COORDINATING PEBBLE MOTION ON GRAPHS, THE DIAMETER OF PERMUTATION GROUPS, AND APPLICATIONS

Daniel Kornhauser
Computer Science
MIT
Cambridge, MA 02139

Gary Miller
Mathematics
MIT
Cambridge, MA 02139

Paul Spirakis
Computer Science
New York University
New York, NY 10012

Abstract. We consider the following generalization of the familiar '15-puzzle' which arises from issues in memory management in distributed systems: Let G be a graph with n vertices with $k < n$ pebbles numbered 1,...,k on distinct vertices. A move consists of transferring a pebble to an adjacent unoccupied vertex. Is one arrangement of the pebbles reachable from another?. We present a P-time decision algorithm, and prove matching $O(n^3)$ upper and lower bounds on the number of moves required. These results extend those of Wilson (1974), who considered G biconnected and k=n-1, with no analysis of number of moves.

We also consider the question of permutation group diameter. Driscoll and Furst (1983) obtained a polynomial upper bound on the diameter of groups generated by bounded length cycles. We have the following subexponential bound for certain unbounded cycles: If G (on n letters) is generated by cycles, one of which has prime length $p < 2n/3$, and G is primitive, then $G = A_n$ or S_n and has diameter $< 2^{6\sqrt{p}+4}n^8$.

1. Introduction

The management of memory in totally distributed computing systems is an important issue in hardware and software design. On an existing hardware network of devices, there is the problem of how to coordinate the transfer of one or more indivisible packets of data from device to device without ever exceeding the memory capacity of a device. Depending on the severity of the memory capacity, a considerable number of intermediate transfers may be necessary to clear a "path" for the movement of a data packet along a network. A combination of almost filled devices and a network configuration with few paths can, in fact, make impossible the transfer of the data packets intact.

Suppose we consider a simplified version of the above problem, where each device has unit capacity and each packet occupies one unit of memory. Then at any moment in time, any given device is either empty or is totally filled. Suppose also that at any time each data packet resides in some device. It is also assumed that only one packet may be moved at a time, from its current device to any empty immediately adjacent device. Under these assumptions, it is interesting to know whether it is possible to start from one given distribution of the packets in the network, and end with another given rearrangement, and to know how many moves are required when the rearrangement is possible.

This version of the network problem immediately translates into the following movers' problem on graphs:

Let G be a graph with n vertices with $k < n$ pebbles numbered 1,...,k on distinct vertices. A move consists of transferring a pebble to an adjacent unoccupied vertex. The problem is to decide whether one arrangement of the pebbles is reachable from another, and to find the shortest sequence of moves to find the rearrangement when it is possible.

It is seen that this latter problem is a generalization of Sam Loyd's famous "15-puzzle". In this puzzle, 15 numbered unit squares are free to move in a 4x4 area with one unit square blank. The problem is to move from one arrangement of the squares to another. One can easily show that this puzzle is equivalent to the graph puzzle on the square grid in Figure 1, with 15 numbered pebbles on the vertices and one blank vertex.

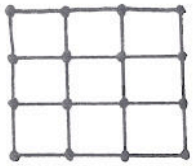

Figure 1. 15-Puzzle Graph

In the case that G is biconnected and $k = n - 1$, Wilson (1974) gave an efficient decision procedure. However, he did not consider the number of moves required for solution; a naive implemenation of his proof yields solutions requiring exponentially many moves. We provide a simplified proof of the decision procedure, and in this way, an $O(n^3)$ upper bound is obtained for the number of moves required in the Wilson case.

Then we generalize the decision procedure to all

0272-5428/84/0000/0241$01.00 © 1984 IEEE

graphs and any number of pebbles, and we show that again at most $O(n^3)$ moves are needed and can be efficiently planned. Finally, we find an infinite family of graph puzzles for which it is proved that $O(n^3)$ moves are needed for solutions. Thus the upper and lower bounds match to within a constant factor.

A topic of related interest is the subject of permutation groups and their diameter with respect to a set of generators. Briefly, the diameter of a permutation group G with respect to a set S of generators for G is defined to be the smallest positive integer k such that all elements of G are expressible as products of the generators of length at most k.

Consideration of the pebble coordination problem leads naturally to questions about permutation groups. Consider the graph in Figure 2, with vertex x blank and pebbles $a_1, ..., a_t, c_1, ..., c_r, b_1, ..., b_s$, and y on the other vertices. It is seen that any sequence of moves from this position will, upon the first return of the blank to x, net one of the following permutations on the pebbles: $A = (c_1 c_2 ... c_r y a_t ... a_2 a_1)$ or $B = (y c_r ... c_2 c_1 b_1 b_2 ... b_s)$ or $C = (b_1 b_2 ... b_s y a_t ... a_2 a_1)$ or A^{-1}, B^{-1}, C^{-1} or the identity permutation. Hence the set of rearrangements of the pebbles (with x blank) is the group of permutations generated by $S = \{A, B, C, A^{-1}, B^{-1}, C^{-1}\}$. Deciding whether a rearrangement is solvable amounts to testing membership of the corresponding permutation in the group generated by S; minimum number of moves is clearly related to the shortest product of generators yielding the permutation.

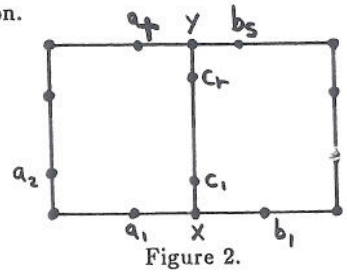

Figure 2.

We view the introduction of algebraic methods as useful for the solution of movers' problems. Whereas general geometric movers' problems are PSPACE-hard (Reif (1979); Hopcroft et. al. (1984)), it is hoped that the techniques introduced for the solution of the pebble coordination problem may be applicable to special cases of the general geometric problem.

