructed from an SGS. To accomplish the first phase of the algorithm we have each message in the network select at random $u_i \in U_i$, for $1 \leq i \leq k$. A message at node s is then sent to node $su_ku_{k-1}\cdots u_1$ along the path described by the edge labels u_i. In the second phase of the algorithm we route each message to its final destination using the point-to-point routing algorithm independent Poisson trials with respective probabilities $p_1, p_2, \ldots, p_N$ where $\Sigma_{i=1}^N p_i = Np$, and if $m \geq Np + 1$ is an integer, then the probability of at least m successes is at most $B(m, N, p)$.

Fact 5.6 If $m \geq Np$ is an integer then,

$$\begin{aligned} B(m,N,p) &\leq (\tfrac{Np}{m})^m (\tfrac{N-Np}{N-m})^{N-m} \\ &\leq (\tfrac{Np}{m})^m e^{N-m} \end{aligned}$$

$(e = 2.71\cdots)$.

Fact 5.4 follows from the observation that the routing scheme is nonrepeating. Fact 5.5 is a Theorem of Hoeffding [22]. The first inequality in Fact 5.6 is due to Chernoff [13], and the second follows from the inequality $(1 + \frac{c}{n})^n < e^c$.

Theorem 5.7 Let $\Gamma(G, W)$ be the directed Cayley graph constructed from an SGS for the tower $G = G_0 \leq G_1 \leq \cdots \leq G_k = \{id\}$ where the coset representatives are U_i for $1 \leq i \leq k$. Let $\alpha = (|U_1| + |U_2| + \cdots + |U_k|)^{-1}$, and let (Γ, IC) be any initialized scheme. Then the probability that some message is delayed by ν or more, in one phase of the permutation routing scheme described above, is

$$(\frac{eh\alpha}{\nu})^\nu h|G|,$$

provided that $\nu > h\alpha + 1$.

Proof: We say that a message M intersects some edge e in $\Gamma(G, W)$ in a run of the scheme if its route contains the edge e. Consider some fixed route $R : e_k e_{k-1} \cdots e_1$, where $\ell(e_i) \in U_i$. Let P_M be the probability that message M intersects at least one edge of route R. Let P_{M_i} denote the probability that message M intersects edge e_i. Then,

$$\begin{aligned} \Sigma_M P_M &\leq \Sigma_M P_{M_1} + \Sigma_M P_{M_2} + \cdots + \Sigma_M P_{M_k} \\ &\leq \Sigma_{i=1}^k \text{traff}(e_i) \\ &\leq h\alpha. \end{aligned}$$

(in one phase of the routing scheme) is no more than,

$$(\frac{e\alpha}{\nu})^{\nu}|G| \leq (\frac{\ln n}{\sqrt{n}})^{e(n-1)}.$$

Example 5.7 Let G_h be the automorphism group of a complete binary tree of height h. Label the internal nodes of the tree from top to bottom and from left to right, $v_1, v_2, \ldots, v_{2^h-1}$ (i.e., v_1 is the root of the tree and v_{2^h-1} is the parent of the rightmost leaf). Let $g_i \in G_h$ be the automorphism that flips the subtrees of node v_i, and define $G_i = \langle g_{i+1}, \cdots g_{2^h-1}\rangle$. Let $\Gamma(G_h, W)$ be the directed Cayley graph constructed from an SGS for the subgroup tower, $G_0 > G_1 > \cdots G_{2^h-1} = \{id\}$. Note that $U_i = \{id, g_i\}$ and the graph has $d = k = 2^h - 1$, and $\alpha = k\frac{1}{2}$. If $h = 1$ and $\nu = ek$, then by Theorem 5.7 the probability that a message is delayed by at least ν is no more than,

$$(\frac{e\alpha}{\nu})^{\nu}|G| \leq (\frac{1}{|G|})^{e-1}.$$

If we are only interested in point-to-point routing, then it is possible to create Cayley networks that are more dense than the ones presented above. In fact, in many cases it is possible to reduce the size of the SGS (and hence the degree of the network) by a significant amount and still compute routes (via sift) of a reasonable length. For example, if we replace the SGS in example 5.6 with Jerrum's generators for $Sym(n)$, then we can reduce the degree of the graph by a factor of n and increase the routing diameter by only a factor of 2. The same technique can be used to decrease the degree of the Cayley graph constructed in Example 5.7.

Example 5.8 Consider the tower $Sym(n) = G^0 \geq G^1 \geq \cdots \geq G^{n-1} = \{id\}$, where $G^i = G_{\{1,2,\cdots,i\}}$. Let U_i be a set of coset representatives for G_i in G_{i-1}, and let $W = \{(1,2),(2,3),\ldots,(n,n-1)\}$. For any $u \in U_i$ there exist $w_1, w_2 \in W$ such that $G_i u = G_i w_1 w_2$, $1 \leq i < n-1$. Thus the Cayley graph $\Gamma(Sym(n), W)$ has degree $d = n$ and a routing diameter of $k = 2(n-2)+1$. Note that the routing is done via a sift where elements of U_i are realized as a product of no more than two elements from W.

CHAPTER VI

UNIVERSAL BROADCAST

This chapter is an extension of Faber's work on universal broadcast schemes. The first section contains the definitions and methodology used in [18] to obtain an optimal universal broadcast in the d-cube. In the second section we modify several of the definitions, and prove that it is possible to perform a universal broadcast, in optimal time, in a number of Cayley networks. We also answer the question, asked in [18], as to whether or not there exists an optimal universal broadcast scheme for every QCG. In the last section we descibe an optimal universal broadcast in several Cayley networks whose groups are wreath products.

Background and Methodology

A directed Cayley graph $\Gamma(G, W)$ models a multiprocessor network that in one time step may:

(a) use all edges (lines) in parallel
(b) send at most one message per edge
(c) store, retrieve, and operate on data

We say that a processor $g \in G$ broadcasts a message in $\Gamma(G, W)$, if g sends the message to all the other processors in the network.

Definition 6.1 A task graph T is a sequence of edges $\{e_i | i \in I\}$ of a graph Γ, each with an associated direction, labeled by positive integers (called times) $t(e_i)$, satisfying:

(a) $t(e_i) < t(e_j)$ implies that $e_i < e_j$ (see below) or e_i, e_j incomparable
(b) $t(e_i) = t(e_j)$ implies that $i = j$ or e_i, e_j incomparable

The regular ordering is used to define a task graph, T_{id}, with edges (g_{i_j}, g_{ad+j}) as defined above in (c). The time assigned to edge (g_{i_j}, g_{ad+j}) is $a+1$. This task graph defines a broadcast from processor id to the rest of the network.

The broadcast T_{id} is used as a template to define a broadcast T_g, for every $g \in G$. The graph automorphism $A_g : G \to G$, defined by $A_g(g') = gg'$, is used in conjunction with T_{id} to construct T_g. Let $T_g = A_g(T_{id})$ denote the subgraph of $\Gamma(G, W)$ with vertex set G and edge set $E_{T_g} = \{(gv, gvw) | (v, vw) \in E_{T_{id}}\}$. The function t_g is defined so that $t_g((gv, gvw)) = t_{id}((v, vw))$.

Using condition (c) and the fact that the graph automorphism A_g preserves edge labels (i.e., $t(e) = t(A_g(e))$ and $\ell(e) = \ell(A_g(e))$) Faber proves that $C = \{A_g(T_{id}) | g \in G\}$ is a universal broadcast in $\Gamma(G, W)$.

Lemma 6.6 [Faber] Let T_{id} be a task graph constructed from a regular ordering of $\Gamma(G, W)$ and let $C = \{A_g(T_{id}) | g \in G\}$. Then C describes an optimal universal broadcast in $\Gamma(G, W)$.

Proof: Each T_g, $g \in G$, defines a broadcast from processor g to the rest of the Cayley network, and each broadcast takes time $\lceil \frac{(|G|-1)}{|W|} \rceil$. To finish the proof we must show that no edge in C is labeled with the same time more than once.

