Various One-Factorizations of Complete Graphs

A. Prasant Gopal Kishore Kothapalli V. Ch. Venkaiah
Center for Security, Theory, and Algorithmic Research
International Institute of Information Technology
Gachibowli, Hyderabad - 500 032.
India.
Email:{kkishore, venkaiah}@iiit.ac.in, prasant_a@students.iiit.ac.in

C. R. Subramanian
The Institute of Mathematical Sciences
Chennai - 600 113
India.
Email: crs@imsc.res.in

April 2, 2007

Abstract

Methods to compute 1–factorizations of a complete graphs of even order are presented. For complete graphs where the number of vertices is a power of 2, we propose several new methods to construct 1–factorizations. Our methods are different from methods that make use of algebraic concepts such as Steiner triple systems, starters and all other existing methods. We also show that certain complete multipartite graphs have 1–factorizations by presenting a method to compute 1–factorizations of such graphs. This method can be applied to obtain 1–factorizations of complete graphs with the number of vertices being a multiple of 4 or complete graphs with mn vertices provided a 1–factorization of K_m and a 1–factorization of K_n are known.

Finally, deterministic and randomized back-tracking based algorithms to produce a 1–factorization for K_{2n} are presented. Both the algorithms always produce a 1–factorization if one exists.

1 Introduction

A one factor of a graph G is a regular spanning sub-graph of degree one. In other words, a one factor is a set of pairwise disjoint edges of G that between them contain every vertex. A one factorization of G is a partition of the edge set of G into edge disjoint one factors. For a graph to possess a one factorization, an obvious necessary condition is that the graph must have an even number of vertices. Another necessary condition for a graph to have a one factorization is that it must be regular. However, a regular graph with a bridge can not have a one factorization. There are also bridge-less regular graphs that do not have one factorization e.g., complete graphs of odd order [1]. It has been conjectured that a regular graph with $2n$ vertices and degree greater than n will always have a one factorization. Also, for many classes of graphs including the class of complete graphs the existence of one factorizations can be proved. There have also been many works that show how to construct 1-factorizations of complete graphs on an even number of vertices, see for example [17, 11] and the references therein. In this paper, we use standard notation from graph theory followed by most books, e.g., West[19].

An immediate application of 1-factorizations is that of edge colouring. A 1-factorization of a given graph G partitions the edge set into classes so that each class can be coloured with the same colour. For $G = K_{2n}$ this can be readily seen to produce a valid $2n-1$-edge colouring as a 1-factorization of K_{2n} consists of $2n-1$ factors. In this case, since $\Delta(K_{2n}) = 2n-1$, this is also the best possible for complete graphs. Similar result holds for also complete bipartite graphs.

The study of 1-factorizations is motivated by other combinatorial applications such as scheduling tournaments [17], especially round-robin tournaments. Here, the schedule of games played at the same time can be seen to form a 1-factor of the underlying complete graph. Several variations of tournaments such as ideal tournaments [17, 18, 3] and competition schedules can also be reduced to that of 1-factorizations in graphs. Other applications of 1-factorizations include block designs, 3-designs, and Room squares and Steiner systems [17, 13].

There are other related notions of 1-factorizations namely sequentially uniform, uniform, and perfect 1-factorizations. A 1-factorization $F = \{F_1, F_2, \cdots, F_{2n-1}\}$ of K_{2n} is said to be uniform if the union of any two distinct 1-factors $F_i, F_j, i \neq j$, is isomorphic to the same graph. The 1-factorization F is said to be sequentially uniform if the above property holds for any two consecutive (modulo-$2n-1$) 1-factors. Since the union of any two 1-factors is a 2-edge colourable 2-regular graph, it is isomorphic to a disjoint union of cycles which can be succinctly represented as follows. The multi-set $C = (c_1, c_2, \cdots, c_k)$, with $\sum_{i=1}^{k} c_i = 2n$, is called the *type* of a sequentially uniform 1-factorization if $F_i \cup F_{i+1}$ is isomorphic to the disjoint union of cycles of length $c_1, c_2, \cdots, c_k$. A 1-factorization is said to be perfect if the union of any two 1-factors is isomorphic to a Hamiltonian cycle. Perfect 1-factorizations find applications in several combinatorial problems such as acyclic edge colouring [1, 16, 12] and constructing short length erasure codes [4]. Perfect 1-factorizations are known to exist for very few classes of graphs, for example K_n where n is a prime or K_{2n} when $2n-1$ is a prime. It is however conjectured that every complete graph on an even number of vertices has a perfect 1-factorization and the sizes of graphs for which this is known to be true are $2n = 16, 28, 36, 40, 50, 126, 170, 244, 344, 730, 1332, 1370, 1850, 2198, 3126, 6860, 12168,$ 16808, and 29792 [2].

There has been a lot of work in devising methods to arrive at 1-factorizations of complete graphs using algebraic and analytical techniques alike. Given the huge number of possible 1-factorizations, several questions such as those listed below are still open.

- What other methods exist to construct 1-factorizations?
- How to produce a random 1-factorization?
- What are other classes of graphs for which one-factorizations can be shown to exist and constructed?

[1] However, complete graphs of odd order are known to have what is called a near 1-factorization.

In this paper we answer the above three questions by producing few methods of arriving at 1-factorizations of K_{2n}, presenting an algorithm to construct a random 1-factorization, and showing that certain complete multipartite graphs also have (perfect) 1-factorizations. A one-factorization of complete multipartite graphs yields another way to construct 1-factorizations of complete graphs. Apart from the above, we also present a deterministic algorithm to construct a 1-factorization of complete graphs of even order.