We now briefly discuss some results in permutation group membership and diameter questions. Furst, Hopcroft and Luks [FHL] give a $O(n^6)$ analysis of Sims' [S] algorithm for deciding whether a given permutation g is in $G(S)$, the group generated by S. Later Knuth [K] and Jerrum [J2] gave algorithms with successively better upper bounds $O(n^5 log n)$ and $O(n^5)$. Thus the analogue of the graph decision problem is in P. One also immediately has a P-time criterion for deciding solvability of the Rubik's Cube and the Hungarian Rings puzzles. The situation is not as fortunate when one tries to find the length of the shortest generator sequence for a given permutation: Jerrum [J1] has recently shown this to be PSPACE-complete! The difficulty may be related to the fact that some groups may have superpolynomial diameter. For example, the group G generated by the single permutation (12)(345)(6789 10)...(...s) where s is the sum of the first n prime numbers, can be shown to have diameter roughly on the order of $2^{O(\sqrt{n})}$. This contrasts with the analogous question for the pebble coordination problem, where no solution can ever require more than $O(n^3)$ moves. Thus finding the length of shortest move sequences is in NP (on the other hand, Goldreich [G] has recently shown that it is NP-complete!). Therefore the group diameter question is in some sense more general, and probably more difficult, than the corresponding question for pebble motion.

There are nonetheless some interesting recent results concerning upper bounds on group diameter, for special generating sets. Driscoll and Furst [DF] have shown that if all the generators are cycles of bounded length, then the group has $O(n^2)$ diameter where n is the number of letters that the group acts on. More recently, McKenzie [M] obtained the upper bound $O(n^k)$ on diameter for groups, each of whose generators moves at most k letters. This is polynomial if k is bounded.

The foregoing results leave open the question of a group's diameter when the generators are arbitrary (not of bounded length) cycles. In chapter 3 we informally discuss certain generalizations of the Hungarian Rings puzzle, and find sufficient conditions for the required number of moves to be polynomial. Examples which do not meet these sufficient conditions are offered as possible candidates for groups with superpolynomial diameter. The rest of chapter 3 consists of a number of new results in permutation groups, which extend classical theorems by providing upper bounds on diameter. We obtain the following theorem as a corollary:

If G (on n letters) is generated by cycles, one of which has prime length $p < 2n/3$, and G is primitive, then $G = A_n$ or S_n and has diameter less than $2^{6\sqrt{p}+4} n^8$.

This is a moderately exponential upper bound, but is nonetheless superpolynomial. It remains of interest to know whether the bound can be significantly improved, or whether the diameter really can be this large.

At the end of the paper we present conjectures, open problems, and suggestions for further research in movers' problems and permutation group diameter.

2. Coordinating Pebble Motion on Graphs

In this chapter we will solve the pebble coordination problem given in the introduction:

Let G be a graph with n vertices with $k < n$ pebbles numbered 1,...,k on distinct vertices. A move consists of

transferring a pebble to an adjacent unoccupied vertex. The problem is to decide whether one arrangement of the pebbles is reachable from another, and to find the shortest sequence of moves to find the rearrangement when it is possible.

2.1. General remarks

We make the assumption that the set of occupied vertices of G is the same in both the initial and final positions. Then two positions define a permutation on the pebbles in a natural way, and so we can readily introduce the methods of group theory. There is no loss of generality, as we can show how to efficiently convert a puzzle into this form.

We also assume that all graphs are *simple*, that is, no two vertices are directly joined by more than one edge, and no vertex is joined to itself by an edge. It is clear that if a graph G is nonsimple, we can remove the "extraneous" edges to get a simple graph G', and the graph puzzle on G' is exactly equivalent to that on G, both with respect to solvability and the number of moves needed to solve it. Hence there is no loss of generality in making this assumption.

Since the set $R(P)$ of permutations induced on the pebbles by going from some fixed initial position P to reachable positions forms a group under composition, our task is to analyze the structure of the group $R(P)$.

It turns out to be natural to divide the analysis of $R(P)$ into two cases:

1. $R(P)$ is a transitive permutation group, i.e. any pebble can move to where any other pebble is located, without changing the set of occupied vertices.
2. $R(P)$ is an intransitive permutation group.

Case 2 occurs, intuitively, when the graph G contains an isthmus which is too long, compared to the number of blanks, to be "crossed" by a pebble. The graph of Figure 3 consists of a simple nonclosed path of edge length m which connects subgraphs A and B. Suppose we wish to move pebble T from v to w. Since A has no blank vertices, it is clear that T can reach w if and only if B has $m+2$ or more blank vertices. Therefore, the number of blanks has a direct effect on the ability of pebbles to cross isthmuses. Conversely, the lengths of the isthmuses will determine whether or not certain pebbles can cross from one component into another. And the more uncrossable isthmuses there are, the greater the number of transitive constituents ("orbits") the pebbles get divided into.

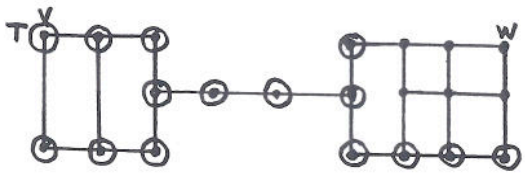

Figure 3. The Isthmus

It turns out that $R(P)$ is the direct product of its actions on the orbits. Thus in a sense the puzzle decomposes into independent transitive "subpuzzles" on pebbles in an orbit situated on an appropriate subgraph of G (the proof will be in the final version). The intuition is that we can "move blanks" temporarily to a subpuzzle site, solve the subpuzzle, and return the blanks without disturbing disturbing the pebbles in the other subpuzzles.