Suppose that there exists an edge e in both T_g and T_h, $g, h \in G$, such that $t_g(e) = t_h(e)$. By the construction of T_g and T_h we know that there exists edges $(v, vw), (v', v'w')$ in T_{id} such that,

$$A_g((v, vw)) = e = A_h((v', v'w')).$$

This implies that $w = \ell(e) = w'$, since A_g and A_h preserve the edge label $\ell(e)$. By condition (c) of the regular ordering and the fact that $t_g(e) = t_h(e)$ it follows that $v_g = v_h$. Thus, $g = h$ and there is only one message passed along edge e at time $t_g(e)$. □

Theorem 6.7 [Faber] The time for a universal broadcast in a d-cube is $\lceil \frac{(2^d-1)}{d} \rceil$.

Proof: A proof of this result may be found in [18]. We give an alternate proof in the next section. □

defines a universal broadcast on Γ if and only if $\ell(e_1) \neq \ell(e_2)$ whenever $t_{id}(e_1) = t_{id}(e_2)$ (e_1, e_2 distinct edges in T_{id}).

Proof: The proof of Lemma 6.6 shows that if $\ell(e_1) \neq \ell(e_2)$ for all edges e_1, e_2 in T_{id} with the same time label, then C is a universal broadcast in Γ.

To prove that this condition is necessary, let us assume that there exist distinct edges e_1, e_2 in T_{id} such that $t_{id}(e_1) = t_{id}(e_2)$ and $\ell(e_1) = \ell(e_2) = w$. Then there exist distinct vertices $g, h \in G$, such that $e_1 = (g, gw)$ and $e_2 = (h, hw)$. The broadcast tree $T_{hg^{-1}}$ contains the edge $e_2 = A_{hg^{-1}}(e_1)$ and $t_{hg^{-1}}(e_2) = t_{id}(e_2)$. This will cause a collision on edge e_2 when the messages broadcast from processors id and hg^{-1} both try to cross e_2 at time $t_{id}(e_2)$. □

Let $W = \{w_1, w_2, \ldots, w_d\}$, $G = \langle W \rangle$ and $\langle w_{d+1} \rangle = Z_q$. Given a regular ordering, $\{id, g_1, \ldots, g_{|G|-1}\}$, for $\Gamma(G, W)$ we shall describe a regular ordering for $\Gamma(H, Y)$, where $H = G \times \langle w_{d+1} \rangle$ and $Y = \{(w, id) | w \in W\} \cup \{(id, w_{d+1})\}$. This will be done by organizing the elements of H into blocks, B_i, of size $d+1$. A typical block will have the form, $B_i = [a_1, a_2, \ldots, a_{d+1}]$, where $a_j \in H \setminus \{id\}$ and there exists $b_j \in \bigcup_{l<i} B_l$ with $b_j w_j = a_j$, for $1 \leq j \leq d+1$. The first block is $B_0 = [(w_1, id), (w_2, id), \ldots, (id, w_{d+1})]$.

If we can partition $H \setminus \{id\}$ into $\lceil \frac{|G|q-1}{d+1} \rceil$ blocks (the last block may have less than $d+1$ elements in it), then we have defined a regular ordering for $\Gamma(H, Y)$.

The following notation will be used in our discussion. Let $\rho = \lfloor \frac{|G|-1}{d} \rfloor$ and let $r = (|G| - 1) \bmod d$. Note that $|G| - 1 = \rho d + r$. We denote the element $(g, w_{d+1}^j) \in H$ by g^j, $0 \leq j \leq q-1$.

We describe a procedure, Level(s, m, i, z), that accepts as input the nonnegative integers s, m, i, and z with $0 \leq \rho - s < d$ and $i < q - 1$. The procedure Level will construct blocks for the processors $g^i_{sd+1}, g^i_{sd+2}, \ldots, g^i_{|G|-1}$ and $g^{i+1}_{md+1}, g^{i+1}_{md+2}, \ldots, g^{i+1}_{nd}$ in $\Gamma(H, Y)$. It is assumed that the processors $(G \times \{id\})w^j_{d+1}$, $0 \leq j \leq i-1$, $g^i_1, g^i_2, \ldots, g^i_{sd}$ and $g^{i+1}_0, g^{i+1}_1, \ldots, g^{i+1}_{md}$ have already been organized into blocks $B_0, B_1, \ldots, B_{z-1}$.

```
Procedure Level($s, m, i, z$):
(* assume $0 \leq \rho - s < d$, $0 \leq m$ and $i < q - 1$ *)
```

lowing lines (1), (2.1) and (3.1) are satisfied, then procedure Level organizes processors $g^i_{sd+1}, g^i_{sd+2}, \ldots, g^i_{|G|-1}$ and $g^{i+1}_{md+1}, g^{i+1}_{md+2}, \ldots, g^{i+1}_{nd}$ into blocks.

Proof: We must show that for each new block $B_i = [a_1, \ldots a_{d+1}]$ defined by procedure Level there exists $b_j \in \dot{\bigcup}_{l<i} B_l$ with $b_j w_j = a_j$, $1 \leq j \leq d+1$. The first d elements in each block are organized with respect to the regular ordering given for $\Gamma(G, W)$. Thus, it will suffice to show that there exists $b \in \dot{\bigcup}_{l<i} B_l$ with $b w_{d+1} = a_{d+1}$. One checks that the bounds on m and x given after lines (1), (2.1) and (3.1) insure that this requirement is met. Note that $n \leq \rho$ guarantees that the indices are defined (i.e., not too large). □

Lemma 6.12 Let $W = \{w_1, w_2, \ldots, w_d\}$, $G = \langle W \rangle$ and $\langle w_{d+1} \rangle = Z_2$. If $\{id, g_1, \ldots, g_{|G|-1}\}$ is a regular ordering for $\Gamma(G, W)$ and $\rho - 1 > d \geq 2$, then we can construct a regular ordering for $\Gamma(H, Y)$, where $H = G \times \langle w_{d+1} \rangle$ and $Y = \{(w, id) | w \in W\} \cup \{(id, w_{d+1})\}$.

Proof: Let $m = \lfloor \frac{\rho-1}{d} \rfloor$ and let $s = md + 1$. The first s blocks are $B_j = [g^0_{jd+1}, \ldots, g^0_{jd+d}, g^1_j]$, $0 \leq j \leq md$. At this point we call Level$(s, m, 0, md+1)$.

Recall that the following preconditions must be satisfied for procedure Level to operate correctly:

(a) processors $g^0_1, g^0_2, \ldots, g^0_{sd}$ and $g^1_0, g^1_1, \ldots, g^1_{md}$ are organized into blocks

(b) $0 \leq \rho - s < d$

(c) if $\rho - s = r - 1$ then $s > m$ and $n = m + 1 \leq \rho$

(d) if $0 \leq \rho - s < r - 1$ then $m + r - 1 < \rho$ and $m + r + s \leq 2\rho$

(e) if $r - 1 < \rho - s < d$ then $s > m$, $d + m < \rho$ and $d + r + s + m + 1 \leq 2\rho$

Condition (a) is satisfied by blocks $B_0, \ldots, B_{md}$, and condition (b) follows from the definitions of s and m. To prove that conditions (c), (d) and (e) hold it suffices to show that $\rho > d + m$. Note that $\rho(d-1) > d^2 - 1$, since $\rho > d+1$ (hypothesis). Thus, $\rho > d + \frac{\rho-1}{d}$ and the result follows. When procedure Level is finished we will have organized into blocks the elements $g_1, g_2, \ldots, g_{|G|-1}$ and $g^1_0, g^1_1, \ldots, g^1_{nd}$.

There are $|G| - (nd+1)$ elements left in $H \setminus \{id\}$ that have not been processed into blocks. Let $p = \lfloor \frac{|G|-(nd+1)}{d+1} \rfloor$, let $z = \frac{|G|+nd}{d+1}$ and let $y = (|G| - (nd+1)) \bmod d+1$. We

$\{(id, w_{d+1})\}$.

