FINDING HAMILTON CYCLES IN CUBIC DIGRAPHS AND RESTRICTED CAYLEY DIGRAPHS

by

RAMYAA

(Under the Direction of Robert W. Robinson)

ABSTRACT

This thesis explores randomized algorithms for finding Hamilton cycles in cubic digraphs and in diregular Cayley digraphs of degree 2 in which one generator is an involution. For random cubic digraphs two approaches for finding Hamilton cycles are discussed – a random permutation method and a Markov chain method.

The special properties of restricted Cayley digraphs which are relevant to finding Hamilton cycles in them are explored. Then, three algorithms are presented for finding Hamilton cycles in restricted Cayley digraphs.

Results of testing the algorithms on random cubic graphs and restricted Cayley digraphs of moderate size (1000 – 2000 vertices) are reported.

INDEX WORDS: cubic digraph, Cayley graph, Hamilton cycle, perfect matching, alternating cycle, Markov chain, randomized algorithm.

FINDING HAMILTON CYCLES IN CUBIC DIGRAPHS AND RESTRICTED CAYLEY DIGRAPHS

by

RAMYAA

B.E., Madurai Kamaraj University, India, 1997

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2004

FINDING HAMILTON CYCLES IN CUBIC DIGRAPHS

AND RESTRICTED CAYLEY DIGRAPHS

by

RAMYAA

Major Professor:	Robert W. Robinson
Committee:	E. Rodney Canfield Liming Cai

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2004

ACKNOWLEDGEMENTS

I would like to thank my committee members, Dr. Canfield and Dr. Cai, for their assistance and guidance for my graduate study.

This work could not have been completed without the help of my major professor Dr. Robinson. I would like to express my gratitude to him for caring and patience. Working with him has taught me how to do research. He always makes himself available to help me and I would like to express my thanks.

TABLE OF CONTENTS

Page

CHAPTER 1

INTRODUCTION

1.1 BASIC DEFINITIONS

A *graph* $G = (V, E)$ consists of a set V of *vertices* and a set E of unordered pairs of distinct vertices called *edges*. A *directed graph (digraph)* $G = (V, E)$ consists of a set V of vertices and a set E of ordered pairs of vertices of V called edges or *arcs*. The (random) graphs which are considered in this thesis are directed and *simple* (containing no loops or multiple edges).

The *order* of G is the number $|V|$ of vertices, denoted by n. A vertex u is *adjacent* to a vertex v if (u,v) is an edge of G, and the edge (u,v) is *incident* with the vertices u and v. The *degree* of a vertex v is the number of edges incident with it. The *in-degree* of a vertex v is the number of edges coming into its *out-degree* is the number of edges going out of it. A graph G is *r-regular* if every vertex in G has degree r. A *diregular digraph* of degree 2 is a regular digraph of in-degree and out-degree 2. A *cubic digraph* is a regular digraph with in-degree and out-degree 3.

A *path* from a vertex u to a vertex v in a directed graph $G = (V, E)$ is a sequence $(v_0, v_1, ..., v_k)$ of vertices such that $u = v_0$, $v = v_k$ and $(v_{i-1}, v_i) \in E$ for all i from u to v. Such a path is a cycle if $v_0 = v_k$ and the path contains at least one edge. It is a simple cycle if $v_1, v_2 ... v_k$ are distinct. If the path $(v_0, v_1, ..., v_k)$ is such that $(v_{i-1}, v_i) \in E$ for all odd i and $(v_i, v_{i-1}) \in E$ for all even i and $v_0 = v_k$, it forms an *alternating* cycle.

A *Hamilton cycle* in an undirected graph $G = (V, E)$ is a simple cycle that contains all the vertices in V. A graph that contains a Hamilton cycle is a *hamiltonian* graph; otherwise it is a *nonhamiltonian* graph.

Given an undirected graph, a *matching* is a subset of edges $M \subseteq E$ such that for all vertices $v \in V$, at most one edge of M is incident to v. We say that v is matched if some edge of M is incident with v. A *maximum matching* is a matching of the maximum cardinality, i.e. a matching M such that for any matching M', we have $|M| \geq |M'|$. A *perfect matching* is a matching in which every vertex is matched. The vertices incident to the edges of a matching M are *saturated* by M. All other vertices are *unsaturated.*

In an undirected graph, a *k-factor* is a k-regular spanning subgraph. Here, *spanning* means that all vertices are included. In a directed graph, a k-factor is a diregular spanning subdigraph of degree k.

A *bipartite graph* is a set of graph vertices decomposed into two disjoint sets such that no two graph vertices within the same set are adjacent.

An undirected graph $G' = (V', E')$ can be constructed from a directed graph $G(V, E)$ as follows: for every vertex v in V, there are two vertices v' and v'' in V'. For every edge (u, v) in E, there is an edge (u', v'') in E'. Then G' is an undirected bipartite graph. The graph G can be reconstructed from G', the bipartition $(\{v' : v \in V\}, \{v'' : v \in V\})$ and the 1-1 correspondence $\{(v', v'') : v \in V\}$.

In an undirected graph constructed from a directed graph, a perfect matching will correspond to a subdigraph of the directed graph in which each vertex has exactly one edge coming in and exactly one edge going out. Thus a 1-factor in G directly corresponds to a perfect matching in G'. So, if a perfect matching is found in G', the

corresponding 1-factor in *G* may contain only one cycle, in which case it is a Hamilton cycle.

Let Γ be a group, and let $S \subseteq \Gamma$ be a generating set. The Cayley digraph associated with (Γ, S) is the directed graph having one vertex associated with each group element and directed edges (g,h) whenever $hg^{-1} \in S$. The edges are labeled by the generators so that (g,h) is labeled by hg^{-1}. The Cayley digraph may depend on the choice of generating set even without edge labeling and is always vertex-transitive. Here we study restricted Cayley digraphs where Γ is a permutation group, on a finite set X, $|S| = 2$, and exactly one generator is an involution (a permutation that is its own inverse). In this setting, a permutation group on X is simply a set of permutations of X which are closed under composition. The group operation is composition, indicated by juxtaposition as above.