The solvability of a puzzle therefore reduces to the analysis of solvability of its transitive subpuzzles, i.e. transitive $R(P)$ (case 1).

We now indicate how to define the subpuzzles. (An efficient algorithm for actually determining them will be given in the final version.) Let $R(P)$ have orbits $O_1, \ldots, O_r$ such that $\sum_{i=1}^{r} |O_i| = k =$ number of pebbles. Let $G_i =$ the graph consisting of the vertices of G reachable from the initial position by pebbles in O_i (here it isn't required that the occupied vertices be an invariant set), together with the edges of G both of whose endpoints are in this set of vertices.

Then we define the i-th subpuzzle to have starting position consisting of G_i with pebbles on it as induced by the initial position of the entire puzzle. Similarly we define the ending position of the i-th subpuzzle. Note that an obvious necessary condition for solution of a subpuzzle is that the set of pebbles in its initial and final positions be identical (up to a reordering).

Example

The initial position in Figure 4 induces the three subpuzzle initial positions in Figure 5.

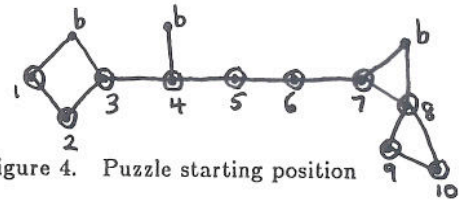

Figure 4. Puzzle starting position

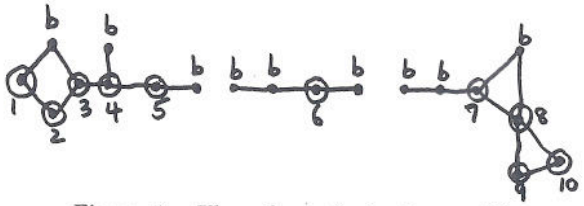

Figure 5. The subpuzzle starting positions

2.2. Criterion for transitive puzzles

Having reduced the general problem to the case where the pebbles move transitively, we now describe the solution for the transitive case.

First we need a few definitions. Define a *polygon* to be a graph consisting of a simple closed path containing at least two vertices (where a simple closed path is a path

from a vertex to itself which visits no intermediate vertex more than once). A polygon looks like a "loop" containing two or more vertices (see Figure 6). Let T_0 be the other graph shown in Figure 6.

Figure 6. A Polygon; graph T_0

Theorem 1

The following is a criterion for solvability for transitive puzzles.

1a. If G is biconnected, with $k = n - 1$ pebbles, then use Wilson's criterion [W]: Let G be a biconnected graph on n vertices, other than a polygon or T_0, with one blank vertex. If G is not bipartite, then the puzzle is solvable. If G is bipartite, then the puzzle is solvable iff the permutation induced by the initial and final positions is even.

1b. If G is biconnected, not a polygon, and $k < n - 1$, then the puzzle is solvable.

2. If G is separable (i.e. only 1-connected), the puzzle is solvable.

Remarks

Since bipartitism can be tested in polynomial time, Wilson's criterion is polynomial time.

For G a polygon, only cyclical rearrangements of the pebbles are possible, so it is easy to check reachability in this case. For the special graph T_0, we can simply precalculate (by exhaustive search) a table of all pairs of positions, indicating which pairs are mutually reachable. Table lookup is constant time, hence we have a P-time decision algorithm for all biconnected graphs with one blank.

1b. and 2. are new results.

Theorem 2 (stated and proved later) gives an $O(n^3)$ upper bound on the number of moves required for solutions of puzzles, based on an analysis of the proof below of Theorem 1.

2.3. Proof of Theorem 1

2.3.1. Case 2: Separable graphs

We start by proving the result for separable graphs, since this is a new result.

A separable graph has one or more cutpoints, and so is either a tree, or a tree-like structure containing one or more biconnected components (see Figure 7).

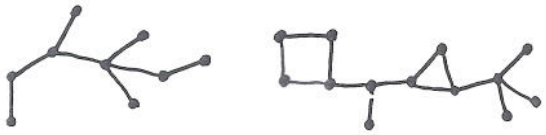

Figure 7. A Tree; a tree-like structure

2.3.1.1. Trees

We first consider the case that G is a tree. We will show that the group $R(P)$ is the symmetric group on the k pebbles by showing that it is k-transitive, i.e. the k pebbles in any order, $p_1, ..., p_k$, can be moved to any other order, $q_1, ..., q_k$.

The high-level plan is to move the pebbles from their initial vertices to k intermediate vertices, giving a position P' (which is not required to have the same vertices occupied as in the starting position P). Then we reverse a sequence of moves which takes the pebbles in their final position to the same intermediate position. The net result is the desired reordering of the k pebbles, which established k-transitivity.

The strategy is to move one pebble at a time to intermediate vertices, which are chosen to be "out-of-the-way" so that once a pebble reaches its target, it will not need to be moved again while the other pebbles are being moved. So, once a pebble reaches its target, we can "prune" the puzzle by removing the vertex from the graph, along with the pebble, and proceed to move the remaining pebbles on the smaller graph.

Keys to the success of the above strategy are that

1. The puzzle was originally transitive (by hypothesis), so that the first

pebble can be moved to its intermediate vertex.

2. Intermediate vertices are chosen so that when they are pruned, the resulting puzzle is still transitive.

In this way, we can guarantee that all pebbles can be moved to these intermediate vertices.

Detailed plan

We first show how each pebble is moved into place, then how the places are chosen.