Proof: The proof is analogous to the proof of Lemma 6.12. One checks that before each call to procedure Level the following preconditions are satisfied:

(a) elements $(G \times \{id\})w_{d+1}^{j}$, $0 \leq j \leq i-1$, $g_1^i, g_2^i, \ldots, g_{sd}^i$ and $g_0^{i+1}, g_1^{i+1}, \ldots, g_{md}^{i+1}$ are organized into blocks

(b) $0 \leq \rho - s < d$

(c) if $\rho - s = r - 1$ then $s > m$ and $n = m + 1 \leq \rho$

(d) if $0 \leq \rho - s < r - 1$ then $m + r - 1 < \rho$ and $m + r + s \leq 2\rho$

(e) if $r - 1 < \rho - s < d$ then $s > m$, $d + m < \rho$ and $d + r + s + m + 1 \leq 2\rho$ □

Lemma 6.14 Let $w_1 = (\alpha, id)$ and $w_2 = (id, \beta)$ be generators for $H = Z_{q'} \times Z_q$. There is a regular ordering for the Cayley network $\Gamma(H, \{w_1, w_2\})$.

Proof: Observe that there is a unique regular ordering for $\Gamma(\langle w_1 \rangle, \{w_1\})$. If we make two slight modifications to procedure Process, then the procedure can be used to define a regular ordering for $\Gamma(H, \{w_1, w_2\})$. First, we leave out line (2.4) (i.e., skip the call to procedure Level). Second, we replace line (2.7) with "n:=m". The procedure call, Process($q' - 1, 1, q$), will result in a regular ordering of $\Gamma(H, \{w_1, w_2\})$. □

Lemma 6.15 Let $L = \langle w_1, \ldots, w_d, w_{d+1} \rangle$ be an abelian group. Let $G = \langle w_1, \ldots, w_d \rangle$ and let $L/G = Z_q$, $q \geq 2$. Let T_1 be a broadcast tree for processor id in $\Gamma(H, Y)$, where $H = G \times \langle \alpha \rangle$, $Y = \{(w, id) | w \in \{w_1, \ldots, w_d\}\} \cup \{(id, \alpha)\}$ and α is a generator for Z_q. Let T_2 be the broadcast tree obtained by replacing, edges of T_1 labeled α with edges labeled w_{d+1} and nodes labeled (g, α^i) with gw_{d+1}^i, $0 \leq i < q$. Then T_2 is a broadcast tree, in $\Gamma(L, \{w_1, \ldots, w_{d+1}\})$, for processor id.

Proof: We need only check that all of the nodes in T_2 are distinct. If $gw_{d+1}^i = g'w_{d+1}^j$, then $i = j$ since q is the smallest integer for which $w_{d+1}^q \in G$. Thus $g = g'$ and we are done. □

construct Cayley networks that cannot be regularly ordered. This answers the question posed by Faber as to whether or not every QCG can be regularly ordered.

Example 6.10 Let $G = Z_{2n}$ and let $W = \{1, n, n+1\}$, then the Cayley network $\Gamma(G,W)$ has diameter $n-1$. Since a broadcast will need at least $n-1$ time steps, there is no regular ordering of the Cayley network when $n \geq 5$.

We conclude this section with a result that shows that an optimal universal broadcast exists for any Cayley network, $\Gamma(G,W)$, where G is a cyclic group and each $w \in W$ is a generator for G.

We say that a path

$$g \xrightarrow{e_1} g_1 \xrightarrow{e_2} g_2 \xrightarrow{e_3} \cdots \xrightarrow{e_s} g_s,$$

in $\Gamma(G,W)$ is a w-path if $\ell(e_i) = w$ for $1 \leq i \leq s$.

Lemma 6.19 Let $\Gamma(G,W)$ be a Cayley network where G is a cyclic group and each $w \in W$ is a generator for G. If $G = X \dot{\cup} Y$, $\{r_1, r_2, \ldots, r_m\} = R \subseteq W$ and $m \leq |Y|$, then there exists $x_1, \ldots, x_m \in X$ and distinct $y_1, \ldots, y_m \in Y$, such that $x_i r_i = y_i$ for $1 \leq i \leq m$.

Proof: We give a constructive proof that uses induction on $m = |R|$ to find the x_i and y_i for $1 \leq i \leq m$. Since each $w \in W$ is a generator, the statement holds for $|R| = 1$. Suppose the lemma is true for $|R| \leq m$ and we have found $x_1, \ldots, x_m \in X$ and distinct $y_1, \ldots, y_m \in Y$, such that $x_i r_i = y_i$ for $1 \leq i \leq m$. Let $\{y_1, \ldots, y_m\} = Y'$ and let

$$x \xrightarrow{e_1} y_{i_1} \xrightarrow{e_2} y_{i_2} \xrightarrow{e_3} \cdots \xrightarrow{e_{s-1}} y_{i_{s-1}} \xrightarrow{e_s} y$$

be a r_0-path from X to $Y \setminus Y'$, where $r_0 \in W \setminus \{r_1, \ldots, r_m\}$. If this path has length 1, then we are done. If the path has length $s > 1$, then we replace the path with an r_0-path from X to $Y \setminus Y'$ of length at most $s-1$.

The procedure we describe for computing the new r_0-path has the property that after the k^{th} step either an r_0-path of length at most $s-1$ is found or the following conditions are true:

$$\begin{array}{lcl} x_j r_j = y_j & \text{with} & x r_j = y, \\ x_{j-2} r_{j-2} = y_{j-2} & \text{with} & x_{j-1} r_{j-2} = y_j, \\ \vdots & & \vdots \\ x_2 r_2 = y_2 & \text{with} & x_3 r_2 = y_4. \end{array}$$

Hence y_2 is unused and we have the r_0-path $x_1 r_0 = y_2$ from X to $Y \setminus Y'$.

Otherwise $x_j r_{j-1} \in Y' \setminus \{y_1, \ldots y_j\}$, and we may order the elements, with indices greater than j, so that $x_j r_{j-1} = y_{j+1}$.

To finish the proof observe that for $j = m$ condition (2) guarantees that $x_m r_{m-1} \notin Y'$. □

Corollary 6.20 Let G be a finite cyclic group and let $W \subseteq G$ such that $G = \langle w \rangle$, for all $w \in W$. Then we can find a regular ordering for $\Gamma(G, W)$. Moreover, any separating set for the graph must have size at least $|W|$.

Universal Broadcast Schemes and Wreath Products

Let $G \leq Sym(A)$ generated by $W = \{w_1, \ldots, w_d\}$, and let $A \dot{\cup} A_1$ denote the disjoint union of two copies of A. Let α be the permutation that interchanges each $a \in A$ with it counterpart in A_1. We extend each $w \in W$ to a permutation on $A \dot{\cup} A_1$, such that w acts trivially on A_1. Then the wreath of G by Z_2, $G \wr Z_2$, is the subgroup of $Sym(A \dot{\cup} A_1)$ generated by $W \cup \{\alpha\}$.