1.2 OUTLINE OF THE THESIS

This thesis explores randomized algorithms for finding Hamilton cycles in random cubic digraphs and in restricted Cayley digraphs. Since the problem of determining hamiltonicity in cubic digraphs is NP-complete [3], no polynomial time deterministic algorithm is known or expected. The complexity of determining the hamiltonicity of the restricted Cayley graphs is not known; all existing algorithms are heuristic or else exhaustive and exponential. Hence randomized algorithms for both problems are of interest.

The body of the thesis consists of two main sections.

Chapter 2 deals with random cubic digraphs and is organized as follows: generation of cubic random digraphs is described followed by two algorithms to find Hamilton cycles in them – the random permutation algorithm and the Markov chain method. Both methods use the fact that in the undirected graph constructed from the given random digraph, a perfect matching corresponds to a 1-factor. All Hamilton cycles are 1-factors, but not all 1-factors are Hamilton cycles. First, a perfect matching algorithm is applied, and if a perfect matching does not exist the graph is nonhamiltonian. Then the corresponding 1-factor is checked to see if it corresponds to a Hamilton cycle. In the first approach (random permutation) the graph is randomly permuted and a perfect matching is found in the permuted graph. This process of permuting, finding a perfect matching and checking for hamiltonicity is called a *trial*. In the second approach (Markov chain) a series of small changes is made to the original matching to produce a new perfect matching. This process of changing one matching into another and checking for hamiltonicity is also called a *trial*. Both approaches repeat the trials a number of times depending on the size of the graph and conclude heuristically that the graph is nonhamiltonian if a Hamilton cycle is not found.

Chapter 3 deals with Cayley graphs. First the structure of the restricted Cayley digraphs is studied leading to special properties which are relevant in finding Hamilton cycles. Then the algorithms for finding Hamilton cycles are developed. The diregular Cayley digraphs of degree 2 considered can be divided into edge-disjoint alternating cycles. If in each alternating cycle, only alternate edges are chosen, then the chosen edges from a 1-factor. Such a 1-factor can be checked to see if it consists of only one cycle – a Hamilton cycle. The algorithms obtain a 1-factor by choosing the odd or the

even alternate edges from each alternating cycle. Using the properties of the Cayley graphs, certain restrictions can be imposed on these choices. Three different heuristics (including one that is random) are developed using combinations of the properties of Cayley graphs which are discussed in Chapter 3.

CHAPTER 2

RANDOM CUBIC DIGRAPHS

In this chapter, generation of random cubic graphs and algorithms to find Hamilton cycles in such graphs is then described. The random graphs considered are without loops or multiple edges. A loop in a graph is an edge which joins a vertex to itself. A multiple edge occurs when there is more than one edge between two vertices (in the same direction).

2.1 GENERATION OF RANDOM CUBIC DIGRAPHS

In a cubic digraph, each vertex has three incoming edges and three outgoing edges. We can associate each vertex with its three outgoing edges, and then randomly choose incoming edges into which those edges would go. For a cubic digraph with n vertices, there are $3n$ edges. So, $(3n)!$ pairings are possible. Since the pairing of outgoing edges with incoming edges is random, it might lead to loops or multiple edges. The graph produced by a random pairing must therefore be checked to see that it does not contain any loops or multiple edges.

Each cell in an array (adjacency matrix) of size $3n$ is initialized to the value of its index. Then it is randomly permuted. That is, the probability of any particular permutation is exactly $1/(3n)!$. The graph is constructed from the resulting array as follows: each vertex i is associated with cells $3i$, $3i+1$ and $3i+2$ as outgoing edges from the vertex i. The values contained in these cells give the vertices into which these edges from i go.

The algorithm is as follows:

1) Initialize the adjacency array.
2) Randomly permute the array to determine the outgoing edges for each vertex.
3) Check if this contains loops or multiple edges. If it does, this trial failed, and the procedure returns to step 1.
4) Return the random simple cubic digraph.

To produce random permutations, the following standard algorithm was used.

```
Procedure RandomPermute(A: Array, m: integer)
Begin
    For i := 0 to m −1 do
    Begin
        r := rand(i, m − 1);
        Swap(A[i], A[r]);
    End;
End;
```

Here, $\text{rand}(x, y)$ returns a random integer in the range $[x, y]$. The whole procedure takes time $\Theta(n)$.

In [3, equations (2.21) and (2.22)] it is established that the occurrences of loops and double edges in the cubic digraphs corresponding to random permutations of the array above are asymptotically independent Poisson distributed with means 3 and 2 respectively. Hence the probability of having no loop and no double edge is asymptotic

to e^{-5}, and the expected number of trials to produce a random cubic digraph is asymptotically e^{5}, which is about 148.413... .

2.2 FINDING HAMILTON CYCLES

As noted at the beginning of the chapter, we will study randomized algorithms. The following section explains the concept behind the design of these algorithms for determining hamiltonicity in cubic digraphs.

2.2.1 IDEA BEHIND THE DESIGN OF RANDOMIZED ALGORITHMS

Given a digraph $G(V,E)$, an undirected graph $G'(V',E')$ can be constructed as follows: Construct the set V' such that for every element v of V, V' contains two elements - v' and v''. Construct the set E' such that for every ordered pair (u,v) which is an element of E, E' contains (the unordered pair) (u',v''). This graph G' has twice as many vertices as the digraph G and the same number of edges as G. Each vertex in G is "split" into two vertices in G' – one corresponding to the edges coming in and another corresponding to the edges going out. That is, vertex v corresponds to v' and v'' and the edges going out of v will go out of v', while the edges coming into v will come into v''. Thus, an edge going from vertex u into vertex v in G will go from vertex u' (the vertex corresponding to the outgoing u) into v'' (the vertex corresponding to the incoming v) in G'. Figure 1 illustrates this. The graph G' thus constructed is an undirected bipartite graph.