1.1 Related Work

There exist many different one factorizations of K_{2n} [18, 17]. One of the one factorizations of K_{2n} is a patterned factorization, GK_{2n} obtained from the patterned starter or the staircase method of Bileski [7]. The factorization G_{2n} is a uniform factorization for all $n \geq 1$ and when $2n-1$ is prime then it is a perfect 1-factorization as well. Another one factorization of K_{2n} that is not isomorphic to GK_{2n} for $n \geq 4$ is WK_{2n} [10]. This is obtained by a family of starters different from patterned starters. Another method to give one factorization is by viewing K_{2n} as the union of three graphs: two disjoint copies of K_n and a copy of $K_{n,n}$. Factorizations obtained using this method are called twin factorizations, GA_{2n} [18].

Various one factorizations have been constructed from Steiner triple systems [5]. Steiner triple systems have also found application in constructing sequentially uniform one-factorizations [5]. Binary projective Steiner triple systems can be used to construct uniform 1-factorizations of K_{2n} of type $[4\ 4\ 4\ \cdots 4]$ if n is a power of 2 [11]. Perfect Steiner triple systems which give rise to uniform one factorizations of type $[2n-4\ 4]$ are also known [8]. Uniform one factorizations of type $[4\ 6\ 6\ \cdots\ 6]$ exist and can be constructed from Hall triple systems if n is a power of 3. When p is an odd prime there is a one factorization of K_{p^s+1} of type $\{p+1\ 2p\ 2p\ \cdots\ 2p\}$ [11].

There has been some work on classifying 1-factorizations according to *isomorphism*. Two one-factorizations are called isomorphic if there exists a bijection that maps one-factors onto one-factors. Recently, Kaski and Ostergard provided a classification of 1-factorizations of regular graphs on 12 vertices [9] extending the results known for graphs on at most 10 vertices [15, 14].

Dinitz and Stinson [6] designed a hill climbing algorithm to produce a random one factorization of K_{2n}, for a particular value of $2n$. Their algorithm has the disadvantage that it reaches a local optimum and cannot proceed further towards the global optimum. However, as the authors note in [6], in over a million trials it never happened but it is not proven that such a situation never occurs.

1.2 Our Results

This paper presents several recursive methods to obtain a 1-factorization of a complete graph where the number of vertices is a power of 2. The resulting 1-factorizations and our methods are different from existing methods.

Another problem we turn our attention to is that of producing a random 1-factorization. We propose a randomized algorithm to iteratively construct a random 1-factorization of a given graph. Our algorithm relies on backtracking and is guaranteed to stop by producing a 1-factorization if one exists or report failure otherwise by exploring the space of 1-factorizations systematically to produce the output. We also implemented our algorithm and tabulated the results of our experiments for inputs being complete graphs on an even number of vertices. The results are shown in Section 3. A deterministic variant that uses backtracking is also studied and implemented.

We then describe a method to construct a 1-factorization of complete multi-partite graphs. Later, using this, we show to how to arrive at a 1-factorization of a complete graph on mn vertices, K_{mn}, provided the 1-factorizations of K_m and K_n. Thus, our method allows one to construct a 1-factorization of a complete graph where the number of vertices is a multiple of 4 easily.

1.3 Organization of the paper

The rest of the paper is organized as follows. Section 2 presents methods to construct 1-factorizations of K_n where n is a power of 2. In Section 3 we present our incremental algorithms to construct 1-factorizations for K_{2n} and also show some implementation results. In Section 4 we show how to construct a 1-factorization for complete multipartite graphs and use it to provide a 1-factorization for K_n when n is a multiple of 4. The paper ends with some concluding remarks.

2 1-Factorization of K_{2^r}, $r > 1$

In this section we report some polynomial time approaches to arrive at a 1-factorization of the complete graph on n vertices where n is a power of 2. Our interest in exploring the space of 1-factorizations for K_{2^r} is to investigate non-algebraic ways of constructing them unlike [5] where Steiner systems were used to arrive at uniform 1-factorizations of K_{2^r}. Moreover, our initial experiments have suggested that when using a simple backtracking strategy presented in Section 3.1 to arrive at 1-factorizations for K_{2n}, no backtracking is required when n is a power of 2. This led us to study the reason behind this.

The complete graph on $2n$ vertices can in general be represented as the union of 3 graphs: two K_n's that are indexed by vertices 1 through n and $n+1$ through $2n$ respectively, and a complete bipartite graph $K_{n,n}$ with n vertices in each side of the partition. This representation allows us to represent the 1-factorization of K_{2n} for n being a power of 2 as combining the 1-factorization of two K_ns along with a 1-factorization for $K_{n,n}$. The 1-Factorizations obtained this way are termed as *twin factorizations* in [17] where also a method to construct them is described. Here we report several different methods that can be used to construct twin factorizations recursively. We can then use this approach to build the 1-factorization of K_n. The recursion stops when $n = 2$ where the 1-factorization is simply the single edge in K_2.