Given a regular ordering for $\Gamma(G, W)$, we have shown how to construct a regular ordering for $\Gamma(H, Y)$, where $H = G \times \langle w' \rangle$ and $Y = \{(w, id) | w \in W\} \cup \{(id, w')\})$. In this section we describe a process for constructing a regular ordering for $\Gamma(G \wr Z_2, W \cup \{\alpha\})$. The following facts about $G \wr Z_2$ are needed:

(a) $|G \wr Z_2| = |G|^2 2$

(b) $G \times G \lhd G \wr Z_2$

(c) $G \wr Z_2 = G \times G \dot{\cup} \alpha(G \times G)$

Thus, each element in $G \wr Z_2$ can be written uniquely as a product $\alpha^e(g_1, g_2)$, where

```
          (1.1) For $i = 0$ to $\rho - 1$ do
                    $(D_i, id)$
Step 2    If $r' = 1$ then
      (*complete the coset $G \times \{id\}$*)
          (2.1) $(R, id)$
          (2.2) $\alpha(R', g_1)$
Step 3
      (*process the blocks of $\alpha(G \times \{id\})$*)
          (3.1) For $i = 0$ to $\rho - 1$ do
                    $\alpha(D_i, id)$
Step 4    If $r' = 1$ then
      (*complete the coset $\alpha(G \times \{id\})$*)
          (4.1) $\alpha(R, id)$
          (4.2) $\alpha(R', g_2)$
Step 5
      (*allow generator $\alpha$ to finish $\{id\} \times G$ and $\alpha(\{id\} \times G)$*)
          (5.1) For $j := 1$ to $\rho d$ do
                    $(D_0, g_j)$
          (5.2) For $j := 1$ to $|G| + r - 2\rho - 2r'$
                    $(D_1, g_j)$
Step 6
      (*process cosets $(id, g_{\rho d+1})(G \times \{id\}), \ldots, (id, g_{\rho d+r})(G \times \{id\})$*)
          (6.1) For $j := 1$ to $r$ do
                    (6.1.1) For $i = 0$ to $\rho - 1$ do
                                $(D_i, g_{\rho d+j})$
                    (6.1.2) $(R, g_{\rho d+j})$
                    (6.1.3) $\alpha(R', g_{j+2r'})$
Step 7
```

```
          (*coincides with step 3 [$i := 1$ to $\rho - 1$] and step 4*)
                (B.1) For $j = 1$ to $\rho - 1 + r'$ do
                            $(id, g_j)$
Step C
          (*process the remaining elements in $\{id\} \times G$ and $\alpha(\{id\} \times G)$*)
          (*coincides with step 5*)
                (C.1) For $j = \rho + r'$ to $|G| - 1$ do
                            $(id, g_j)$
                (C.2) For $j = \rho + r' + 1$ to $|G| - 1$ do
                            $\alpha(id, g_j)$
Step D      If $r' = 1$ then
          (*complete unfinished part of $R'$ blocks started in steps 2, 4 and 6*)
          (*coincides with step 6*)
                (D.1) For $j = 1$ to $r + 2$ do
                            (D.1.1) For $i := 1$ to $r$ do
                                            $\alpha(g_i, g_j)$
Step E      If $r' = 1$ then
          (*process the last $r$ elements in cosets $\alpha(id, g)(G \times \{id\})$*)
          (*step 6 has started by this time*)
                (E.1) For $j = 1$ to $\rho d + r$ do
                            (E.1.1) For $i := \rho d + 1$ to $\rho d + r$ do
                                            $\alpha(g_i, g_j)$
Step F      If $r' = 1$
          (*process the last $r$ elements in cosets $(id, g)(G \times \{id\})$*)
          (*step 7 has started by this time *)
                (F.1) For $j = 1$ to $\rho d$ do
                            (F.1.1) For $i := \rho d + 1$ to $\rho d + r$ do
                                            $(g_i, g_j)$
```

All the elements processed in step A meet this criterion. The elements processed by α in Step B are part of a regular ordering, since step B is not started until the first move of step 3 is finished. Step C presents no problem, since it is not started until after the cosets $G \times \{id\}$ and $\alpha(G \times \{id\})$ are processed.

Condition (b) gaurantees step (5.1) has finished, before step D is started. Step E is not a problem, since step 6 starts processing elements by the time it begins. Likewise step 7 has started processing elements by the time step F is started.

Condition (a) gaurantees that step F runs to completion. In step G we have forced α to process the unordered elements of $\alpha(G \times G)$ in the opposite order that the generators $w_1, \ldots, w_d$ are working. Thus, at some point in time the two procedures will converge on a set of elements of $G \wr Z_2$ of size at most d. These can then be handled by a subset of the generators $w_1, \ldots, w_d$. □

Corollary 6.22 We can construct a regular ordering for $\Gamma(Z_q \wr Z_2, \{w_1, \alpha\})$, where w_1 is a generator for Z_q and $q \geq 2$.

Proof: Note that $\rho = q - 1$, $r = 0$ and $d = 1$. If $q \geq 3$, then $\rho \geq 2$ and generator w_1 can follow the steps outlined in procedure Access1. Every element processed by w_1, after the first step, is multiplied by the generator α. The case $q = 2$ is a simple exercise. □

Corollary 6.23 We can construct a regular ordering for $\Gamma(\wr^k Z_2, \{\alpha_1, \ldots, \alpha_k\})$, where the α_i, $1 \leq i \leq k$, are the canonical generators defined above.

Proof: We use Corollary 6.23 to define a regular ordering for $\Gamma(Z_2 \wr Z_2, \{\alpha_1, \alpha_2\})$. Then given a regular ordering for $\Gamma(\wr^i Z_2, \{\alpha_1, \ldots, \alpha_i\})$, $i \geq 2$, Theorem 6.21 is used to define a regular ordering for $\Gamma(\wr^{i+1} Z_2, \{\alpha_1, \ldots, \alpha_{i+1}\})$. □

The Moebius Graph and Algebraic Tools

Notation:

- Let $(Z_2)^n = \{(x_{n-1},\ldots,x_1,x_0)|x_i \in \{0,1\}\}$
- Let $\vec{0} = (0,0,\ldots,0)$ and let $\vec{1} = (1,1,\ldots,1)$
- For $v \in (Z_2)^n$, let $wt(v) = \Sigma_{i=0}^{n-1} v_i$
- For $v \in (Z_2)^n$, let $pr(v) = wt(v) \bmod 2$
- Let $J \subseteq \{0,2,\ldots,n-1\}$, then we call $v \in (Z_2)^n$ the characteristic vector for J if $v_j = 1$ if and only if $j \in J$

Definition 7.1 For any integer $n \geq 2$ let $M_n(V,E)$ denote the Moebius graph of order n. The graph has $V = (Z_2)^n$, and $E = \{(v, v^s)|v \in V$ and $s \in \{\rho, \rho^{-1}, \delta\}\}$, where ρ and δ are defined by the equations,

$$(x_{n-1},\ldots,x_1,x_0)^\rho = (x_{n-2},\ldots,x_0,\bar{x}_{n-1}), \text{ and}$$
$$(x_{n-1},\ldots,x_1,x_0)^\delta = (x_{n-1},\ldots,x_2,\bar{x}_1,\bar{x}_0).$$

A "path" $v_0, v_1, \ldots, v_m$ starting at vertex v_0 and ending at vertex v_m is denoted by a sequence of edge labels $P = p_1p_2\cdots p_m$, $p_i \in \{\rho, \rho^{-1}, \delta\}$, such that, $v_i = v_{i-1}^{p_i}$. Using this notation, Leland and Solomon described a routing algorithm that constructed paths of the form:

(a) $g_0\rho g_1\rho\cdots g_{n-1}$

(b) $\rho g_0\rho g_1\cdots\rho g_{n-1}$

where $g_i \in \{id, \delta\}$.

They proved that any two vertices in $M_n(V,E)$ could be connected by a path of type either (a) or (b) in which $g_i = id$ for at least $\lfloor\frac{n}{2}\rfloor$ values of i. Thus, the diameter of $M_n(V,E)$ was bounded above by $\lfloor\frac{3n}{2}\rfloor$.

Their algorithm had two shortcomings. First, it could not find a path of length less than $n-1$. As a consequence, a message sent to an adjacent vertex would have to pass of G.

Since $\rho, \delta \in Sym(V)$ it follows from the proof of Lemma 7.3 that $M_n(V,E)$ is isomorphic to $\Gamma(G^n, H^n, W)$, where $W = \{\rho, \delta\}$, and $H^n = G^n_{\underline{0}}$. We denote the isomorphism between $M_n(V,E)$ and $\Gamma(G^n, H^n, W)$ by the function $\Phi(v) = H^n g$, where $\vec{0}^g = v$.