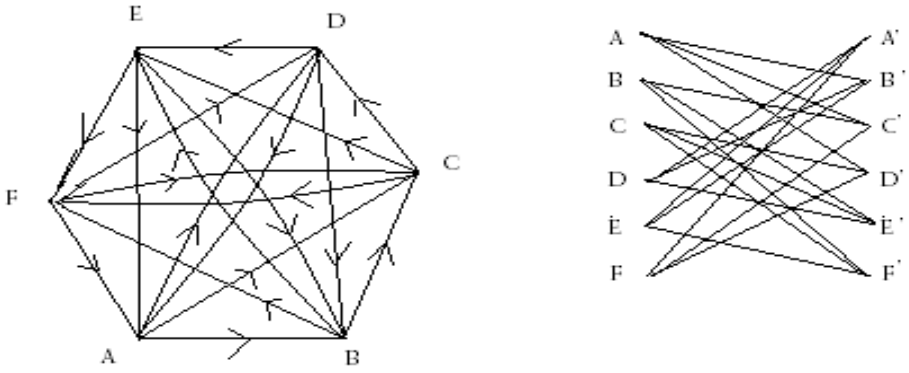


Figure 1 Forming an undirected bipartite graph from a directed graph.

Recall that a matching in the graph G' is a subset of edges $M \subseteq E'$ such that for all vertices $v \in V'$, at most one edge of M is incident to v. A perfect matching is such that each vertex in V has exactly one edge incident with it. Consider a vertex of G – say v. It is associated with two vertices v' and v'' *in* V'. In a perfect matching, one edge is incident with v' and one edge is incident with v''. This corresponds to one edge coming into v and one edge going out of v. Thus, a perfect matching in G' corresponds to choosing for each vertex in G one incoming edge and one outgoing edge – a 1-factor. A Hamilton cycle is a special 1-factor with only one cycle. Since all Hamilton cycles are 1-factors (but not necessarily *vice versa*), some perfect matchings will correspond to Hamilton cycles.

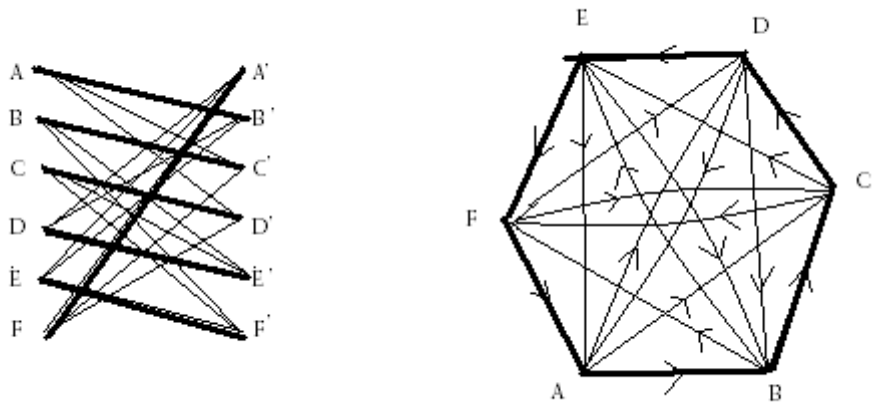


Figure 2 Perfect Matching in G' corresponding to Hamilton cycle in G

König [5] proved that every regular bipartite graph contains a perfect matching (see [7] for a history of the theory of matchings in graphs). Since G' is 3-regular and bipartite, it will always have a perfect matching. Generating a number of perfect matchings and testing if any of them correspond to a Hamilton cycle is the basic idea behind the following algorithms.

2.2.2 RANDOM PERMUTATION METHOD

This algorithm begins with the construction of the undirected graph *G′* associated with a digraph. Then the standard algorithm is used to find a perfect matching for this bipartite graph. This perfect matching is translated as a 1-factor and it is checked whether a Hamilton cycle has been found. If not, a different perfect matching is needed. Given a graph, the algorithm usually finds perfect matchings which are "close" to one another. To avoid this, the graph (labeling) is permuted randomly. This relabeled graph is fed to the standard algorithm, which the finds a new, random perfect matching.

The algorithm is as follows:

1) Given the directed graph G (size n), construct the undirected graph G'.
2) Find a maximum matching in G'.
3) Translate this matching into a 1-factor.
4) Check if this is a Hamilton cycle. If so, return the cycle.
5) If 2-6 (trial) has been done for $100\sqrt{n}$ times, return nonhamiltonian
6) Apply a random permutation to G'. Go to step 2.

Note that this algorithm can err only when the graph is Hamiltonian and the algorithm says that it is not.

The standard algorithm to find a perfect matching in a bipartite graph which contains one is given below. It assumes a matching M which may not be perfect. The terminology for the algorithm is as follows.

Alternating Path: A path such that alternate edges in the path belong to M. (Note: this is different from the alternating path for digraphs).

Augmenting path: An alternating path which has both end vertices unsaturated in M.

Algorithm: (matching M, augmenting path P are sets of edges)

1) Initialize M to $\emptyset$.
2) If M is perfect, return M. If not, choose an unsaturated vertex v and apply breadth first search to find an augmenting path P starting with v.
3) $M :=$ symmetric difference (M, P).
4) Go to step 2.