Let $2n$ be a power of 2. The 1-factorization of K_n consists of $n-1$ 1-factors each containing $n/2$ edges. Thus, putting the 1-factorization of both the K_ns together will result in $(n-1)$ 1-factors each containing n edges. The 1-factorization of $K_{n,n}$ will have n 1-factors each containing n edges. Thus, totally we have $n-1+n = 2n-1$ 1-factors containing a total of $n(2n-1)$ edges, which is the number of edges in K_{2n}. Since each edge of K_{2n} belongs to exactly one 1-factor, this method results in a 1-factorization of K_{2n}.

What is left unspecified is how to generate the 1-factorization of $K_{n,n}$. One standard 1-factorization of $K_{n,n}$ is described in [18]. We have found several strategies to arrive at the 1-factorization of $K_{n,n}$ and we call the one reported in [18] as the shift-and-rotate (S-R) strategy. We describe other strategies called the butterfly strategy and the class of shift-rotate strategies below. We also include an example to describe the approaches. Here we mention that the 1-factorizations we obtain for K_{2^r} are different from $\mathcal{GA}_{2n}$ and $\mathcal{GK}_{2n}$ reported in [18], except the one in Section 2.1.

2.1 The Shift-And-Rotate Strategy (S-R)

In the S-R strategy, to build a 1-factorization for $K_{n,n}$, we note that each 1-factor has $1, 2, \cdots, n$ as the first end-points of the n edges in order and $n+1$ through $2n$ as the second end-points. The second end-points are paired up differently in each 1-factor starting with the 1-factor $(1, n+1), (2, n+2), \cdots, (n, 2n)$. To build the ith 1-factor, shift-and-rotate the sequence $n+1, n+2, \cdots, 2n$ by i places to the left giving rise to the 1-factor $(1, n+1+i), (2, n+i+2), \cdots, (n, 2n+i-n)$.

To show that the resulting 1-factors form a 1-factorization, we can formally argue as follows. In the 1-factors corresponding to the edges of $K_{n,n}$, each edge of the $K_{n,n}$ appears exactly in one 1-factor. Given an edge (i, j) with $1 \leq i \leq n$ and $n+1 \leq j \leq 2n$, in our ordering of 1-factors the edge (i, j) appears in the $j-i+1$th 1-factor as the end-point of i.

2.2 The Butterfly Strategy

The butterfly strategy is another strategy to build a 1-factorization for $K_{n,n}$. By convention we list each 1-factor as n edges where the first end-points are from 1 through n. So a typical 1-factor looks as $(1, v_1), (2, v_2), \cdots, (n, v_n)$ where $n+1 \le v_1, v_2, \cdots, v_n \le 2n$. For $\ell = 0$ the 1-factor F_0, which we call the *identity factor* is simply $\{(1,n), (2,n+1), (3,n+2), \cdots, (n,2n)\}$. For $\ell = 1, 2, \cdots, n-1$, the ℓth 1-factor F_i is computed as follows. If ℓ is a power of 2 then we compute F_ℓ as follows. The edge $(i,j) \in F_\ell$ if the edge $(i',j') \in F_0$ such that $i' = (i+\ell) \bmod n$ and $j' = j$. Otherwise, let $\ell' = 2^{\lfloor \log_2 \ell \rfloor}$ and $r = \ell - \ell'$. Let $F_{\ell'}$ be the ℓ'th factor. Now $(i,j) \in F_\ell$ if $(i',j') \in F_{\ell'}$ such that $i' = (i+r) \bmod n$ and $j' = j$.

Using this recursively, a 1-factorization for K_{16} is shown in the example below.

Example 2.1 We demonstrate the butterfly strategy by building a 1-factorization for K_{16}. We first list the 1-factors corresponding to the cross edges of K_{16} by starting with the *identity* factor $F_0 = \{(1,9), (2,10)$,(3,11) , $(4,12), (5,13), (6,14), (7,15), (8,16)\}$.

For K_{16}, using the butterfly strategy for each of the 1-factors coming from the cross edges, the first end-point is from the set $\{1,2,3,4,5,6,7,8\}$. The 1-factors F_0 and F_1 are shown in the Figure 1.

1,10	2,9	3,12	4,11	5,14	6,13	7,16	8,15

Figure 1: The 1-factors F_0 and F_1.

The 1-factors F_2 and F_3 are shown in the figure below.

1,11	2,12	3,9	4,10	5,15	6,16	7,13	8,14
1,12	2,11	3,10	4,9	5,16	6,15	7,14	8,13

Figure 2: The factors F_2 and F_3.

We now show the four 1-factors corresponding to F_4 through F_7. Putting together the 1-factorization of K_8 and another K_8 with vertices numbered 1 through 8 and 9 through 16 recursively, we arrive at a 1-factorization of K_{16} as shown in Figure 4.

Remark 2.2 *It can be observed that using the MIN strategy, explained in Section 3, one would have arrived at the same 1-factorization that will be obtained using the butterfly strategy. This is the reason why for an input of K_{2n} with $2n$ a power of 2, the MIN algorithm requires no backtracking.*

Remark 2.3 *Notice that the 1-factorization for $K_{n,n}$ we have described above are also perfect 1-factorizations for $K_{n,n}$. The 1-factorization for K_{2n}, n being a power of 2, is also uniform of the type* $[4\ 8\ 16\ \cdots\ 2n]$, *i.e., there exists cycles of all powers of 2 till $2n$.*

2.3 Other Methods to Compute 1-Factorizations of $K_{2^r}, r > 1$:

Apart from the above two, there exist other approaches to generate perfect 1-factorizations for $K_{n,n}$ and hence K_{2n} where n is a power of 2. A simple observation is that to construct a 1-factorization of $K_{n,n}$, we can start with any of the $n!$ 1-factors as F_1 and employ shift-and-rotate strategies described as follows. Consider the 1-factor F_1 and let us represent each edge in F_1 as $\{(u_i, v_i)\}_{i=1}^n$ where all u_is are in the

1,13	2,14	3,15	4,16	5,9	6,10	7,11	8,12
1,14	2,13	3.16	4,15	5,10	6,9	7,12	8,11
1,15	2,16	3,13	4,14	5,11	6,12	7,9	8,10
1,16	2,15	3,14	4,13	5,12	6,11	7,10	8,9

Figure 3: The factors F_4 through F_7.