Showing 1-transitivity

The decomposition of the puzzle into its transitive subpuzzles (given in final version) shows that if a subpuzzle is a tree, then no isthmus has edge length $> m - 2$ (m = number of unoccupied vertices). Furthermore, no "branch", i.e. a path with one end of valence > 2, internal vertices all of valence 2, and the other end of valence 1 (the "leaf"), has edge length $> m - 1$. These facts can be seen intuitively by considering how many blanks are

needed to cross isthmuses and to reach leaves of branches; the proof is simply a formalization of this intuition.

It is not hard to see that these conditions are sufficient to ensure that a pebble can reach any vertex in the tree, i.e. that the subpuzzle is transitive. We proceed as follows. Suppose we wish to move a pebble p from vertex v_1 to vertex v_2. Consider the path through the tree, from v_1 to v_2; for each of its internal vertices of valence > 2, leave attached a single edge and its end vertex (which we call a "leaf"). This subgraph of G will be called G'. (See Figure 8.)

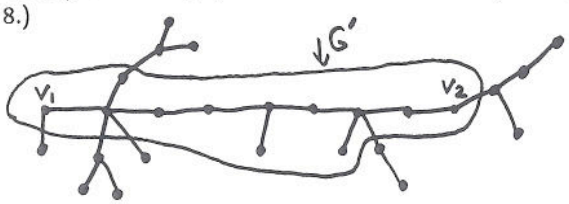

Figure 8. Graph G and subgraph G'

We claim that p can be moved from v_1 to v_2, remaining entirely within G'. The proof is informally as follows. We move spaces next to v_1 so that p can be moved to the leaf closest to v_1. Then relocate the spaces so that p can now move to the next leaf in the direction of v_2. Continue in this way, "hopping" p from leaf to leaf, until p finally reaches v_2. All these steps are possible, because the isthmuses were assumed to have length $\leq$ number of spaces - 2.

This completes the demonstration of transitivity.

Selection of Intermediate Vertices

We wish to select vertices which, when pruned, leave the remaining puzzle transitive.

Now, it is not hard to see that if we prune a leaf from a branch with edge length > 1, and decrease the number of pebbles by 1, then the above transitivity conditions on the puzzle are preserved, and so the reduced puzzle is still transitive (if the branch had length 1, then pruning would remove the whole branch, thus potentially creating a long isthmus or a long branch). Hence, choose as an intermediate vertex the leaf of any branch of length > 1. After filling it with a pebble and pruning, we repeat the choice on the pruned graph. In the case that all branches are of length 1, then locate a vertex which is adjacent to two or more leaves (this is always possible, in this case). Then it is not hard to see that one of these leaves can be used as the next target vertex, and still preserve transitivity when pruned. We call these "multiple leaves" (see Figure 9).

Figure 9. a: Biconnected leaf; b: "Multiple leaves"

2.3.1.2. Tree-like structures

We now indicate how to prove case 2 for separable graphs which are not trees, but rather are tree-like structures containing one or more biconnected components.

This is similar to the tree case, with a few changes. We establish k- transitivity by moving pebbles successively to intermediate vertices. The existence of a biconnected component along the way does not hinder a pebble's movement to a target vertex, since by case 1 (proved below) biconnected puzzles are clearly transitive.

We choose the intermediate vertices, as before, to be leaves of branches having length > 1, or else "multiple leaves". However, it may happen now, that there is no branch of length > 1, and no multiple leaves. In this case, it can be shown that there must be a biconnected component H which is attached to the rest of the graph by only one vertex (see Figure 9). If we imagine shrinking H down to a point, we can think of H as a leaf of the branch to which it is attached. What we will do is to load up the vertices of H, one by one, with pebbles, leaving only the junction blank. Then we will put the pebbles in H into the desired order. Finally we "prune" H, with its pebbles, from the graph.

H is filled with the desired pebbles as follows. If the pebble to load into H is already in H, then there is nothing to do. Otherwise, move the pebble to a vertex next to the junction to H, and move a pebble already in H which doesn't belong there, to a vertex in H next to the junction (this can be done, since H is transitive). Then swap the two pebbles. In this way, we can fill H with any desired set of pebbles.

Then we can put the pebbles in H into the desired order, as follows. We can obtain a swap of two pebbles next to the junction of H, as in Figure 10. If H is not a polygon, then by the proof of 1a (to be proved below) we can move the pebbles of H 2-transitively. 2-transitivity and a swap generates, by conjugation, all swaps, and so we can generate any rearrangement of the pebbles in H. If H is a polygon, then by moving the pebbles around H, we get a cyclic permutation which, when conjugated with the swap, gives us enough swaps to generate all reorderings of the pebbles in H.

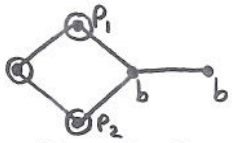

Figure 10. Swapping two pebbles

Having loaded H, and the pebbles in H put into the intended order, we do the following incomplete pruning of H: remove H and its pebbles, except for the junction and one incident edge. (We leave the edge hanging, to avoid the possibility of the remaining graph being a polygon.)

It is not hard to see that the remaining graph is still transitive.

Since there is always a branch of length > 1, or a multiple leaf, or a "biconnected leaf" (as above), eventually all pebbles can be moved into place.

This completes the proof of case 2.

2.3.2. Case 1a: Biconnected, 1 blank

It is a well-known fact in graph theory that a biconnected graph, other than a single edge, can be viewed as being "grown", by starting with a polygon graph and successively adding zero or more "handles" (a handle is a simple path with 0 or more internal vertices). A biconnected graph which can be grown by adding i handles to a polygon, appears pictorially to consist of $i+1$ simple loops joined together in some way. This number of loops is called the *Betti number* of the graph. We will often denote a biconnected graph with Betti number i by the term "T_i-graph". Wilson's theorem will be proved by induction on the Betti number of the graph. We skip the T_1-graphs (the polygons) and begin the induction with the T_2-graphs (except T_0).