2.2.3 MARKOV CHAIN APPROACH

A family of Markov chains where each chain is associated with graphs of order *n* is considered to be *rapidly mixing* if it approaches its stationary distribution with uniform relative accuracy of arbitrary $\varepsilon > 0$ in time (number of steps) bounded by some polynomial function of *n* and $\log(1/\varepsilon)$. This approach exploits the fact that rapidly mixing Markov chains can be used to transform one perfect matching into another which is nearly random. Thus, the time required to find a perfect matching becomes a one-time cost. Instead of permuting the graph and finding a perfect matching in the new graph, the original perfect matching is transformed into another perfect matching.

The algorithm is as follows:

1) Given a directed graph *G* of size *n,* construct the undirected graph *G′*.
2) Find a perfect matching for *G′*.
3) Translate this matching into a 1-factor.
4) Check if this is a Hamilton cycle. If so, return the cycle.
5) If 3-6 (trial) has been done for $100\sqrt{n}$ times, return nonhamiltonian
6) Transform matching to get new matching.

Transforming matching:

The analysis in [2] suggests that an approach based on a rapidly mixing Markov Chain can be expected to generate nearly uniform perfect matchings in polynomial time by starting with an original perfect matching and making some changes to it. We used Sinclair's algorithm to transform matchings [8]. If the digraph *G* has *n* vertices, then the bipartite graph *G′* has $2n$ vertices. Let $M_k(G)$ be the set of matchings of size *k* in graph *G* and let the Markov chain $MC_{pm}(G)$ be defined on the state space

$N = M_n(G) \cup M_{n-1}(G)$. Note that N includes auxiliary states in the Markov chain, namely nearly-perfect matchings in G. Transitions in the chain are: in any state $M \in N$, choose an edge $e = (u,v) \in E$ uniformly at random, and then

1) If $M \in M_n(G)$ and $e \in M$, move to state $M' = M - e$.
2) If $M \in M_{n-1}(G)$ and u, v are unsaturated in M, move to $M' = M + e$.
3) if $M \in M_{n-1}(G)$, u is matched to w in M, and v is unmatched in M, move to $M' = (M + e) - (u,w)$.
4) in all other cases, do nothing.

2.3 ANALYSIS AND EXPERIMENTAL RESULTS

Generation of the cubic digraphs takes $O(n)$ time, where n is the number of vertices. However, the graphs should be simple. This may require repeated generation of graphs and testing them (which also takes $O(n)$ time). This would mean that if the algorithm took k trials, i.e. if the first $k-1$ graphs generated had either loops or multiple edges, then the algorithm would take $O(kn)$ time to successfully generate a simple cubic digraph.

Janson has shown [3, special case of theorem 7] that the expected number of Hamilton cycles in a random cubic digraph on n vertices is asymptotic to $2\left(\frac{\pi \cdot e}{3 \cdot n}\right)^{\frac{1}{2}}\left(\frac{4}{3}\right)^n$. Janson [3, special case of theorem 8] then derived the expected number of 1-factors in random cubic digraph on n vertices in a way that implies that the expected proportion of 1-factors which are Hamilton cycles is asymptotic to $\frac{e}{n}$.

The algorithms used find 1-factors and then check to see if a Hamilton cycle is found. So, the expected number of trials to find a Hamilton cycle is proportional to n. Thus randomized algorithms are expected to have a reasonable chance of finding a Hamilton cycle in polynomial time. Our experiments support the hypothesis that random algorithms are well suited for this problem.

The first algorithm for finding a Hamilton cycle (Random Permutation) takes $O(n)$ time for one trial – as one trial consists of finding a perfect matching using standard algorithm for bipartite graphs. Thus, if the algorithm takes k trials to find a Hamilton path, then the algorithm takes $O(kn)$ time to run.

The second algorithm for finding a Hamilton cycle (Markov Chain) takes less time than the first method as it need not generate a perfect matching every time. Hence, the time taken is reduced.

Table 1 shows the experimental results for generating and finding a Hamilton path using both methods for random cubic digraphs. For each vertex size reported, 100 random graphs were generated and tested for hamiltonicity. For each vertex size, the table gives the average values of a) the number of trials needed to generate a simple cubic graph b) the number of trials the random permutation method needed to find a Hamilton path and c) the number of trials the Markov chain method needed to find a Hamilton path.

Table 1 results for random cubic digraphs

Number of vertices	Number of trials for generation	Number of trials for random permutation	Number of trials for Markov chain
10	160.2	6.8	5.3
50	99.9	18	34.3
100	252	108.3	51.7
500	193	98.1	91.4
1000	187.3	453.6	414.2

CHAPTER 3

CAYLEY GRAPHS

Let Γ be a group, and let $S \subseteq \Gamma$ be a generating set. The Cayley digraph associated with (Γ, S) is the directed graph having one vertex associated with each group element and directed edges (g,h) whenever $hg^{-1} \in S$. The edges are labeled by the generators so that (g,h) is labeled by hg^{-1}. The Cayley digraph may depend on the choice of generating set even without edge labeling and is always vertex-transitive. Cayley digraphs where Γ is a permutation group, $|S| = 2$, and exactly one generator is an involution (a permutation that is its own inverse) are considered in this thesis. We call these *restricted Cayley graphs*.

The purpose of the next section is to collect together information about directed Cayley graphs with two generators. In general, we will use X for the set of generators, and Γ for the group generated by X. A directed Cayley graph is specified as $dCay(X,\Gamma)$. The vertex set of this graph is the set of group elements Γ and the edge set consists of all ordered pairs (x, gx) where x is a group element and g is a generator. We consider these graphs as being edge labelled, where the label on an edge is the generator g. In the figures below, edges with different labels are assigned different colors.