The first thing we must do is decide on a reasonable representation for G^n. If we were to represent a permutation $g \in G^n$ as a product of disjoint cyclics, we would need $O(n2^n)$ space to store g. Instead, we will view G^n as a subgroup of the Affine group A of $(Z_2)^n$, where

$$A = \{(N,v) | N \text{ is an } n \times n \text{ invertible matrix, and } v \in (Z_2)^n\}.$$

For $(N,v) \in A$ and $x \in (Z_2)^n$ the action of (N,v) on x is defined by the equation, $x^{(N,v)} = xN + v$. The product of two elements $(N,v),(L,w) \in A$ is $(N,v)(L,w) = (NL, vL + w)$. This representation allows us to store each $g \in G^n$ in $O(n)$ space.

If we let I be the $n \times n$ identity matrix and let

$$M = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix},$$

then the permutations ρ, ρ^{-1}, and δ can be viewed as elements of A, where

$$\rho = (M, (0, \cdots, 0, 1)),$$
$$\rho^{-1} = (M^{-1}, (1, 0, \cdots, 0)), \text{ and}$$
$$\delta = (I, (0, \cdots, 0, 1, 1)).$$

Definition 7.5 For $0 \le i \le n-1$ let $\delta_i = \rho^{-i}\delta\rho^i$. Then $\delta_i = (I, w)$ where $w_j = 0$ unless $j = i+1 \bmod n$ or $j = i \bmod n$. Note that $\delta_0 = \delta$, and

$$\rho^i \delta_{j+i} = \delta_j \rho^i. \qquad \text{(VII.12)}$$

$$\Phi(v) = \begin{cases} H^n(I,v) & \text{if } n \geq 3 \text{ and odd} \\ H^n(I,v) & n \text{ even and } pr(v) = 0 \\ H^n(M,v) & n \text{ even and } pr(v) = 1 \end{cases}$$

Viewing the Moebius graph as an QCG has already given us some valuable information about the family of Moebius graphs. In particular, it points out the we are really dealing with two separate families, one for odd values of n, and one for even values of n.

An Optimal Routing Algorithm for the Moebius Graph

In this section we reduce the problem of routing on the Moebius graph to the problem of finding a minimum generating sequence (with respect to $\{\rho, \rho^{-1}, \delta\}$) for $g \in G^n$. Let v_1, v_2 be two vertices from the Moebius graph $M_n(V,E)$. Recall that a sequence of edge labels $P = p_1p_2\cdots p_m$, $p_i \in \{\rho, \rho^{-1}, \delta\}$, describes a path from v_1 to v_2 if and only if $v_1^{p_1p_2\cdots p_m} = v_2$. Since $\delta^2 = 1$ we may assume that every path in $M_n(V,E)$ has the form $\rho^{e_{a+1}}\delta\rho^{e_a}\delta\cdots\delta\rho^{e_1}$, where the e_i are integers and nonzero if $2 \leq i \leq a$. The length of the path is $a + \Sigma_{i=1}^{a+1}|e_i|$. Note that a corresponds to the number of δ edges traversed and $\Sigma_{i=1}^{a+1}|e_i|$ to the number of ρ and ρ^{-1} traversed.

Let $OP(v_1,v_2)$ denote an optimal path from v_1 to v_2 in $M_n(V,E)$, and let $MGS(g)$ denote a minimum generating sequence for $g \in G^n$. The algorithm OptimalRoute reduces the problem of computing an optimal route to the problem of finding minimum generating sequence. The input to the algorithm is a pair of vertices from the Moebius graph and the order, n, of the graph. The algorithm returns OP, an optimal path between the vertices.

```
Procedure OptimalPath($v_1, v_2, n, OP$) :
  (1) Compute $g_1$ such that $H^n g_1 = \Phi(v_1)$
  (2) Compute $g_2$ such that $H^n g_2 = \Phi(v_2)$
  (3) $OP := MGS(g_1^{-1} g_2)$
  (4) For $h \in H^n \setminus \{1\}$ do
```

(a) $d_i \in \{1, -1\}$, and $0 \leq t_1 < t_2 < \ldots < t_{wt(v)} \leq m$

(b) $v_s = 1$ if and only if for some $1 \leq t_j \leq wt(v)$, $\Sigma_{i=1}^{t_j} d_i = s \bmod n$

(c) $\Sigma_{i=1}^{m} d_i = d$

(d) if D' and T' are two sequences satisfying properties 1-3, then $|D| \leq |D'|$

Informally, we may think of v as a circular queue of size n, where cell s of the queue is "marked" if and only if $v_s = 1$. A "walk" on the queue consists of a sequence of steps in either the clockwise (positive) direction, or the counterclockwise (negative) direction. The procedure QueueWalk finds a shortest walk, $D = d_1, d_2, \ldots, d_m$, that starts at cell zero, visits every marked cell of the queue, and terminates a distance $|d|$ away from the start position in the direction $\frac{|d|}{d}$. The length of the walk is m, and the t_i indicate the time at which a marked cell is visited. Note that once a walk is described it is a simple exercise to compute the sequence T in time m.

```
Procedure QueueWalk($v, d, D, T$) :
  (1) If $|d| \geq n$, then
      (1.1)  $d_i := \frac{|d|}{d}$ for $1 \leq i \leq |d|$
             (*$m = |d|$*)
      (1.2)  Compute $T$ satisfying condition (b) above
      (1.3)  Return($D$,$T$).
  (2) If $0 \leq d < n$, then
      (2.1)  Find a largest block of zeros, $(s, t)$, between $[d + 1, n - 1]$
             (*the block of zeros has size $E = t - s - 1$*)
      (2.2)  $s_1 := n - t$ $s_2 := s - d$
      (2.3)  $d_i := -1$ for $1 \leq i \leq s_1$
             $d_i := 1$ for $s_1 + 1 \leq i \leq 2s_1 + s$
             $d_i := -1$ for $2s_1 + s + 1 \leq i \leq 2s_1 + s + s_2$
             (*$m = d + 2(n - d - 1 - E)$*)
      (2.4)  Compute $T$ satisfying condition (b) above
      (2.5)  Return($D$,$T$).
```

and this implies that (s,t) is not a largest block of zeros.

Case III ($-n < d < 0$) The proof of this case is analogous to the proof of Case II.□

The following claim constructs a generating sequence for $g \in G^n$ using the procedure QueueWalk. The claim proves that with the appropriate input QueueWalk can be used to find a minimum generating sequence for g.

Claim 7.10 Let $g \in G^n$, and suppose $g = \rho^d k$. Let v be a characteristic vector for k and let $D = d_1, d_2, \ldots, d_m$, $T = t_1, t_2, \ldots, t_a$ ($wt(v) = a$) be the result of a call to QueueWalk(v, d, D, T). Define $p_1 = d_1 + \cdots + d_{t_1}$, $p_{i+1} = d_{t_i+1} + \cdots + d_{t_{i+1}}$ for $1 \leq i < a$, and $p_{a+1} = d_{t_a} + \cdots + d_m$, then $\rho^{p_{a+1}} \delta \rho^{p_a} \delta \cdots \delta \rho^{p_1}$ is a generating sequence for g.

Moreover, if $MGS(g) = \rho^{e_{a+1}} \delta \rho^{e_a} \delta \cdots \delta \rho^{e_1}$, where $d = \Sigma_{i=1}^{a+1} e_i$, and $v_s = 1$ if and only if there exist j such that $\Sigma_{i=1}^{j} e_i = s \bmod n$, then $\rho^{p_{a+1}} \delta \rho^{p_a} \delta \cdots \delta \rho^{p_1}$ is a minimum generating sequence for g.