The main step is to show that the group of possible induced permutations always contains the alternating group A_{n-1} on the $n-1$ pebbles. The final step is to determine whether the group is A_{n-1} or S_{n-1}. The group will be S_{n-1} iff it contains an odd permutation, and it is easy to see that there is an odd permutation iff the graph has a closed path of odd length. As a graph has a closed path of odd length iff it is not bipartite, we see that the group is A_{n-1} if the graph is bipartite, and S_{n-1} if the graph is not bipartite. Therefore, to check solvability on a bipartite graph, it is necessary and sufficient that the induced permutation be even; on a nonbipartite graph, the puzzle is always solvable.

To show that the group of induced permutations contains the alternating group, we show how to obtain a 3-cycle and how to obtain 2-transitivity.

From this, the alternating group is efficiently generated as follows: since A_n is efficiently generated by the set of all 3-cycles, it suffices to show how to efficiently generate, given a 3-cycle (123) and 2-transitivity, any 3-cycle (abc). Using a permutation T taking 1,2 to a, b respectively, the conjugate $T^{-1}(123)T$ is of the form (abd). If $d = c$, we're done. Otherwise, obtain by a similar conjugation a 3-cycle of the form (bce). If $d = e$, then using $A = (abd)$ and $B = (bcd)$ we can cancel out d by $A^2B^2 = (dba)(bdc) = (acb)$. Squaring the result, we get (abc). If $d \neq e$, then conjugate (abd) by (bce) to get (acd). Then using (abd) and (acd), and cancelling d as before, we get (abc).

The reason for using 2-transitivity instead of 3-transitivity is that proof of 2-transivity for graphs is easier and involves fewer exceptional cases than for 3-transitivity. The price we pay is transferred to the algebraic domain, in the form of some extra conjugations.

A 3-cycle is obtained roughly as follows. A T_2-graph looks like that pictured in Figure 11. Assume first that $r = 0$, i.e. the center arc has no internal vertices. $A = (ya_t...a_1)$ and $B = (b_1...b_sy)$ are permutations induced by moving pebbles around, respectively, the left or right loops. Then $ABA^{-1}B^{-1} = (yb_sa_1)$, a 3-cycle. If $r > 0$, then $ABA^{-1}B^{-1}$ is a product of two swaps; we can obtain a 3-cycle from this, if the graph induces 4-transitivity. In this connection, we use the following lemma (use k=3 below).

Lemma 1

Let H be a nonseparable graph, and G be the result of adding a handle to H. If the handle has k internal nodes, then G is $k+1$-transitive.

The proof will be given in the final version. (Intuitively, we move one pebble after another onto the handle, using 1-transitivity of H to reach the handle. Moving a pebble to the handle does not disturb pebbles already there. To prove 1-transitivity uses induction on the Betti number.)

We take this opportunity to mention the following quantitative version, which will be used later. Its proof involves showing that 1-transitivity is $O(n^2)$, by induction on the Betti number (the basis is the polygon, which is easily seen to be $O(n^2)$); then k-transitivity, for bounded k, is also $O(n^2)$. (Proof in final version.)

Lemma 2

For any bounded k, the $k+1$-transitivity guaranteed by Lemma 1 can be done in $O(n^2)$ moves.

Now, using Lemma 1 it is easy to enumerate those T_2-graphs, where $r > 0$, which are not 4-transitive. However, of these graphs, inspection shows (details in final version) that all but T_0 produce a 3-cycle. Hence T_0 is the only T_2-graph which does not induce a 3-cycle. We then show that all T_i-graphs, $i > 2$ give a 3-cycle, because they are formed by adding handles to a T_2-graph which can induce the 3-cycle. The hole in the induction due to T_0 will be taken care of with no difficulty.

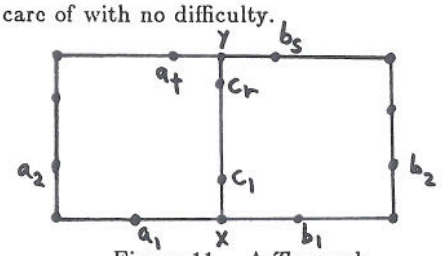

Figure 11. A T_2-graph

2-transitivity will also be shown by induction. It can be shown that all T_2-graphs are 2-transitive, by Lemma 1 above. Then we show how adding a handle to a 2-transitive graph yields a 2-transitive graph.

Putting 3-cycle and 2-transitivity together, we will conclude that all T_i-graphs, $i >= 2$, generate at least the alternating group, except T_0.

2.3.3. Case 1b: Biconnected, > 1 blank

If a biconnected graph is not a polygon, then there is

a vertex v of valence > 2. By hypothesis we have at least 2 blanks. Hence by moving one blank to v, and another blank to a vertex adjacent to v, we can swap two pebbles which are adjacent to v, as in Figure 10. 2-transitivity follows as in the proof of 1a. Putting the swap and 2-transitivity together, the whole group of permutations is generated.

2.4. $O(n^3)$ Upper Bound

Theorem 2

Let G be a graph. Let $n = |V(G)|$. If labeling g can be reached from labeling f at all, then this can be done within $O(n^3)$ moves, and such a sequence of moves can be efficiently generated.