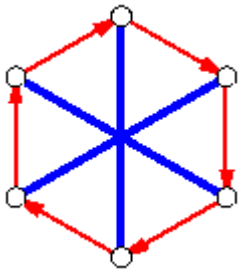
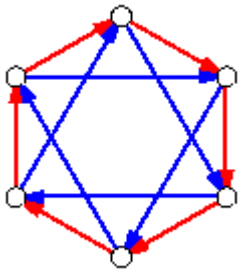
3.1 EXMAPLES OF THE CYALEY GRAPHS CONSIDERED

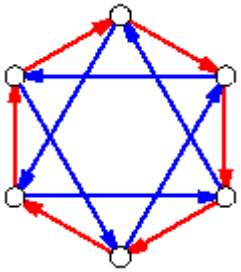
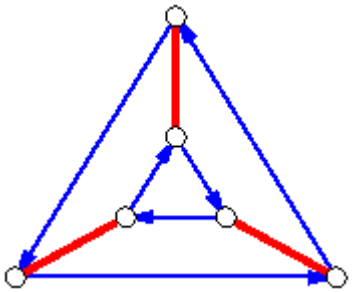
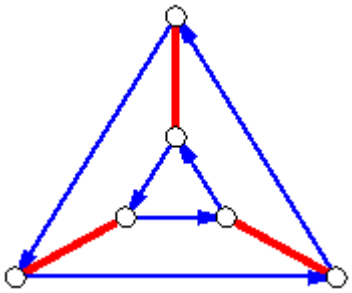
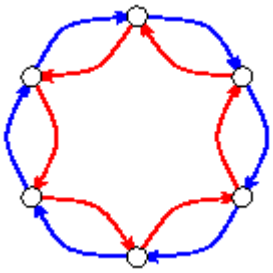
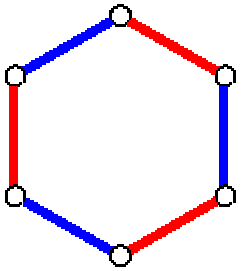
The Cayley graphs used in this study were obtained from Dr. Frank Ruskey's web site [9] and Scott Effler's M.S. Thesis [1], as are the examples explained in this section. Below we show a Cayley graph with generating set {R = (01)(23)(45), B = (02)(14)(35)}.

In a clockwise rotation starting, say, at the top vertex, the successive permutations of *G* are, in one-line notation: 012345, 103254, 351402, 534120, 425031, 240513. In cycle notation the permutations are: *(0)(1)(2)(3)(4)(5)*, (01)(23)(45), (034)(152), (05)(13)(24), (043)(125), (02)(14)(35). Clearly, this graph has only one Hamilton cycle, namely RBRBRB. When a generator is an involution, its edges come in opposing pairs. In the diagrams to follow the two opposing directed edges corresponding to an involution will be drawn as a single thicker undirected edge as shown in the graph on the right.

The table below contains a list of all 2-generated Cayley digraphs on six vertices. The generating sets shown are precisely those that contain two permutations of $\mathbf{S}_6$ with no common fixed points. All Hamilton cycles in the graphs are indicated.

Table 2 Examples of Cayley Graphs

Graph Name	Cayley Graph	Generating Set(s)	Hamilton Cycle(s)
G_1		R = (012345) B = (03)(14)(25)	RRRRRR RBRBRB
G_2		R = (012345) B = (024)(135)	RRRRRR

G_3		R = (012345) B = (042)(153)	RRRRRR BBRBBR
G_4		R = (01)(23)(45) B = (025)(143) R = (23)(45) B = (023)(145)	BBRBBR
G_5		R = (01)(23)(45) B = (024)(135)	Non-Hamiltonian
G_6		R = (012345) B = (543210)	RRRRRR BBBBBB
G_7		R = (01)(23)(45) B = (02)(14)(35) R = (23)(45) B = (02)(14)	RBRBRB

There are several things worth noting from the table above.

- The same Cayley digraph can have more than one set of generators.

- Different Cayley digraphs can have the same underlying digraph; i.e., the last two graphs are identical if we ignore edge colors.

- A Cayley digraph need not be Hamiltonian.

3.2 STRUCTURE AND PROPERTIES OF CAYLEY GRAPH RELEVANT TO HAMILTONICITY

The cayley graphs considered are generated by two generators, called σ and τ (an involution). Since both σ and τ are permutations, they form cycles. A τ cycle has only two edges as τ is an involution (the cycle goes from vertex n to $\tau(n)$ and to $\tau(\tau(n))$ which is n as $\tau\tau$ is the identity.) Also, $\sigma\tau$ is a permutation, which implies that it forms a cycle when repeated. Similarly $\sigma\tau^{-1}$ $\left(\tau = \tau^{-1}\right)$ also forms a cycle when repeated. This is an alternating cycle, as shown in the Figure 3. The red, blue and green cycles correspond to alternating cycles 1,2 and 3 respectively.

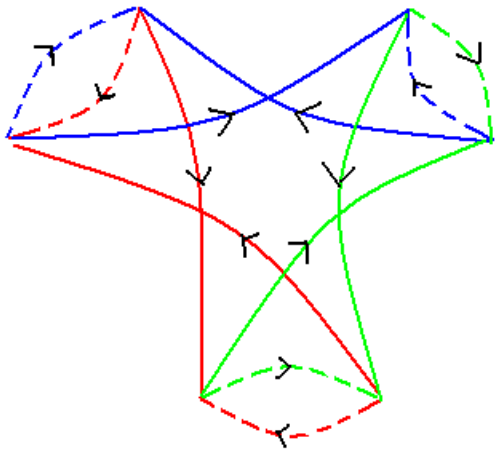


Figure 3 Alternating Cycles

Two alternating cycles are either disjoint or "intersect" at two (or an even number of) vertices. If two alternating cycles intersect at two vertices v_1 and v_2, then one cycle contains the τ edge from v_1 to v_2 and the other cycle contains the τ edge from v_2 to v_1.