Proof: First, observe that $\Sigma_{i=1}^{j} p_i \equiv s \bmod n$ if and only if $v_s = 1$. Now using identity (VII.12) we have $\rho^d k = \rho^{p_{a+1}} \delta \rho^{p_a} \delta \cdots \delta \rho^{p_1}$. To finish the proof it will suffice to show that $\Sigma_{i=1}^{a+1} |p_i| = \Sigma_{i=1}^{a+1} |e_i|$. Since $\rho^{e_{a+1}} \delta \rho^{e_a} \delta \cdots \delta \rho^{e_1}$ is a minimum generating sequence for g it follows that $\Sigma_{i=1}^{a+1} |e_i| \leq \Sigma_{i=1}^{a+1} |p_i|$. Just suppose $\Sigma_{i=1}^{a+1} |p_i| > \Sigma_{i=1}^{a+1} |e_i|$; then we could use the the e_i to define a walk of length $\Sigma_{i=1}^{a+1} |e_i| < m$ for v and d. But by Claim 7.9 such a walk cannot exist. □

Corollary 7.11 (a) If $g = \rho^d k$, and v is the characteristic vector of k, then there exists a generating sequence for g of size

$$\begin{cases} wt(v) + |d| & \text{if } |d| \geq n \\ wt(v) + d + 2(n - d - 1 - E) & \text{if } 0 \leq d < n \\ wt(v) + |d| + 2(n - |d| - 1 - E) & -n < d < 0. \end{cases}$$

Recall that $E = t - s - 1$ (E is defined in *QueueWalk*).

(b) If $MGS(g) = \rho^{e_{a+1}} \delta \rho^{e_a} \delta \cdots \delta \rho^{e_1}$, then $-n \leq \Sigma_{i=1}^{a+1} e_i \leq n$. This follows from the fact that $\rho^{2n} = (I, \vec{0})$.

Claim 7.10, then statement (8) takes time $O(n)$.

To finish the proof we must prove that the algorithm computes a minimum generating sequence for g. Suppose $MGS(g) = \rho^{e_{a+1}}\delta\rho^{e_a}\delta\cdots\delta\rho^{e_1}$, and $c_j = \Sigma_{i=1}^{j} e_i \bmod n$, for $1 \le j \le a$. By Corollary 7.11 and Claim 7.7 we have $d = \Sigma_{i=1}^{a+1} e_i$, and it follows that either v or v' is the characteristic vector for $k = \Pi_{j=1}^{a} \delta_{c_j}$. The proof follows from Claim 7.10. □

The procedure for computing a minimum generating sequence for $g \in G^n$ when n is even is analogous to the previous procedure. The only difference is that now there are two possible choices for d (Claim 7.7).

```
Procedure MinGenSeqEven($g = (M^i, v), n, mgs$):
(*We shall assume that $n$ is even, and $0 \le i \le n-1$*)
  (1)  $k_1 := \rho^{-i} g$ $(g = \rho^i k_1)$
  (2)  $k_2 := \rho^{n-i} g$ $(g = \rho^{i-n} k_2)$
  (3)  Compute $v_1$ and $v_1'$, the characteristic vectors for $k_1$
  (4)  Compute $v_2$ and $v_2'$, the characteristic vectors for $k_2$
  (5)  QueueWalk($v_1, i, D_1, T_1$)
  (6)  QueueWalk($v_1', i, D_1', T_1'$)
  (7)  QueueWalk($v_2, i-n, D_2, T_2$)
  (8)  QueueWalk($v_2', i-n, D_2', T_2'$)
  (9)  If $|D_1'| + |T_1'| < |D_1| + |T_1|$
           then set $D_1 := D_1'$ and $T_1 := T_1'$
  (10) If $|D_2'| + |T_2'| < |D_2| + |T_2|$
           then set $D_2 := D_2'$ and $T_2 := T_2'$
  (11) If $|D_1| + |T_1| < |D_2| + |T_2|$
           then use $D_1$ and $T_1$ to compute $MGS(g)$
           else use $D_2$ and $T_2$ to compute $MGS(g)$
  (12) Return ($mgs := MGS(g)$)
```

Theorem 7.13 Procedure MinGenSeqEven computes a minimum generating sequence for $g = (M^i, v) \in G^n$ in $O(n)$ time for even values of n.

Table 3: For n Odd and $pr(v_1) = pr(v_2)$

For n Odd and $pr(v_1) = pr(v_2)$	
Elements in G^n	Dependencies
$g_1^{-1}(I, \vec{0})g_2 = \rho^0 k_0$	
$g_1^{-1}(M^2, \vec{0})g_2 = \rho^2 k_2$	$\delta_0 k^{(0)} k_0 = k_2$
$g_1^{-1}(M^4, \vec{0})g_2 = \rho^4 k_4$	$\delta_2 k^{(2)} k_2 = k_4$
$\vdots$	$\vdots$
$g_1^{-1}(M^{n-1}, \vec{0})g_2 = \rho^{n-1} k_{n-1}$	$\delta_{n-3} k^{(n-3)} k_{n-3} = k_{n-1}$
$g_1^{-1}(M^1, \vec{0})g_2 = \rho^{1-n} k_1$	$\delta_{n-1} k^{(n-1)} k_{n-1} = k_1$
$g_1^{-1}(M^3, \vec{0})g_2 = \rho^{3-n} k_3$	$\delta_1 k^{(1)} k_1 = k_3$
$\vdots$	$\vdots$
$g_1^{-1}(M^{n-4}, \vec{0})g_2 = \rho^{-4} k_{n-4}$	$\delta_{n-6} k^{(n-6)} k_{n-6} = k_{n-4}$
$g_1^{-1}(M^{n-2}, \vec{0})g_2 = \rho^{-2} k_{n-2}$	$\delta_{n-4} k^{(n-4)} k_{n-4} = k_{n-2}$
	$\delta_{n-2} k^{(n-2)} k_{n-2} = k_0$

Table 4: For n Odd and $pr(v_1) \neq pr(v_2)$

For n Odd and $pr(v_1) \neq pr(v_2)$	
Elements in G^n	Dependencies
$g_1^{-1}(I, \vec{1})g_2 = \rho^1 k_1$	
$g_1^{-1}(M^3, \vec{0})g_2 = \rho^3 k_3$	$\delta_1 k^{(1)} k_1 = k_3$
$g_1^{-1}(M^5, \vec{0})g_2 = \rho^5 k_5$	$\delta_3 k^{(3)} k_3 = k_5$
$\vdots$	$\vdots$
$g_1^{-1}(M^{n-2}, \vec{0})g_2 = \rho^{n-2} k_{n-2}$	$\delta_{n-4} k^{(n-4)} k_{n-4} = k_{n-2}$
$g_1^{-1}(I, \vec{0})g_2 = \rho^n k_0$	$\delta_{n-2} k^{(n-2)} k_{n-2} = k_0$
$g_1^{-1}(M^2, \vec{0})g_2 = \rho^{2-n} k_2$	$\delta_0 k^{(0)} k_0 = k_2$
$\vdots$	$\vdots$
$g_1^{-1}(M^{n-3}, \vec{0})g_2 = \rho^{-3} k_{n-3}$	$\delta_{n-5} k^{(n-5)} k_{n-5} = k_{n-3}$
$g_1^{-1}(M^{n-1}, \vec{0})g_2 = \rho^{-1} k_{n-1}$	$\delta_{n-3} k^{(n-3)} k_{n-3} = k_{n-1}$
	$\delta_{n-1} k^{(n-1)} k_{n-1} = k_1$

Lemma 7.15 If n is odd and v_1 and v_2 are two vertices from the $M_n(V,E)$, then there exists a path between the vertices of length no more than $\lceil\frac{3n}{2}\rceil - 2$.

Proof: Let $\Phi(v_1) = H^n g_1$ and $\Phi(v_2) = H^n g_2$. By the definition of Φ we have $g_1 = (I, v_1)$ and $g_2 = (I, v_2)$. It will suffice (by claim 7.8) to prove that there exists $h \in H^n$ such that $g_1^{-1}hg_2$ has a generating sequence of length no more than $\lceil\frac{3n}{2}\rceil - 2$.