Sketch of proof (details in final version)

a. G biconnected

We can show that a 3-cycle can always be obtained in $O(n^2)$ moves (either $ABA^{-1}B^{-1}$ gives a 3-cycle in $O(n)$; or we get a product of two swaps, in which case we can do 4-transitivity in $O(n^2)$ moves to get a 3-cycle), and that 2-transitivity requires at most $O(n^2)$ moves (see Lemma 2 above). Then by the algebra given in the proof of Theorem 1 (case 1a) for obtaining (abc) from (123) and 2-transitivity, we obtain any 3-cycle within $O(n^2)$ moves. Since any element of A_n is a product of $O(n)$ 3-cycles, the total for A_n is $O(n^3)$. If the group is S_n, then any permutation is a product of an odd permutation and an element of A_n. An odd permutation is generated by a closed path of odd length in $O(n)$ moves. Hence S_n also requires at most $O(n^3)$ moves.

b. G Separable and transitive

If G is a tree, then the proof of transitivity implies that at most $O(n)$ moves are needed to move a pebble anywhere; so the proof of Theorem 1, case 2, implies an upper bound of $O(n^2)$ to move all the $k < n$ pebbles. The existence of biconnected subgraphs, however, can force us to an upper bound of $O(n^3)$ (see Figure 12, which is essentially the same as the graph used in the lower bound proof below).

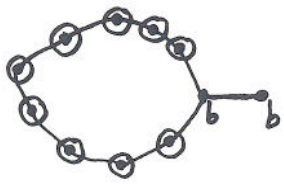

Figure 12.

c. G intransitive

The puzzle can be solved by solving the transitive subpuzzles on subgraphs $G_1, ..., G_r$. It is not hard to show that the $O(n_i^3)$ upper bounds on each subpuzzle combine to give a $O(n^3)$ upper bound for the whole puzzle.

2.5. $O(n^3)$ lower bound

We now complement the above result with a lower bound which matches, to within a constant factor.

Theorem

There exists a constant $c > 0$ and an infinite sequence of graph puzzles Puz_i on increasingly large graphs G_i with n_i vertices, such that for each i, Puz_i requires at least cn_i^3 moves for solution.

Proof

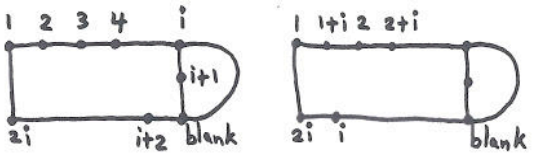

Figure 13. Lower bound graph

Let Puz_i consist of graph G_i shown in Figure 13, with $2i + 1$ vertices and $2i$ pebbles, and starting and ending positions as shown. We will show that Puz_i requires $O(i^3)$ moves, as follows. A move sequence that does not waste moves (by retracing move sequences just made) is seen to consist of cycles A, B and their inverses, interspersed in some order (e.g. $ABAAAABA^{-1}B$). It would be wasteful to do B twice in succession, since this would cancel itself. Hence a move sequence can be represented by the form $A^{i_1}BA^{i_2}B...A^{i_k}BA^{i_{k+1}}$ where i_j is a nonzero integer (positive or negative), except i_1 and i_{k+1} may be 0.

Now consider the "entropy function" of position

$E = \sum_{j=0}^{i}$ (shortest circular distance from pebbles j to $j + i$)

where circular distance is either clockwise or counterclockwise. Initially, $E = i^2$; at the end, $E = i$. Change in E is $i^2 - i$.

It is seen that A does not change E, and B changes E by 0 or by 2. Hence to effect the change in E requires $O(i^2)$ occurrences of B in the move sequence. But because occurrences of A^{i_j} and B alternate, this implies that A occurs at least $O(i^2)$ times. Since the number of moves to perform the cycle A is $O(i)$, we need at least $O(i^3)$ moves for solution.

This completes the proof of the lower bound.

3. The Diameter of Permutation Groups

As mentioned in the introduction, this chapter is concerned with the diameter of permutation groups generated by sets of cyclic permutations. We begin with some examples of generator sets which yield groups of polynomial diameter, then speculate on some conditions on the generator set which might give groups of superpolynomial diameter. The main part of the chapter consists of theorems which give information about the diameter of a group under various conditions. They imply the result given in the introduction, which is a moderately exponential upper bound on the diameter of groups generated by cycles which satisfy a few conditions.

3.1. What is not of exponential diameter, and what might be

The Hungarian Rings puzzle consists of two intersecting circular rings in which distinguished marbles circulate. The problem is to obtain a desired rearrangement of the marbles by a sequence of operations, where an operation consists of circulating the marbles in one of the rings. This problem immediately translates into the permutation problem of determining membership in the group generated by two intersecting cyclic permutations. By [HFL], we can decide membership in polynomial time; however, it is of interest to know how many "moves" are required, i.e. the length of the shortest word which gives the desired permutation.

In Figure 14 is shown schematically two cyclic permutations which intersect at two points. This corresponds to the commercial version of the Hungarian Rings. Note that this is not like a pebble puzzle on a T_3-graph, because only A and B are possible, and not the third loop; the Hungarian rings is a physical movers' problem which imposes this restriction mechanically. This gives reason to expect that the number of moves may need to be larger in some permutation puzzles than in the pebble puzzles.

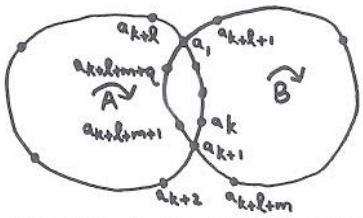

Figure 14. The Hungarian Rings

What is the diameter of the group generated by these two cycles? It is first useful to observe that, if some arc C contains at least r internal nodes, and an arc D on the other cycle contains at least one internal node, then we can get $r+1$-transitivity in $O(rn)$-long moves. This is done, roughly speaking, by moving one desired marble after another to a_1, then rotating it onto arc C. The cycle not containing arc C is rotated to bring the next desired marble to a_1, leaving the contents of C undisturbed. Arc D serves as temporary "storage" of a marble which, already on arc C, needs to be removed from C and then placed onto C at the right place.