These Cayley graphs are diregular graphs of degree 2. An alternating cycle chooses either two incoming vertices or two outgoing vertices for all the vertices present in the cycle. To have a 1-factor, each vertex should have exactly one incoming edge and one outgoing edge. So, in the alternating cycle, if first edge, say coming into vertex v, is chosen (to be in the 1-factor), then the second edge cannot be chosen as it also comes into vertex v. Also, say that the second edge is from vertex u to vertex v. Since it was not chosen, and u has only two outgoing edges, to have a 1-factor, the other edge that is going out of u must be chosen. As the cycle is an alternating one, this is the third edge. Using induction, it can be seen that if a set of edges belongs to the same alternating $\sigma\tau$ cycle, they should have an odd number of edges between them.

Also, if alternate edges from an alternating $\sigma\tau$ cycle are chosen, they have exactly one edge coming in or one edge going out of the vertices concerned. Thus, it can be seen that a set of edges form a 1-factor if and only if they are alternating edges of (all) the alternating $\sigma\tau$ cycles. Choosing different sets of edges (odd numbered or even numbered) from the alternating cycles result in different 1-factors. As the alternating cycles considered are $\sigma\tau$ cycles, either σ edges are chosen or τ edges are chosen from each cycle.

As a Hamilton cycle is a special 1-factor, the problem reduces to choosing the right set of edges from the different alternating cycles. Figure 4 shows a Cayley graphs with its alternating cycles, τ and σ edges demarcated as well as a Hamilton cycle. The

red, blue and green cycles correspond to alternating cycles 1, 2 and 3 respectively. Broken arcs are τ edges. The Hamilton Cycle, shown in black, has τ edges from Alternating cycle 1 and σ edges from alternating cycles 2 and 3.

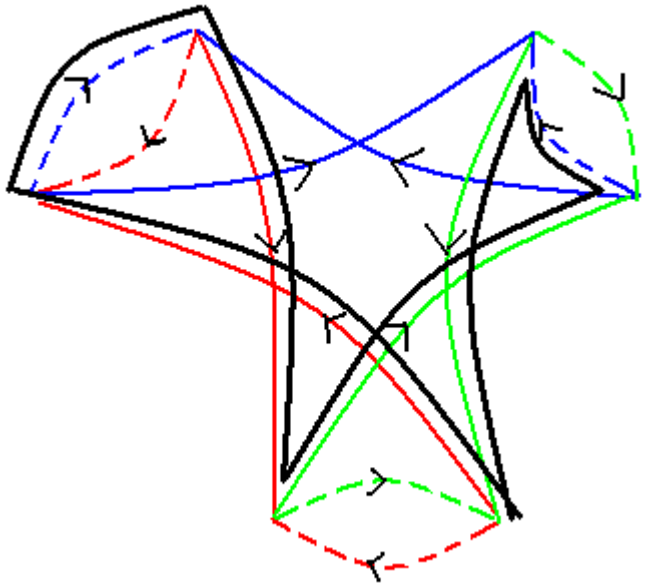


Figure 4 Hamilton Cycle

There are some properties of Cayley graphs which can be exploited to help in choosing the right set of edges – σ or τ - from an alternating cycle.

Two successive τ edges should not be chosen because τ is an involution, so two successive τ edges would from a cycle of length two. This would not be part of a Hamilton path. As mentioned earlier, two alternating cycles intersect at two vertices connected by τ edges. If only one of these edges is to be chosen, then only one of the cycles should choose τ edges and the other cycle should choose σ edges. If a cycle has eight edges in it, it will have four τ edges, so it will touch (a maximum of) four cycles. If

the τ edges from this cycle are chosen, then the four cycles that touch it should choose σ edges. Thus, this restriction propagates as alternating cycles are marked to be σ or τ.

Let ℓ denote the order of σ. Then, if ℓ successive σ edges are chosen, a cycle would result. As this cannot be part of a Hamilton cycle, this should be avoided. In other words, in each σ cycle, only a maximum of $\ell-1$ edges can be chosen. A σ cycle of length ℓ has ℓ edges – these can be part of a maximum of ℓ $\sigma\tau$ cycles. At least one of these $\sigma\tau$ cycles should be chosen to contribute τ edges. (This reasoning can be extended to avoid regular cycles of any nonidentical words over σ and τ).

3.3 RANDOMIZED ALGORITHMS

The properties of the Cayley graphs discussed above relate to Hamiltonicity and directly lead to designing randomized algorithms for finding Hamilton Cycles. The following sections discuss these algorithms.

3.3.1 GENERAL ALGORITHM IDEA

As explained above, the problem of finding 1-factors reduce to choosing either σ or τ edges from each $\sigma\tau$ cycle. As a Hamilton cycle is a special 1-factor, generating 1-factors and checking if it is a Hamilton cycle is a method to get Hamilton cycles. This directly leads to a randomized algorithm which generates some fixed number of one cycles and returns a Hamilton cycle if it finds one or declares the input graph to be non-hamiltonian if a cycle is not found. Note that this algorithm errs only when the given graph is Hamiltonian and the algorithm is not able to find a Hamilton cycle.

If, from every $\sigma\tau$ cycle, either σ edges or τ edges are chosen, then the final selection of edges forming the 1-factor will have say $t\tau$ edges and $s\sigma$ edges. As a whole $\sigma\tau$ should

be marked as σ or τ, the number of these edges i.e. t and s will be a multiple of the length of $\sigma\tau$ cycles, say a. The ratio t/a, say m, is the number of $\sigma\tau$ cycles which contributed τ edges and the ratio s/a, say d, is the number of $\sigma\tau$ cycles which contributed σ edges. Thus, m (or conversely d) is a parameter in marking the $\sigma\tau$ cycles as σ or τ. Note that if the 1- factor obtained turns out to be a Hamilton cycle, then the cycle will contain t τ edges along it.