Case I ($pr(v_1) = pr(v_2)$) Let $g_1^{-1}(M^{n-1}, \vec{0})g_2 = \rho^{n-1}k_{n-1}$ (notation from Table 3). By Remark 7.6 we know that there exists a characteristic vector, v, for k_{n-1} such that $wt(v) \leq \lceil\frac{n}{2}\rceil - 1$. Now, by Corollary 7.11 we have,

$$|MGS(\rho^{n-1}k_{n-1})| \leq wt(v) + n - 1 + 2(0) \leq \lceil\frac{3n}{2}\rceil - 2.$$

CaseII($pr(v_1) \neq pr(v_2)$) Let $g_1^{-1}(M^{n-2}, \vec{0})g_2 = \rho^{n-2}k_{n-2}$. By Remark 7.6 we know that there exists a characteristic vector, v for k_{n-2}, such that $v_{n-1} = 0$. If the $wt(v) \leq \lceil\frac{n}{2}\rceil$, then

$$|MGS(\rho^{n-2}k_{n-2})| \leq wt(v) + n - 2 + 2(0) \leq \lceil\frac{3n}{2}\rceil - 2.$$

On the other hand if $wt(v) > \lceil\frac{n}{2}\rceil$, then let v' be the complement of v. We know that v' is a characteristic vector of k_{n-2} and $wt(v') \leq \lceil\frac{n}{2}\rceil - 2$. Thus,

$$|MGS(\rho^{n-2}k_{n-2})| \leq wt(v') + n - 2 + 2(1) \leq \lceil\frac{3n}{2}\rceil - 2.$$

□

Theorem 7.16 If n is odd then the diameter of $M_n(V,E)$ is $\lceil\frac{3n}{2}\rceil - 2$.

Proof: Let $v_1 = (0,1,0,1,0,\ldots,0,1,0)$ and let $v_2 = (1,0,1,0,1,\ldots,1,0,1,0,0)$. If $\Phi(v_1) = H^n g_1$ and $\Phi(v_2) = H^n g_2$, then by the definition of Φ, $g_1 = (I, v_1)$ and $g_2 = (I, v_2)$. By Lemma 7.15 and Claim 7.8 it suffices to show that $|MGS(g_1^{-1}hg_2)| \geq \lceil\frac{3n}{2}\rceil - 2$, for all $h \in H^n$.

Since $pr(v_1) = pr(v_2)$ we have for $0 \leq i \leq n-1$, $(I, v_1)(M^i, \vec{0})(I, v_2) = \rho^d k_d$, where $d = i$ if i is even, and $d = i - n$ if i is odd (d must be even). Solving for k_d we find

On the other hand if $wt(v) = \frac{n}{2}$, then consider $g_1^{-1}(I, \vec{0})g_2 = \rho^{1-n}k_{1-n}$. By Table 5 we have,

$$k_{1-n} = \delta_{n-1}k^{(n-1)}k_{n-1}.$$

Let w be the characteristic vector for k_{1-n} and recall that $k = (I, v_1 + v_1M^2)$. Using Fact 7.17 we conclude that the $wt(w) \neq \frac{n}{2}$. Without loss of generality we may assume that $wt(v) < \frac{n}{2}$ and

$$|MGS(\rho^{1-n}k_{1-n})| \leq \frac{n}{2} - 1 + n - 1.$$

□

Lemma 7.19 Let n be even, and let v_1 and v_2 be two vertices in the Moebius graph $M_n(V, E)$. If $pr(v_1) = pr(v_2)$, then there exists a path between the vertices of length no more than $\frac{3n}{2} - 1$.

Proof: Let $\Pi(v_1) = H^n g_1$ and $\Pi(v_2) = H^n g_2$. If $pr(v_1) = pr(v_2) = 0$, then $g_1 = (I, v_1)$ and $g_2 = (I, v_2)$. If $pr(v_1) = pr(v_2) = 1$, then $g_1 = (M, v_1)$ and $g_2 = (M, v_2)$. It will suffice to show that there exists $h \in H^n$, such that $g_1^{-1}hg_2$ has a generating sequence of length no more than $\frac{3n}{2} - 1$. Let $g_1^{-1}(I, \vec{0})g_2 = \rho^{-n}k_{-n}$, and let v be a characteristic vector of k_{-n} such that $wt(v) \leq \frac{n}{2}$. If $wt(v) \neq \frac{n}{2}$ then we have,

$$|MGS(\rho^{-n}k_{-n})| \leq \frac{n}{2} - 1 + n.$$

If the $wt(v) = \frac{n}{2}$, then we consider the element $g_1^{-1}(M^2, \vec{0})g_2 = \rho^{2-n}k_{2-n}$, and let w be a characteristic vector of k_{2-n}. As before, we use Table 6 and Fact 7.17 to prove that $wt(w) \neq \frac{n}{2}$ $(k_{2-n} = \delta_0 k^{(0)}k_{-n})$. So we may assume that $wt(w) \leq \frac{n}{2} - 1$; and it follows that,

$$|MGS(\rho^{2-n}k_{2-n})| \leq \frac{n}{2} - 1 + |2 - n| + 2(1).$$

□

Theorem 7.20 Let n be an even integer greater than 10. Then the diameter of $M_n(V, E)$ is $\frac{3n}{2} - 1$.

Table 8: Characteristic Vectors for $n = 14$

k_j	Characteristic vector for k_j
k_{-14}	01011010110100
k_{-12}	01011110110100
k_{-10}	01001110110100
k_{-8}	00001110110100
k_{-6}	00001110110101
k_{-4}	00001110110001
k_{-2}	00001110100001
k_0	00001111100001
k_2	00001011100001
k_4	00011011100001
k_6	01011011100001
k_8	01011011100000
k_{10}	01011011100100
k_{12}	01011011110100

(a) $wt(v_j) \geq \frac{n}{2} - 1$

(b) no string of digits in v_j has length more than 5

By Corollary 7.11 we conclude that $|MGS(\rho^j k_j)| \geq \frac{3n}{2} - 1$ whenever $|j| \leq n - 12$. Thus, we need only check (by hand) that $|MGS(\rho^j k_j)| \geq \frac{3n}{2} - 1$ for $j = n - 10, n - 8, \ldots, n - 2, n, 2 - n, \ldots, 10 - n$.

Case III ($n = 16 + 4m$) Let $v_1 = 0^{n-8}1^8$, and let $v_2 = 00101100(1010)^m11011100$. Note that $k = \delta_0\delta_8$, $k_{j+2} = k_j\delta_{8+j}$, and the characteristic vector for k_{-n} is $k_{-n} = 10110001(0011)^m01001011$. It is a simple (but tedious) task to check that properties (a) and (b) hold. Now Corollary 7.11 and Table 9 may be used to check that $|MGS(\rho^j k_j)| \geq \frac{3n}{2}$ for the remaining 11 values in question.

Case IV ($n = 18 + 4m$) Let $v_1 = 0^8 1^{n-8}$, and let $v_2 = 001011001100(1010)^m101100$. Note that $k = \delta_{n-8}\delta_8$, $k_{j+2} = k_j\delta_{j-8}$, and the characteristic vector for k_{-n} is $010011101011(1001)^m100100$. Now Corollary 7.11 and Table 10 may be used to to finish the proof. □

CHAPTER VIII

SUMMARY AND FUTURE WORK

In this dissertation we have focused our attention on bases, SGSs and subgroup towers for permutation groups. We investigated both the sequential and parallel complexity of several algebraic problems involving bases and SGSs. We have also shown how subgroup towers and SGSs can be used to design dense interconnection networks that are accompanied by efficient routing algorithms.

In Chapter II we answered in the negative the question asked by Finkelstein as to whether or not the Greedy1 algorithm always computes a minimum base. In fact, we proved that the problem of computing a minimum base for $G \leq Sym(n)$ is NP-hard. Moreover, the problem remains NP-hard even if we restrict G to be an abelian group with orbits of size no more than 8.