1,2	3,4	5,6	7,8	9,10	11,12	13,14	15,16
1,3	2,4	5,7	6,8	9,11	10,12	13,15	14,16
1,4	2,3	5,8	6,7	9,12	10,11	13,16	14,15
1,5	2,6	3,7	4,8	9,13	10,14	11,15	12,16
1,6	2,5	3,8	4,7	9,14	10,13	11,16	12,15
1,7	2,8	3,5	4,6	9,15	10,16	11,13	12,14
1,8	2,7	3,6	4,5	9,16	10,15	11,14	12,13
1,9	2,10	3,11	4,12	5,13	6,14	7,15	8,16
1,10	2,9	3,12	4,11	5,14	6,13	7,16	8,15
1,11	2,12	3,9	4,10	5,15	6,16	7,13	8,14
1,12	2,11	3,10	4,9	5,16	6,15	7,14	8,13
1,13	2,14	3,15	4,16	5,9	6,10	7,11	8,12
1,14	2,13	3,16	4,15	5,10	6,9	7,12	8,11
1,15	2,16	3,13	4,14	5,11	6,12	7,9	8,10
1,16	2,15	3,14	4,13	5,12	6,11	7,10	8,9

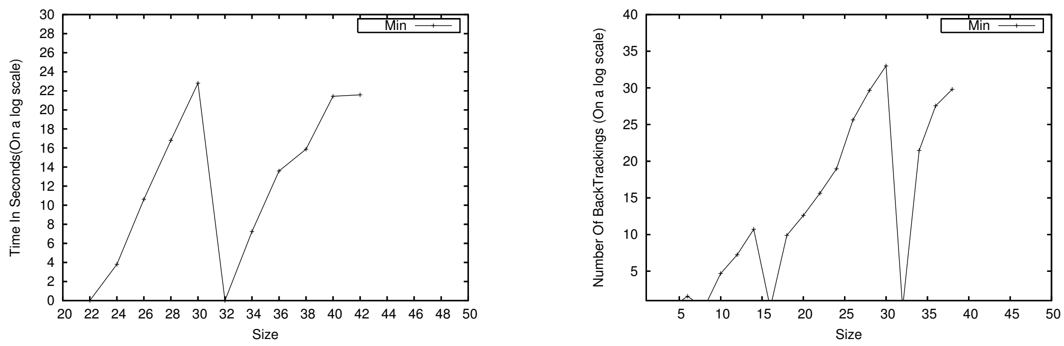
Figure 4: The 1-factorization of K_{16} obtained using the butterfly strategy.

same partition and the v_is belong to another partition. Let us call the u_is as the first end-points and v_is as the second end-points. When using shift-and-rotate strategies, we keep either u_is or v_is fixed throughout the other 1-factors. The other set of end-points, say $\{u_1, u_2, \cdots, u_n\}$ are shifted with rotation to get the other 1-factors. The effect of the shift operation on $\{u_1, u_2, \cdots, u_n\}$ is $\{u_2, u_3, \cdots, u_n, u_1\}$ when we shift to the left (with rotation). In the class of shift-and-rotate strategies, we can shift the first end-points or the second end-points of the edges in F_1 to obtain F_2 through F_{n-1}.

An example of a 1-factorization obtained using the shift-and-rotate class is as follows for $n = 16$. The first end-points of the 8 factors of $K_{8,8}$ are 1 through 8 and the second end-points are 16 down to 9. Thus in the 1-factorization of $K_{n,n}$ we construct, $F_1 = \{(1,16), (2,15), \cdots, (8,9)\}$. Now, shift and rotate the first end-points to the left to get the remaining $n-1$ factors. Using this recursively, we get the following 1-factorization for K_{16}.

1,8	2,7	3,6	4,5	9,16	10,15	11,14	12,13
2,8	3,7	4,6	1,5	10,16	11,15	12,14	9,13
3,8	4,7	1,6	2,5	11,16	12,15	9,14	10,13
4,8	1,7	2,6	3,5	12,16	9,15	10,14	11,13
1,4	2,3	5,8	6,7	9,12	10,11	13,16	14,15
2,4	1,3	6,8	5,7	10,12	9,11	14,16	13,15
1,2	3,4	5,6	7,8	9,10	11,12	13,14	15,16
1,16	2,15	3,14	4,13	5,12	6,11	7,10	8,9
2,16	3,15	4,14	5,13	6,12	7,11	8,10	1,9
3,16	4,15	5,14	6,13	7,12	8,11	1,10	2,9
4,16	5,15	6,14	7,13	8,12	1,11	2,10	3,9
5,16	6,15	7,14	8,13	1,12	2,11	3,10	4,9
6,16	7,15	8,14	1,13	2,12	3,11	4,10	5,9
7,16	8,15	1,14	2,13	3,12	4,11	5,10	6,9
8,16	1,15	2,14	3,13	4,12	5,11	6,10	7,9

Figure 5: A 1-factorization of K_{16} obtained from the class of shift-rotate strategies.



