A GRAPH-AIDED METHOD FOR PLANNING TWO-LEVEL EXPERIMENTS WHEN CERTAIN INTERACTIONS ARE IMPORTANT

C. F. J. Wu and Youyi Chen

IIQP Research Report
RR-91-07

September 1991

A GRAPH-AIDED METHOD FOR PLANNING TWO-LEVEL EXPERIMENTS WHEN CERTAIN INTERACTIONS ARE IMPORTANT*

C.F.J. Wu and Youyi Chen

Department of Statistics and Actuarial Science
University of Waterloo
Waterloo, Ontario, N2L 3G1

ABSTRACT

In planning a fractional factorial experiment prior knowledge may suggest that some interactions are potentially important and should therefore be estimated free of the main effects. In this paper we propose a graph-aided method to solve this problem for two-level experiments. First we choose the defining relations for a 2^{n-k} design according to a goodness criterion such as the minimum aberration criterion. Then we construct all the nonisomorphic graphs that represent the solutions to the problem of simultaneous estimation of main effects and two-factor interactions for the given defining relations. In each graph a vertex represents a factor and an edge represents the interaction between the two factors. For the experiment planner the job is simple: draw a graph representing the specified interactions and compare it with the list of graphs obtained above. Our approach is a substantial improvement over Taguchi's linear graphs.

Keywords: feasible graphs, linear graphs, minimum aberration designs, interaction graphs, clear interaction, eligible interaction.

This paper is a substantial revision of "Graph-aided Assignment of Interactions in Two-level Fractional Factorial Designs", which appeared previously in this series as research report RR-89-10.

1 Introduction

In a factorial experiment some background knowledge may suggest that certain interactions are potentially important. The experimental plan should therefore be chosen so that these interactions can be studied without being aliased with each other and with the main effects. Importance of interactions can arise in several contexts. (1) An underlying physical mechanism may suggest a large interaction between some factors. For example, in a circuit pack assembly process, a machine mounts components onto the surfaces of printed wiring boards in two steps, epoxy application and component placement. In the first step, an epoxy dispenser head covers the board with a pattern of epoxy dots. In the second step a different tool places components on the epoxy dots in the board. Some important factors in the first step may include dispensing pressure, dispensing time, dispenser speed, dispenser height, epoxy type, and epoxy temperature. The important factors in the second step include component pressure and tool height. The responses are the x-axis and y-axis of the component position. Process knowledge suggests that the two steps are independent but the factors within each step are likely to interact with each other. (This example is based on information provided by Kwok Tsui.) (2) It is desired to estimate specific interactions even when they may turn out to be small. For example, to improve the throughput of a casting process, one may consider increasing the line speed. If the main effect of line speed and its interactions with other key factors such as iron chemistry (e.g. percent of copper, silicon, etc.) are insignificant, then the speed can be increased to achieve higher throughput without sacrificing quality. It is therefore important to be able to estimate the interactions between line speed and other factors. (3) In robust parameter design, the interaction between a control factor (e.g. diameter of pipes) and a noise factor (e.g. flow rate of inlet gas) may be important for making a product (in this case, a heat exchanger) insensitive to noise variations. For other examples see Shoemaker, Tsui and Wu (1991) and Wu, Mao and Ma (1990).

The interactions specified by the investigator can be estimated if a design of resolution V (or higher) is used. Quite often for economic or other reasons a smaller design is preferred, in which some 2-factor interactions (2fi's) are aliased with other 2fi's or main effects. The main question of interest is how to select a fractional factorial design that allows the main effects and the set of interactions specified by the investigator, which are called a *requirements set*

(Greenfield 1976), to be estimable. If the number of interactions is small, such a problem can often be solved by searching over different fractions until a satisfactory one is found, or by using the interaction table (see Section 4) to assign factors and interactions. Both require trial and error and give no clear-cut answer to the existence of solutions. When the number of interactions is large, the methods become quite unwieldy. Alternatively Taguchi (1959, 1960) proposed a method for assigning factors and interactions based on a class of graphs which he calls *linear graphs.* This method has gained popularity in recent years. In Section 2 we briefly review his method and point out some deficiencies. Taguchi (1987, Volume I) also used linear graphs for other purposes which are not the subject of this study.

In practice it is difficult for the investigator to predict exactly which interactions are significant. Lack of information or misjudgement may lead to misspecification of significant interactions. Therefore a satisfactory approach to this problem should meet the following objectives:

(i) it can estimate all the main effects and the interactions in the requirements set, assuming that the other interactions are negligible;

(ii) the estimation in (i) should be achieved under less stringent assumptions;

(iii) some interactions outside the requirements set can be estimated under less stringent assumptions.

Taguchi's method and several other methods reviewed in Section 4 can only satisfy the first objective. Hedayat and Pesotan (1992) study the existence and optimality of designs that satisfy (i).