Suppose that in the figure, $l >= 6$ and $m >= 1$. Then we have efficient 6-transitivity. Now $ABA^{-1}B^{-1} = P = (a_1 a_{k+l+m+q} a_{k+l})(a_k a_{k+1} a_{k+l+m})$. Using 6-transitivity, we can find a permutation P_1 which sends $a_1, a_{k+l+m+q}, a_{k+l}$ to $a_1, a_{k+l}, a_{k+l+m+q}$ respectively and fixes a_k, a_{k+1}, a_{k+l+m}. Then conjugating P by P_1 gives $P_2 = (a_1 a_{k+l} a_{k+l+m+q})^* (a_k a_{k+1} a_{k+l+m})$. P_2 is a product of two 3-cycles, one the inverse of the one in P, the other the same as the other in P. So $PP_2 = (a_k a_{k+l+m} a_{k+1})$, a 3-cycle. Then, using 3-transitivity, we get the alternating group. Hence $l >= 6$ and $m >= 1$ implies a polynomial diameter for the Hungarian Rings puzzle with the rings intersecting at two places.

What happens if the number of intersection of the two cycles is some number k greater than 2? By similar reasoning to the above, we get $ABA^{-1}B^{-1}$ to be a product of k 3-cycles. Then a conjugation argument similar to the above yields that, if we have $3k$-transitivity, then we can get a single 3-cycle. How do we get efficient $3k$-transitivity? Well, an arc of $3k-1$ nodes and another arc with one node would suffice. Or, in the case that k is bounded, then it is known [DF] that the existence of k-transitivity is enough to ensure k-transitivity in $O(n^k)$-long words, which is polynomial for fixed k. However if k is large, then this bound is exponential. If no arc has enough nodes in it, there might be no efficient way to get the desired degree of transitivity.

The foregoing considerations suggest that a good candidate for a Hungarian Rings puzzle with superpolynomial diameter is one with lots of crossings and no long arcs (see Figure 15). To be more quantitative, suppose that there are k equally spaced crossings. Then the arcs have length on the order of n/k. We want this to be less than $3k$. So: $n/k < 3k$, i.e. $k > \sqrt{n/3}$. This suggests that we should use at least on the order of $\sqrt{n}$ crossings to create a likely exponential puzzle. It would be of great interest to establish an exponential or moderately exponential lower bound for some of these "candidate" puzzles.


Figure 15. Hungarian Rings with multiple crossings

We now leave these examples and speculations, and state some results about the diameter of permutation groups (proofs in final version).

3.2. Some results about the diameter of permutation groups

The following are classical theorems in the theory of permutation groups.

Theorem A

If the group G on n letters is k-transitive and $k > n/3 + 1$, then $G = A_n$ or S_n.

Theorem B

If G is primitive on n letters, and a subgroup H moves only $m < n$ letters and is primitive on them, then G is $n - m + 1$-transitive.

We prove the following versions of these theorems, which give information about the diameter:

Theorem 1

If group G on n letters is k-transitive in words of length $<= L$, the generator set S is closed under inverses, and $k > n/3 + 1$, then $G = A_n$ or S_n and $\mathrm{Diam}(G(S)) < 4n^2L$.

Theorem 2

If G is primitive on n letters, and H is the primitive subgroup generated by a cyclic permutation of prime length $p < n$, and the generator set S is closed under inverses, then G is $n - p + 1$-transitive using words of length $< 2^{6\sqrt{p}+1}n^3(n^2 + diam(H(S)))$.

Theorem 3

If G is primitive on n letters, and H is a 2-transitive subgroup which moves only $2 <= m < n$ letters, and the generating set S is closed under inverses, then G is $n - m + 1$-transitive using words of length $< 2^{9\sqrt{p}+1}n^3(n^2 + diam(H(S)))$.

We were not able to prove an effective version of theorem B for arbitrary primitive H, but did obtain the special cases contained in theorems 2 and 3.

The following is an easy corollary.

Theorem 4

If a primitive group G on n letters is generated by a set S of cyclic permutations, one of prime length $p < 2n/3$, then G is A_n or S_n, and $\mathrm{Diam}(G(S)) < 2^{6\sqrt{p}+4}n^8$.

3.3. Proofs of the Theorems

In this section, we motivate the proofs of Theorems 2 and 3, and prove Theorem 4 as a corollary of Theorems 1 and 2. Complete proofs of all the theorems will appear in the final version.

First we will need the following preliminary Lemmas.

Lemma 2a

If G is primitive on n letters, and H is the primitive subgroup generated by a cyclic permutation of prime length $p < n$, and the generator set S is closed under inverses, then there exists a $g \in G$ which takes $D = \mathrm{Domain}(H)$ to D', such that D and D' overlap on exactly $m - 1$ letters, and g has wordlength $< 2^{6\sqrt{p}}(n^2 + diam(H(S)))$.

Lemma 3a

If G is primitive on n letters, and H is a 2-transitive subgroup which moves only $2 \leq m < n$ letters, and the generating set S is closed under inverses, then there exists a $g \in G$ which takes $D = \mathrm{Domain}(H)$ to D', such that D and D' overlap on exactly $m - 1$ letters, and g has wordlength $< 2^{9\sqrt{m}}(n^2 + diam(H(S)))$.

The purpose of the Lemmas is roughly as follows. By making a set of letters overlap itself by all but one letter, and repeating this process, it is possible to build a tower of conjugates of H whose domains look like the diagram in Figure 16. It is then possible to achieve $n - m + 1$ transitivity by using the fact that the domains intersect, to move any letter to the right end of the bottom row (as pictured in the figure), then move any letter to the right end of the next-to-bottom row without disturbing the previous element, and so on to get $n - m + 1$ transitivity. The details will be given in the final version, as well as an analysis of the wordlength needed to do these operations. This, combined with Lemma 2a, gives Theorem 2, and with Lemma 3a it gives Theorem 3.