(a) (b)

Figure 6: Figure (a) shows the time taken and (b) shows the number of backtrackings required by the MIN algorithm.

3 Experimental Results

In this section, we consider incremental methods of constructing an one-factorization of K_{2n} using some heuristics. It can be easily seen and also verified that a one-factorization F of K_{2n} has $2n-1$ perfect matchings in it. Each perfect matching has n edges in it. Let F be = $\{F_1, F_2, F_3 \ldots, F_{2n-2}, F_{2n-1}\}$. Note that, $\{F_i \cap F_j\} = \phi \; \forall \, i, j \in [1, 2n-1]$ if $i \neq j$ and $|F_1 \cup F_2 \cup F_3 \ldots \cup F_{2n-1}| = (2n-1) \cdot (n)$. In this section, we not only build F incrementally but also in a lexicographic order. To extend a matching, we need to add an edge that is not included in any of the previous perfect matchings of F. For picking up this edge, we do not go totally random. A matching F_i always has $(1, i)$ as its first edge in it and is then incrementally extended so that F_i becomes a perfect matching. At any step if we are constructing a j^{th} edge, (u_j, v_j), for the F_i^{th} matching, we may have the choice of adding edges only from the set $E \setminus \{F_1 \cup F_2 \ldots \cup F_{i-1} \cup E_i\}$, where $E_i = \{e_{i_i}, e_{i_2}, e_{i_3}, \ldots e_{i_{j-1}}\}$ and e_{i_k} is the k^{th} edge chosen for the F_i^{th} matching. For convenience, we choose our first vertex u_j as the minimum [2] of all the available vertices $V' = \{v_1, v_2, v_3 \ldots v_i\}$ where $v_i \in V'$ *if and only if* it is not saturated by F_i. The second vertex v_j can be chosen using the *MIN and RAND* methods. So, any $F_i = \{(1, i), (u_1, v_1), (u_2, v_2), (u_3, v_3), \ldots\ldots (u_{n-2}, v_{n-2}), (u_{n-1}, v_{n-1})\}$ where, u_i = minimum of all vertices unsaturated by F_i and v_i = vertex chosen using MIN or RAND method.

3.1 The MIN Method

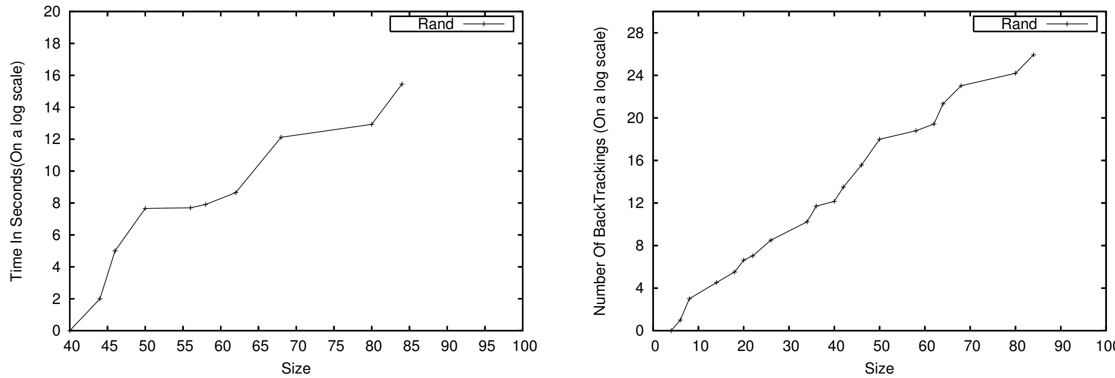
In this method, to choose a j^{th} edge, (u_j, v_j), for the F_i^{th} matching, the minimum of all the unsaturated vertices by F_i, is chosen as its first vertex u_j. To pick up the second vertex, we again follow the same strategy used for picking up the first one. If the edge (u_j, v_j) is already used in the one-factorization F, then the next minimum vertex, $v_j \in V'$, is chosen as v_j. This process is continued until a suitable edge can be picked up so that F_i can be extended. If we are unsuccessful in extending F_i using all the vertices available to us, then we backtrack and rearrange the previous edge (u_{j-1}, v_{j-1}) of this matching. If $j = 0$ then we rearrange the last edge of the factor F_{i-1}.

3.2 The RAND Method

In this method, to add a j^{th} edge, (u_j, v_j), for the F_i^{th} matching, vertex u_j is chosen as the minimum vertex in V'. Vertex v_j is then picked up uniformly at random from the set $V' \setminus \{u_j\}$. If the edge (u_j, v_j) cannot be used in the extension of F_i, then we keep choosing vertex v_j uniformly at random

[2] Each vertex can be enumerated.

from $V' \setminus \{u_j\}$ until we find an edge that can be used in the extension of F_i. If we are unsuccessful in extending F_i using all the vertices available to us, then we backtrack and rearrange the previous edge (u_{j-1}, v_{j-1}) of this matching. If $j = 0$ then we rearrange the last edge of the factor F_{i-1}. The RAND method is justified in its name as the one-factorizations it generates spans the entire space of one-factorizations. The RAND method can produce any 1-factorization via suitable random choices during every step. The only difference is that we build and list the 1-factorizations in a lexicographic order.