The algorithm framework (inputs parameter m)

1) input the cayley graph of size n.
2) Form alternating $\sigma\tau$ cycles.
3) Choose m of these cycles to be the ones contributing τ edges to the 1- factor.
4) Mark the rest of the $\sigma\tau$ cycles as contributing σ edges.
5) Thus, selecting alternate edges from $\sigma\tau$ cycles as chosen, from a 1- factor.
6) Check if this 1- factor is Hamilton. If so, return Hamilton cycle.
7) If steps 3-6 (a trial) have not been done some $F(n)$ number of times, go to step 2.
8) Return that the graph is not hamiltonian.

This gives the general outline of the different randomized algorithms tried. The algorithms differed in step 3 – the way they chose m $\sigma\tau$ cycles to be contributing τ edges.

The following sections describe the different ways this choice was made.

3.3.2 RANDOM SELECTION

This is the simplest algorithm. It simply chooses m out of the total number of $\sigma\tau$ cycles at random without replacement. The rest are marked to be cycles contributing σ edges. This does not utilize any of the properties of the Cayley graphs which help in

making this choice. There are $\binom{k}{m}$ permutations of choices if there are k $\sigma\tau$ cycles. This will always be successful in choosing m cycles to be τ and a 1- factor is guaranteed.

3.3.3 τ RESTRICTIONS

This algorithm uses the fact that τ is an involution and that two successive τ edges will result in a cycle. To avoid this, the algorithm will prohibit choosing two adjacent τ edges. As explained previously, when two $\sigma\tau$ cycles intersect, they do so at two adjacent vertices which are connected by τ edges. So, if (and only if) two intersecting $\sigma\tau$ cycles are chosen to be contributing τ edges, the 1- factor will contain a completed τ cycle. To avoid this, if a $\sigma\tau$ cycle is marked as τ, all the cycles that intersect it should be marked as σ. This is the basic idea of this algorithm.

Algorithm: (step 3 of the generalized algorithm)

1) For each $\sigma\tau$ cycle, find the other $\sigma\tau$ cycles it touches.
2) While the number of $\sigma\tau$ marked as τ is less than m
    a. Choose an as yet unmarked $\sigma\tau$ cycle at random. If this is not possible, fail.
    b. Mark this to be τ.
    c. Mark all the $\sigma\tau$ cycles this touches to be σ.

Let us suppose that there are k $\sigma\tau$ cycles, and each $\sigma\tau$ cycle intersects g other $\sigma\tau$ cycles. Then choosing one to be τ will mark g cycles to be σ. So, in the first step, there are k cycles to choose from. In the next step, there are only $k-g-1$ cycles to choose from. If this continues, at step i, there would be $k-i*(g-1)$ cycles to choose

from. This is quite a strong restriction as a $\sigma\tau$ cycle typically touches at least three or four other cycles.

However, the analysis done above is optimistic because two $\sigma\tau$ may touch at more than once. Also, if cycle 1 touches cycle 2 and cycle 2 touches cycle 3, and cycle 1 is marked as τ forcing cycle 2 to be σ, choosing cycle 3 to be σ does not have any effect on cycle 2 – the restriction is not propagated as it is already marked as σ.

Note that this procedure may fail to mark m $\sigma\tau$ cycles as τ. If the initial choice of $\sigma\tau$ cycles to be marked as τ is unlucky, it may be the case that for some m's, it may not be possible to mark $\sigma\tau$ cycles as τ. In this case the step 3 of the generalized algorithm would fail, meaning that a 1- factor was not formed. Then, the whole trial fails, and the generalized algorithm retries from scratch by calling this procedure again.

3.3.4 σ RESTRICTIONS

This algorithm uses the fact that σ is a permutation and that ℓ successive σ edged will form a σ cycle – which will typically not be a Hamilton cycle. So, it avoids choosing ℓ successive σ edges. A single σ cycle can be a part of to a maximum of ℓ $\sigma\tau$ cycles. Only if all of these are marked as σ, will the σ cycle will be completed. To avoid this, at least one of the $\sigma\tau$ cycles which touch a single σ cycle should be marked as τ.

Algorithm (step 3 of the generalized algorithm)

1) Form σ cycles. Let there be r such cycles, each of length l.
2) For all the σ cycles, find the $\sigma\tau$ cycles which include the same edges.
3) For each $\sigma\tau$ cycle, find the other $\sigma\tau$ cycles it touches.
4) For each σ cycle do
    a. Choose at random a $\sigma\tau$ cycle that touches it which is not marked.

b. If this is not possible fail.

c. Mark it as τ.

d. Mark all the $\sigma\tau$ cycles which touch this as σ.

5) While the number of $\sigma\tau$ cycles marked as τ is less than m (will repeat for a max of $m - r$ times)

a. Find an as yet unmarked $\sigma\tau$ cycle at random

b. If this is not possible, then fail

c. Mark it as τ.

d. Mark all the $\sigma\tau$ cycles touching it as σ.

Note that this algorithm may also fail, not being able to assign m $\sigma\tau$ cycles as τ. This would fail step 3 in the generalized algorithm, which would fail the trial. Also, this poses a restriction on the numerical value of m. If there are r σ cycles, there have to be at least r $\sigma\tau$ cycles chosen as τ.

3.4 ANALYSIS AND EXPERIMENTAL RESULTS

These randomized algorithms need two parameters – how many trials to make as a function of the size of the graph, and the value of m. Number of trials is fixed at $100n$. The m value is input at the beginning of the algorithm. For a single Cayley graph, the algorithm is run several times with m values beginning from r (number of s cycles) and increasing until a Hamilton cycle is found.

The first algorithm tried was totally random. The proportion of random digraphs of degree 2 which are Hamiltonian is very small. However, Cayley graphs are very structured and a lot of them contain Hamilton cycles. Hence, if the algorithm that is

used to find this does not exploit the structure of the Cayley graphs, it is not expected to perform very well. This was proved experimentally. The totally random algorithm had to choose from an exponential number of choices. It was able to find Hamilton cycles in only relatively small graphs – till 60 vertices.