Figure 16. A tower of conjugates

Proofs of the Lemmas

We will motivate the proofs of the Lemmas. Roughly speaking, we first find a permutation which maps D to a D_1 which overlaps D partially but not totally. Then a conjugating device is repeated, which increases the overlap with each iteration, but never reaches total overlap. Naturally, we must reach a D' where overlap is all but one letter. Counting arguments (quite different for the two Lemmas) show that the overlap increases can be chosen large enough so that at most $O(\sqrt{D})$ iterations are needed to reach overlap of all but 1. Since each iteration, which involves conjugating the permutation by a new permutation of short wordlength, at most doubles the wordlength of the permutation, the total wordlength can be calculated to be $O(2^{\sqrt{D}})$.

The details and explicit constants will be provided in the final version of the paper.

Proof of Theorem 4

The generator h of the cyclic subgroup $H = H_0$ of

order p is (by hypothesis) in the generator set of G. So $Diam(H_0) \leq p$. We are not assuming that the generators are closed under inverses, but because they are cyclic of order $\leq n$, the inverse of a generator is at most the n-th power of that generator. Hence the wordlength is at most a factor of n longer than obtained previously, where we assumed closure under inverses. Therefore, $n-m+1$-transitivity requires wordlength

$< 2^{6\sqrt{p}+1} n^4 (n^2+p)$

$< 2^{6\sqrt{p}+2} n^6$.

Then, as $m < 2n/3$, we have $n-m+1 > n/3+1$, so using Theorem 1, we get an additional factor of $4n^2$, giving $Diam(G) < 2^{6\sqrt{p}+4} n^8$, which proves the corollary.

This last theorem provides a partial extension of [DF]'s upper bound for bounded cycles to unbounded cycles. It would be desirable to generalize the result to apply to all cycles, and to find a matching lower bound on diameter.

4. Conclusion and Open Problems

We have obtained some results in pebble coordination problems and the diameter of permutation groups. Specifically, we derived:

1. An efficient decision algorithm for the general pebble coordination problem on graphs.

2. $O(n^3)$ matching upper and lower bounds on the number of moves to solve pebble coordination problems.

3. $2^{6\sqrt{p}+3} n^8$ upper bound on diameter of A_n or S_n when generated by cycles, one of which has prime length $p < 2n/3$.

We see 1. as being a complete and satisfactory result as it stands. It would be of interest to apply the algebraic methods used in the pebble movers' problem to special cases of the general geometric movers' problem which may admit an algebraic approach.

2. could stand a number of refinements.

a. Find exact constants in the O-terms.

b. It would be useful to at least have an efficient algorithm which approximates the number of moves required. For it seems that only a small fraction of the graph puzzles actually require $O(n^3)$ moves. As an example, it is not hard to show that the "15-puzzle" generalized to square grids of arbitrary size (with one blank) requires only $O(n^{3/2})$ moves (where n is the number of vertices).

3. is only a first step towards understanding the diameter of groups generated by arbitrary cycles. A number of related questions are open:

a. Is the upper bound in 3. tight? Is there a corresponding lower bound of $O(2^{O\sqrt{p}})$ for some instances of 3. ? This would settle the following well-known open problem:

b. Can a transitive group have larger than polynomial diameter for some generator set? Can this be the case for A_n or S_n?

c. Can the upper bound in 3. be generalized to less restrictive conditions on the generating cycles? Is it even true that the following conjecture holds?:

d. Is the diameter of a group, relative to any generating set, always bounded above by $O(n^{\sqrt{n}})$? E.g. the group generated by S ={(12)(345)...(...[sum of first n primes])} has diameter $O(2^{\sqrt{n}})$, which satisfies the conjecture.

Bibliography

[DF] J.R. Driscoll and M.L. Furst, "On the diameter of permutation groups", 15th STOC, 1983, pp. 152-160.

[FHL] M. Furst, J. Hopcroft, E. Luks, "Polynomial-time algorithms for permutation groups", 21st FOCS, 1980, pp. 36-41.

[G] O. Goldreich, "Shortest Move-Sequence in the Generalized 15-Puzzle is NPH", manuscript, Laboratory for Computer Science, MIT, June 1984.

[HSS] J.E. Hopcroft, J.T. Schwartz, and M. Sharir, "On the Complexity of Motion Planning for Multiple Independent Objects; PSPACE-hardness of the 'Warehouseman's Problem' ", memo, Comp. Sci. Dept., Cornell Univ., 1984.

[J1] Mark Jerrum, "The Complexity of Finding Minimum-Length Generator Sequences", Internal Report CSR-139-83, Dept. of Comp. Sci., Univ. of Edinburgh, Aug. 1983.

[J2] Mark Jerrum, "A Compact Representation for Permutation Groups", 23rd FOCS, 1982, pp. 126-133.

[K] D.E. Knuth, "Note on Efficient Representation of Permutation Groups", 1980.

[M] Pierre McKenzie, "Permutations of Bounded Degree Generate Groups of Polynomial Diameter", manuscript, Dept. of Comp. Sci., Univ. of Toronto, Jan. 1984.

[R] J. Reif, "Complexity of the mover's problem and generalizations", 20th FOCS, 1979, pp. 421-427.

[SS] J.T. Schwartz and M. Sharir, "On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds", to appear in Adv. Appl. Math.

[SY] Paul Spirakis and Chee Yap, "On the combinatorial complexity of motion coordination", Tech. Rept., Courant Institute, N.Y.U., April 1983.

[Wie] Wielandt, "Finite permutation groups", Academic Press, New York, 1964.

[W] R.M. Wilson, "Graph puzzles, homotopy, and the alternating group", Journal of Comb. Theory (B) 16, 86-96 (1974).