(a) (b)

Figure 7: Figure (a) shows the time taken and (b) shows the number of backtrackings required by the RAND algorithm.

Both of our methods are better than the *hill climbing algorithm* [6] provided by Dinitz and Stinson for one-factorizations of k_{2n}. This is because our methods are known to terminate with certainty by producing a one-factorization when one exists and report otherwise [3] in case no one-factorization exists where as the *hill climbing algorithm* [6] does not guarantee to produce an one-factorization.

3.3 Experimental Setup and Observations

The experiments are conducted on a 4 * 3.00GHz, Intel(R) Xeon(TM) processor with 2 Ghz of main memory. The language used for implementing the RAND and MIN methods is ANSI C. The results obtained so were averaged over hundred rounds. It should be noted that since MIN is a deterministic approach the number of backtrackings will be system and time independent.

We shall now try to analyze our experiments from the above plots. For the MIN method, the time taken blows up heavily after a size of 42 or so. This is due to the large number of backtrackings required. However, the RAND method does not seem to require that many backtrackings and hence saves on time.

It can also be noticed that n a power of 2, the MIN method requires no backtracking where as RAND may require some as RAND does not follow any order in choosing the 1-factors. It is the lack of backtracking for the MIN method on complete graphs where the number of vertices is a power of 2 that led us into investigate the reason behind this.

Another interesting observation is with regard to the MIN method. Notice that the time taken and the number of backtrackings required resemble the function $\mathrm{sinc}(x) = \frac{\sin(x)}{x}$ with zeroes (and minimum time) at n being a power of 2. This suggests that there might exist a mechanism via which we can obtain a 1-factorization of K_{2^r+2} or K_{2^r-2} given a 1-factorization of K_{2^r} by systematically adding an edge and deleting an edge respectively. As can be observed, RAND does not show such connections but runs fast compared to MIN for most input sizes.

While we performed the experiments only on complete graphs, the MIN and the RAND method are however implemented in a general way so that they work also for any input graph. Both these methods

[3]For any input graph apart from K_{2n}.

also report a 1-factorization if one exists and report otherwise of no 1-factorization can be found for the input graph.

4 1-Factorizations for m-partite Graphs

In this section we describe how to obtain 1-factorizations for a complete m-partite graph where each partition has size n. This is denoted as $K_{n,n,\cdots,n}$ where there are a total of mn vertices. We require that m and n be even. We say that the graph has n parts and the vertices in the ith part, $i \in [n]$, are numbered from $(i-1)\cdot n$ to $i \cdot n$. In the above, $[n]$ denotes the set of natural numbers $\{1, 2, \cdots, n\}$.

To arrive at a 1-factorization of $K_{n,n,\cdots,n}$ we first define the following graph $H = (V_H, E_H)$ where $V_H = [m]$ and $(u, v) \in E_H$ if and only if there is an edge from some vertex in the uth part of $K_{n,n,\cdots,n}$ to some vertex in the vth part of $K_{n,n\cdots,n}$. The graph H can be seen as the graph obtained by compressing each part of the the m-partite graph into a single vertex and deleting multiple edges resulting out of the compression. The resulting graph is a K_m, the complete graph on m vertices.

Let K_m and K_n have a known 1-factorization. Let a 1-factorization of K_m be denoted as $F = \{F_1, F_2, \cdots, F_{m-1}\}$. Now given a 1-factorization of K_m, consider factor $F_i = (u_1, v_1), (u_2, v_2), \cdots,$ $(u_{m/2}, v_{m/2})$. The edges of F_i pair up vertices of K_m. Now suppose we expand each vertex u in F_i as a set S_u of n vertices and pair up the corresponding sets. That is, for the edge $(u_j, v_j) \in F_i$ we pair up the set S_u with S_v. In fact, the set S_u corresponding to vertex u is the set of vertices in the uth partition. Thus each 1-factor of K_m partitions the m-partite graph into $m/2$ disjoint bipartite sub-graphs. So we can treat each edge (u_j, v_j) in any 1-factor F_i of K_m to be expanded to a 1-factorization of the bipartite graph with S_u and S_v being the two sides of the bipartition. Moreover, each 1-factor of K_m will be indeed expanded into n 1-factors for $K_{n,n,\cdots,n}$ as the bipartite graph K_{S_u,S_v} has n 1-factors. Of course, we can consider any way of generating 1-factorization of a complete bipartite graph $K_{r,r}$ as described in the Section 2. But in this section we restrict ourselves to the 1-factorization obtained by Shift-and-Rotate strategy. For $K_{r,r}$ let us denote the 1-factors thus obtained as $J_1, J_2, \cdots, J_n$. We now describe the formation of the n 1-factors corresponding to F_i.

Let $F_i = \{(u_{i,\ell}, v_{i,\ell})\}_{\ell=1}^{m/2}$ be a 1-factor of K_m. Then the n 1-factors of $K_{n,n,\cdots,n}$ corresponding to F_i are given as follows. Let the n 1-factors of the bipartite graph corresponding to the edge $(u_{i,\ell}, v_{i,\ell})$, $1 \le \ell \le m/2$ be $H^j_{u_{i,\ell},v_{i,\ell}}$, where $j \in [n]$. Then,

$$\{G_j\}_{j=1}^n = H^j_{u_{i,1},v_{i,1}} \cup H^j_{u_{i,2},v_{i,2}} \cup \cdots \cup H^j_{u_{i,\ell},v_{i,\ell}}, j \in [n]$$

are the n 1-factors corresponding to the factor F_i of K_m. In the above union, $H^j_{u_{i,\ell},v_{i,\ell}}$ corresponds to the jth 1-factor in the complete bipartite graph corresponding to the ℓth edge in the ith 1-factor of K_m.