For abelian groups with orbits of size 7 or less we described a polynomial time algorithm for computing minimum bases. Thus, for abelian groups this bound on the size of the orbits is sharp. The computational complexity of computing minimum bases for arbitrary groups with orbits of size less than 8 remains open. We have preliminary results that reduce this problem to the cases where the orbits have size 4, 6 and 7.

In Chapter III we examined the problem of approximating minimum bases for permutation groups. We observed that it was possible for $G \leq Sym(n)$ to have a nonredundant base of size $\frac{1}{3}\mathcal{M}(G)\log n$. In contrast, the Greedy1 algorithm always produces a base of size no more than $\lceil \mathcal{M}(G) \log\log n \rceil + \mathcal{M}(G)$. We went on to prove that, up to a constant, this bound on the size of a Greedy1 base is sharp. That is, for any n sufficiently large there exists $G \leq Sym(n)$, such that every Greedy1 base for G has size at least $\frac{1}{5}\mathcal{M}(G)\log\log n$.

We examined a second greedy algorithm, Greedy2, for constructing small bases.

adapted to run on directed SGS Cayley networks. In fact, the algorithm solves the partial permutation routing problem. This is one of three subroutines needed in the simulation of idealistic (PRAM) computers by realistic (multiprocessor network) computers [43, page 227]. The other two subroutines are sorting and distribution. One of the problems we are working on now is an efficient sorting algorithm for SGS Cayley networks. We hope to show that the SGS Cayley networks can use a modified version of odd-even merge sort.

There are two other questions concerning permutation routing that warrant further investigation. First, Pippenger has described a network in which a variant of Valiant's algorithm performs permutation routing and uses bounded queues [38]. Can this algorithm be adapted to SGS Cayley networks? Second, Leighton, Maggs and Rao have described an off-line algorithm that eliminates the probabilistic component of permutation routing [28]. Is there an on-line version of this algorithm that will run on SGS Cayley networks?

In Chapter VI we extended Faber's work on universal broadcast schemes. We proved that it is possible to find an optimal universal broadcast algorithm for a number of Cayley networks. In particular, we showed that if there is an optimal universal broadcast for $\Gamma(G,W)$ and $\lfloor\frac{|G|-1}{|W|}\rfloor - 1 > |W| > 2$, then there is an optimal universal broadcast for $\Gamma(H,Y)$, where $H = G \times \langle w_{d+1}\rangle$ and $Y = \{(w,id)|w \in W\} \cup \{(id,w_{d+1})\}$ $(|\langle w_{d+1}\rangle| \geq 2)$.

As a consequence of this result we proved that if G is an abelian group and $W = \{w_1, w_2, \ldots, w_k\}$ is a generating set for G such that $w_i \notin \langle w_1, w_2, \ldots w_{i-1}\rangle$, $1 < i \leq k$, then the time needed for a universal broadcast in $\Gamma(G,W)$ is $\lceil\frac{|G|-1}{|W|}\rceil$. This yields an alternate proof of Vaber's result that the time for a universal broadcast in a d-cube is $\lceil\frac{(2^d-1)}{d}\rceil$.

We also proved that an optimal universal broadcast can be found for the Cayley network $\Gamma(G,W)$, where $G = \wr^k Z_2$ and W is the canonical set of minimal generators defined in Example 5.9. Recently we described a universal broadcast that runs in time $\lceil\frac{|G|-1}{|W|}\rceil$ on the Cayley network $\Gamma(G,W')$, where $G = \wr^k Z_2$ and W' is the canonical set of generators defined in Example 5.7. We would like to extend these results to other nonabelian Cayley networks.

In Chapter VII we used QCGs to analyze nonsymmetric networks. Our first result was a useful characterization of QCGs. We proved that a connected directed graph is

BIBLIOGRAPHY

[1] Arden, B., and Lee, H. Analysis of Chordal Ring networks. *IEEE Trans. Electron. Comput. C-30* (Apr. 1981), 291–301.

[2] Babai, L. On the order of uniprimitive permutation groups. *Annals of Math. 113* (1981), 553–568.

[3] Babai, L. On the length of subgroup chains in the symmetric group. *Communications in Algebra 14* (1986), 1729–1736.

[4] Babai, L., Luks, E., and Seress, A. Permutation groups in NC. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing* (1987), vol. 19, pp. 409–420.

[5] Bannai, E., and Ito, T. On finite Moore graphs. *Journal of Fac. Sci. Univ. Tokyo 20* (1973), 191–208.

[6] Brown, C., Finkelstein, L., and Purdom, P. Efficient implementation of Jerrum's algorithm for permutation groups. Pre-print.

[7] Brown, C., Finkelstein, L., and Purdom, P. Backtrack searching in the presence of symmetry. Tech. Rep. NU-CCS-87-2, Northeastern University, 1987.

[8] Cameron, P. Personal correspondence to K. D. Blaha.

[9] Cannon, J. A computational toolkit for finite permutation groups. In *Proceedings of Rutgers Group Theory, 1983-1984* (1984), pp. 1–18.

[10] Carlsson, G., Cruthirds, J., Sexton, H., and Wright, C. Interconnection networks based on a generalization of Cube-connected cycles. *IEEE Trans. on Comput. C-34 No. 8* (Aug. 1985), 769–722.

[11] Carlsson, G., Fellows, M., Sexton, H., and Wright, C. Group theory as an organizing principle in parallel processing. Pre-print.

[12] Carlsson, G., Sexton, H., and Wright, C. Cayley networks and generalized Cube-connected cycles. Pre-print.

[13] Chernoff, H. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Math. Stat. 23* (1952), 493–507.

[31] Lovász, L. The matroid matching problem. In *Proceedings of the Conference on Algebraic Methods in Graph Theory* (1978).

[32] Lovász, L. Matroid matching and some applications. *Journal of Combin. Theory Ser. B 28* (1980), 208–236.

[33] Lovász, L., and Plummer, M. *Matching Theory.* North-Holland, Amsterdam, 1986.

[34] Luks, E. Parallel algorithms for permutation groups and graph isomorphism. In *Proceedings 27th Annual Symposium on Foundations of Computer Science* (1986), vol. 27, pp. 292–302.

[35] Luks, E., and McKenzie, P. Fast parallel computation with permutation groups. In *Proceedings 26th Annual Symposium on Foundations of Computer Science* (1985), vol. 26, pp. 505–514.

[36] Hall, M., Jr. *The Theory of Groups.* Macmillan, New York, 1959.

[37] McKenzie, P., and Cook, S. The parallel complexity of abelian permutation group problems. Tech. Rep. No. 181-85, Dept. of Computer Science, University of Toronto, 1985.

[38] Pippenger, N. Parallel communication with limited buffers. Tech. Rep., IBM Research Laboratory, San Jose, Calif., 1984.

[39] Preparata, F., and Vuillemin, J. The Cube-connected cycles: A versatile network for parallel computation. *Commun. Ass. Comput. Mach. 24* (1980), 300–309.

[40] Seitz, C. The CosmicCube. *Commun. of the ACM 28* (1985), 22–33.

[41] Sims, C. Computational methods in the study of permutation groups. In *Computational Problems in Abstract Algebra* (1970), J. Leech, Ed., Pergamon Press, pp. 169–183.

[42] Sims, C. Determining the conjugacy classes of a permutation group. In *Computers in Algebra and Number Theory* (1970), G. Birkhoff and J. M. Hall, Eds., vol. 4, pp. 191–195.

[43] Ullman, J. *Computational Aspects of VLSI.* Computer Science Press, Rockville, Maryland, 1984.

[44] Valiant, L. A scheme for fast parallel communication. *SIAM Journal of Comput. 11* (1982), 350–361.

[45] Valiant, L., and Brebner, G. Universal schemes for parallel communication. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing* (1981), vol. 13, pp. 263–277.