The second algorithm used the τ restrictions. This proved to be a very useful restriction as the number and size of graphs for which Hamilton cycles could be found increased considerably. Graphs with over 2000 vertices could be proved to be Hamiltonian with this. The number of trials was not obviously related to the number of vertices. However, the *m* value was important in finding a Hamilton cycle. For some critical optimal value of *m*, the algorithm is able to find a Hamilton cycle fairly easily. For lower m values, the algorithm is not able to find a Hamilton cycle even when it has finished all 100*n* trials. For larger values of *m*, the algorithm fails to find *m* τ cycles. The algorithm was coded so that if it is not able to find *m* τ cycles, the rest were assumed to be σ. For values of *m* higher than the correct critical value, but close enough, this found some Hamilton cycles.

The third algorithm used the σ restrictions. This did not prove to be as powerful a restriction. Graphs with up to 2000 vertices were proved to be Hamiltonian using this restriction. This interacts with the τ restriction and the combination reduces the choices that are available for the randomized algorithm. This improves the algorithm's performance. However, the same $\sigma\tau$ cycle touches many σ cycles. This would mean that marking one of these $\sigma\tau$ cycles as τ would satisfy the restriction for many σ cycles. Once again, the value of *m* proved critical, in a way similar to the previous algorithm.

Although a random 2 regular digraph does not contain a lot of Hamilton cycles, Cayley graphs are very structured and most of them are Hamiltonian. It is reasonable to expect that randomized algorithms using the structure of Cayley graphs will have a good chance of finding Hamilton cycles. While this has proved to be true, *i.e.* increasing the restrictions which exploit more and more of the structure of Cayley graphs result in an increase in performance, most of the inherent structure was not captured by these algorithms. They were unable to find Hamilton cycles in graphs with a very large number of vertices, even though they were Hamiltonian.

The results are presented in Table 2. In the table, *m* denotes the number of τ cycles along the Hamilton path, *rnd* denotes the number of trials the random algorithm took, τ denotes the number of trials τ restrictions took and σ denotes the number of trials σ restrictions took. For each vertex size reported, the website presented many non-isomorphic graphs. Each algorithm was tested on all these graphs. However, as the number of trials needed was extremely graph dependent, the values were not averaged. Instead, the values for the first graph presented are reported. When Hamilton cycles could not be found for some graphs while some others of the same vertex size were found to be Hamiltonian, the table reports the failure as "xxx", except for the vertex size of 2880. There is a graph of vertex size 2880 for which hamiltonicity is unknown. Though our algorithms do not find this, they were able to solve anther graph of the same size. So, this is reported. For each vertex size, the table gives the values of a) the number of trials the random algorithm needed to find a Hamilton path; b) the number of trials τ restrictions needed to find a Hamilton path; c) the number of trials σ restrictions needed to find a Hamilton path.

Table 2 results for restricted Cayley digraphs

number of vertices	m	rnd	τ	σ
6	1	1	1	1
8	1	1	1	1
10	1	1	1	2
12	1	1	1	1
14	1	1	1	2
16	1	1	1	2
20	1	1	1	2
24	1	1	1	2
28	1	1	1	2
30	1	1	3	1
32	1	1	1	2
36	1	1	1	2
40	1	1	1	2
42	3	xxx	4	3
54	1	1	1	2
60	1	1	1	2
64	3	xxx	1	2
72	5	xxx	1	2
84	6	xxx	44	40
96	6	xxx	44	40
108	5	xxx	1	3
144	9	xxx	1	3
162	4	xxx	2	2
168	11	xxx	3	2
192	7	xxx	9	14
216	11	xxx	14	14
240	11	xxx	14	14
288	11	xxx	2873	2066
324	13	xxx	84	110
336	11	xxx	615	1526
360	27	xxx	1331	1622
384	13	xxx	14317	4655
432	17	xxx	915	1370
480	31	xxx	9630	3675
576	25	xxx	1595	251
648	33	xxx	1584	15842
720	xxx	xxx	xxx	xxx
1080	25	xxx	770	46613
1152	xxx	xxx	xxx	xxx
1296	61	xxx	185	185
1344	xxx	xxx	xxx	xxx
1440	61	xxx	18175	18175
1512	xxx	xxx	xxx	xxx
2160	xxx	xxx	xxx	xxx
2520	xxx	xxx	xxx	xxx
2880	26	xxx	144001	144001

References

[1] Effler, S., "Enumeration, Isomorphism and Hamiltonicity of Cayley Graphs: 2-Generated and Cubic" M.S. Thesis, University of Victoria, (2002).

[2] Frieze, A., Jerrum, M. R., Molloy, M. S. O., Robinson, R. W. and Wormald, N. C., Generating and counting Hamilton cycles in random regular graphs, *J. Algorithms* **21**, 176-198, (1996).

[3] Garey, M. R., Johnson, D. S. and Tarjan, R. E., The planar Hamiltonian circuit problem is NP-Complete, *SIAM J. Comput.* **5**, 704-714, (1976).

[4] Janson, S., Random regular graphs: Asymptotic distributions and contiguity*, Combin. Probab. Comput*. **4**, 369-405, (1995)

[5] König, D., Über Graphen und ihre Anwendung auf Determinanthen theorie und Mengenlehre, *Math. Annalen* **77**, 453-465, (1916).

[6] Mae, X., "Finding Hamilton Cycles in Cubic Graphs", M.S. Thesis, University of Georgia, (2001).

[7] Plummer, M. D,, Matching theory - a sampler: from Dénes König to the present, *Discrete Math*. **100**, 177-219, (1992).

[8] Sinclair, A., "Algorithms for Random Generation and Counting: A Markov Chain Approach", Birkhäuser, Boston, (1993).

[9] Ruskey, F., Hamiltonicity of 2-generated Cayley Graphs, http://theory.cs.uvic.ca/~cos/cayley/#hamiltonicity.