Thus, for each 1-factor F_i of K_m we have n 1-factors of $k_{n,n,\cdots,n}$. This results in $(m-1)\cdot n$ 1-factors for $K_{n,n,\cdots,n}$ each containing $m \cdot n/2$ edges. The total number of edges in $K_{n,n,\cdots,n}$ is exactly $m(m-1)n^2/2$ as $K_n, n, \cdots, n$ can be viewed as $\binom{m}{2}$ individual $K_{n,n}$ graphs.

Further, each edge of $K_{n,n,\cdots,n}$ will appear in exactly one 1-factor as can be shown in the following. Consider the edge (u, v). Let u belong to the ith partition and v belong to the jth partition. Now the 1-factorization of K_m has edge (i, j) in some 1-factor, say F_k. Then when we expand the 1-factor F_k we create n 1-factors corresponding to the bipartite graph K_{S_1,S_2} with $S_1 = [(i-1)\cdot n + 1, i \cdot n]$ and $S_2 = [(j-1)\cdot n + 1, j \cdot n]$. The edge (u, v) will appear in exactly one of these n 1-factors. Hence we have the following claim.

Claim 4.1 *There is a polynomial time algorithm to construct a 1-factorization of the m-partite complete graph on mn vertices, $K_{n,n,\cdots,n}$ where m and n are even.*

We see an example of the above approach in the next subsection along with an application.

4.1 Application: 1-Factorization for K_N, where N a multiple of 4

To illustrate the algorithmic usefulness of the above approach consider the case where m and n are both even. Then, for $N = m \cdot n$ which is a multiple of 4, let us try to construct a 1-factorization of K_N. We can represent K_N as the union of m complete graphs of size n vertices each and a complete m-partite graph $K_{n,n,\cdots,n}$. To arrive at a 1-factorization of K_N we can then proceed as follows.

Let the vertices of each K_N be numbered from 1 through N with the vertices in the range $[(i-1) \cdot n, i \cdot n]$ being the ith group for $i \in [m]$. So $K_N = (\cup_{i=1}^{m} K_i) \cup K_{n,n,\cdots,n}$. Let F^i denote a 1-factorization of the ith K_m. Then a 1-factorization of K_N is as follows:

$$F = \left(\cup_{i \in [m-1]}\right) G \cup H$$

where

$$G_i = \cup_{\ell \in [n]} J_{i,\ell}$$

with $J_{i,\ell}$ being the ith 1-factor in a 1-factorization of the ℓth K_m, and H is a 1-factorization of the complete m-partite graph $K_{n,n,\cdots,n}$. It can be observed that the above procedure does generate a 1-factorization of K_N using arguments similar to that of [18]. Hence the following claim.

Claim 4.2 *The above procedure constructs a 1-factorization of K_N, N being a multiple of 4, in polynomial time.*

We now explain the above approach with an example.

Example 4.3 If we take $N = 12$, and $n = 2$, then we are representing K_N as m paths of length 2 plus the complete $N/2$-partite graph where each part has just 2 vertices. We know that the 1-factorization of K_2 is simply (i, j) when the vertices are numbered i, j. The standard numbering we can use is to number the vertices of the ith K_2 as i and $i + 1$. Then, to construct a 1-factorization of K_N we first write the 1-factor corresponding to each K_2 as shown:

$$F_1 = (1, 2), (3, 4), (5, 6), (7, 8), (9, 10), (11, 12)$$

Below we construct a 1-factorization for the complete 6-partite graph with each partition having 2 vertices as described earlier. Let the resulting 1-factorization be $F_2, F_3, \cdots, F_{11}$. Then the set of 1-factors $F_1, F_2, \cdots, F_{11}$ is a 1-factorization of K_{12}.

A known 1-factorization for K_6 is:

$$\begin{array}{ccc}
(1,2) & (3,4) & (5,6) \\
(1,3) & (2,6) & (4,5) \\
(1,4) & (2,5) & (3,6) \\
(1,5) & (2,3) & (4,6) \\
(1,6) & (2,4) & (3,5)
\end{array}$$

So on expanding each of the above factors as described earlier and using the Shift-and-Rotate for arriving at a 1-factorization of a complete bipartite graph we get the following set of 1-factors for K_{12}. Here, 1-factors F_2 and F_3 are obtained from the 1-factor $\{(1, 2), (3, 4), (5, 6)\}$ of K_6 as follows. The edge $(1, 2)$ would be expanded to a 1-factorization of the complete bipartite graph with the partitions being vertices $\{1, 2\}$ and $\{3, 4\}$ of K_12. Similarly the edge $(3, 4)$ would be expanded to a 1-factorization of the complete bipartite graph with partition $\{5, 6\}$ and $\{7, 8\}$ and the edge $(5, 6)$ would be expanded to a 1-factorization of the complete bipartite graph with partition $\{9, 10\}$ and $\{11, 12\}$. This is because for the 6–partite complete multipartite graph $K_{2,2,2,2,2,2}$ each part has 2 vertices according to our numbering scheme. The other 1-factors F_4 through F_{11} are obtained by similarly expanding the 1-factors of K_6.

$$
\begin{array}{llllll}
F_1 = \{(1,2) & (3,4) & (5,6) & (7,8) & (9,10) & (11,12)\} \\
F_2 = \{(1,3) & (2,4) & (5,7) & (6,8) & (9,11) & (10,12)\} \\
F_3 = \{(1,4) & (2,3) & (5,8) & (6,7) & (9,12) & (10,11)\} \\
F_4 = \{(1,5) & (2,6) & (5,11) & (6,12) & (7,9) & (8,10)\} \\
F_5 = \{(1,6) & (2,5) & (5,12) & (6,11) & (7,10) & (8,9)\} \\
F_6 = \{(1,7) & (2,8) & (3,9) & (4,10) & (5,11) & (6,12)\} \\
F_7 = \{(1,8) & (2,7) & (3,10) & (4,9) & (5,12) & (6,11)\} \\
F_8 = \{(1,9) & (2,10) & (3,5) & (4,6) & (7,11) & (8,12)\} \\
F_9 = \{(1,10) & (2,9) & (3,6) & (4,5) & (7,12) & (8,11)\} \\
F_{10} = \{(1,11) & (2,12) & (3,7) & (4,8) & (5,9) & (6,10)\} \\
F_{11} = \{(1,12) & (2,11) & (3,8) & (4,7) & (5,10) & (6,9)\}
\end{array}
$$

We note that the results of this section can also be viewed in an existential sense that if K_m and K_n have a known 1-factorization then a 1-factorization for K_{mn} can be found. However, we were not able to use the above approach when m or n is odd due to the fact that complete graphs of odd order only have a near-1-factorization.

5 Conclusions and Open Problems

While we have reported new methods of arriving at 1-factorizations of complete graphs, several questions remain to be answered. One open question is to see whether we can extend(contract) a 1-factorization of K_{2^r} to a 1-factorization of K_{2^r+2} (*resp.* K_{2^r-2}) without any backtracking. Also interesting problems are to find non-algebraic ways of generating uniform or sequentially uniform 1-factorizations. Although there are estimates of the number of 1-factorization of K_{2n} [17], we are not aware of any estimates for the number of 1-factorization of $K_{n,n}$. It would be interesting to count the number of 1-factorization of $K_{n,n}$.

References

[1] N. Alon, B. Sudakov, and A. Zaks. Acyclic edgfe colourings of graphs. *J. of Graph Theory*, pages 157–167, 2001.

[2] L. D. Andersen. Factorizations of graphs. In *The CRC handbook of Combinatorial Designs*, pages 653–666. CRC Press, 1996.

[3] A. F. Beecham and A. C. Hurley. A scheduling problem wioth a simple graphical solution. *Journal of Australian Mathematical Soceity B*, 21:486–495, 1980.

[4] V. Bohossian and J. Bruck. Shortening array codes and the perfect 1-factorization conjecture. In *IEEE International Symposium on Information Theory*, pages 2799–2803, 2006.

[5] J. H. Dinitz, P. Dukes, and D. R. Stinson. Sequentially perfect and uniform one-factorizations of the complete graph. *The Electronic journal of combinatorics*, 12, 2005.

[6] J. H. Dinitz and D. R. Stinson. A hill-climbing algorithm for the construction of one-factorizations romm squares. *SIAM Journal of Algebraic and Discrete Methods*, 8:430–438, 1987.

[7] J. H. Dinitz and W. D. Wallis. Trains: an invariant for 1-factorizations. *Ars Combinatorica*, 32:161–180, 1991.

[8] M. J. Garnnell, T. S. Griggs, and J. P. Murphy. Some new perfect Steiner triple systems. *Journal of Combinatorial Designs*, 7:327–330, 1999.

[9] P. Kaski and P. R. J. Ostergard. One-factorizations of regular graphs of order 12. *The Electronic Journal of Combinatorics*, 12, 2005.

[10] E. Mendelsohn and A. Rosa. On some properties of 1-factorizations of complete graphs. *Congr. Numer*, 24:43–65, 1979.

[11] E. Mendelsohn and A. Rosa. One factorizations of th complete graph: A survey. *Journal of Graph Theory*, 9:43–65, 1985.

[12] R. Muthu, N. Narayanan, and C. R. Subramanian. Optimal acyclic edge colouring of grid like graphs. Technical report, The Institute of Mathematical Sciences, India, 2005.

[13] A. Rosa. Algebraic properties of designs and recursive constructions. *Congr. Numer.*, 13, 1975.

[14] A. Rosa and D. R. Stinson. One-factorizations of regular graphs and howell designs of small order. *Utilitas Math.*, 29:99124, 1986.

[15] E. Seah and D. R. Stinson. An enumeration of non-isomorphic one-factorizations and howell designs for the graph K_{10} minus a one-factor. *Ars Combinatorica*, 21:145161, 1986.

[16] C. R. Subramanian. Analysis of a heuristic for acyclic edge colouring. *Information Processing Letters*, 99:227–229, 2006.

[17] W. D. Wallis. One-factorizations of complete graphs. In *Contemporary Design Theory*, pages 692–731. Wiley, 1992.

[18] W. D. Wallis. *One factorizations*. Kluwer Academic Press, 1997.

[19] D. West. *Introduction to Graph Theory*. Prentice-Hall, 2001.