The main purpose of this paper is to propose in Section 3 a graph-aided method that can meet the three objectives. We use the minimum aberration criterion (defined at the end of the section), which is more general than the notion of resolution, to capture the objectives (ii) and (iii). The method consists of two phases as described in Sections 3.1 and 3.2. In the first phase we obtain, for a given design, all the graphs that represent the sets of estimable interactions of the design. In the second phase the user draws a graph to represent the main effects and interactions in the requirements set and then compares it with the graphs for the minimum aberration design. Either a match is found or the same procedure is repeated for the next best design until a match is found. If no match is found for the given run size, we can either increase the run size or modify the requirements set. (The method also works for other design criteria.) For many 2-level fractional factorial designs of practical interest, we have obtained all the graphs for solving the posed problem. These are given in the Appendix. A computer implementation of the method is described in Section 3.3. In Section 4 we compare the relative merits of our method with Taguchi's linear graphs, an improved method due to Li et al. (1990) and other related methods. All the graph-theoretic definitions and results are given in Section 5.

Throughout the paper we use 2^{n-k} to denote a 2-level fractional factorial design with n factors and 2^{n-k} runs. Such a design is defined by a set of relations among its factors, which are called defining relations. For details, see Box, Hunter and Hunter (1978). The length of a defining relation is the number of factors it contains. The shortest length among the defining relations is called the *resolution* of the design. Usually designs of higher resolution are preferred. Designs of the same resolution can be further discriminated by using the following criterion. For a design $\boldsymbol{d}$ let $A_i(\boldsymbol{d})$ be the number of its defining relations with length i. For two designs $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$, $\boldsymbol{d}_1$ has *less aberration* than $\boldsymbol{d}_2$ if $A_j(\boldsymbol{d}_1) < A_j(\boldsymbol{d}_2)$, where j is the smallest i such that $A_i(\boldsymbol{d}_1) \neq A_i(\boldsymbol{d}_2)$. A design $\boldsymbol{d}$ has *minimum aberration* if there is no other design with less aberration than $\boldsymbol{d}$ (Fries and Hunter 1980). Note that minimum aberration implies maximum resolution.

2 Taguchi's Linear Graphs

We use the following example to illustrate Taguchi's method.

Example 1. Suppose for an industrial experiment we need to find a design for estimating 11 factors A, B, $\cdots$, K and all the 2-factor interactions among A, B, C, D, E and F. The six factors A to F could all belong to one step of a process such as the epoxy application of the surface mounting process described in Section 1. Taguchi's method will proceed as follows. The 11 factors are represented by 11 points and the 15 interactions specified above are represented by lines whose ends are the points representing the two factors in the interactions (see Figure 1a). This graph is compared with a list of 29 graphs for the

32-run designs, which can be found, for example, in the Appendix of Taguchi (1987). In the graphs prepared by Taguchi, a number is attached to each point and to each line. These numbers represent the column numbers of a 2^{n-k} design, in this case, a 32-run 2^{11-6} design. (Arrangement of columns in 2^{n-k} designs is explained in the next paragraph.) It turns out that Graph No. 1 of his list (see Figure 1b) includes the graph in Figure 1a as a subgraph. By comparing the two graphs, we can assign the factor names of the graph in Figure 1a to the corresponding column numbers in Figure 1b. That is, we assign factors A to F to columns 4, 2, 1, 16, 15, 8 respectively and their interactions to the columns which represent their corresponding lines. For example, BC is automatically assigned to column 3 because B is assigned to column 2, C to column 1 and column 3 is the interaction of columns 1 and 2. The remaining 5 factors are assigned to columns 19, 21, 22, 25, 26 in any manner since no specified interaction involves any of these factors.

(Figure 1 about here)

Note that the column numbers here and throughout the paper represent the columns in the 2-level designs arranged in the standard (Yates) order. For 16 runs, the 15 columns are represented by the 15 factorial effects of a 2^4 design arranged in the Yates order

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	12	3	13	23	123	4	14	24	124	34	134	234	1234

The extensions for 32 and 64 runs are obvious. For simplicity we do not include in this paper the column numbers for the lines, which can be easily read off the interaction tables given in Phadke (1989) and Taguchi (1987). For reasons related to run order and split-unit experimentation, each graph in Taguchi's collection has three different versions in column assignment.

This method and a collection of graphs date back to Taguchi (1959, 1960). Since then the graphs have been called *linear graphs*, a misnomer due to improper translation. A literal translation of its Japanese name should be *point-line* (or vertex-edge) graphs.

Taguchi's approach has one main advantage over the classical approach. Drawing a graph to represent the specified interactions is visually appealing. If a matched graph can be found, it is straightforward to assign factors to columns so that each of the specified interactions is assigned to a separate column. No computational search is required and it is user-friendly. It also serves as a reminder that some 2fi's in resolution III and IV designs are estimable if we can assume the other 2fi's are negligible. The method has some deficiencies, however. First, statistical properties of the designs represented by Taguchi's graphs are not considered. The designs for all graphs but two in his collection are of resolution III (see Table 1). It will be shown in Table 2, Section 3 that in many cases, better designs in terms of the maximum resolution or minimum aberration criteria can be found. One exception is a 2^{5-1} design of resolution V, for which all the 2fi's are estimable and therefore does not need any graph method for interaction assignment. Second, for designs with at least 16 runs, the total number of graphs is too large to be all included in his collection. For example, for the 16-run designs, only 6 types of graphs are given, out of more than 800 types of graphs (Taguchi 1987, p.188). Since his collection is not exhaustive, there is a good chance that a solution (i.e. a graph) may be missed by using his method. Furthermore, in some important cases, e.g., the 2^{6-2}, 2^{9-5}, 2^{12-7} designs and 64-run designs with less than 17 factors, not a single graph is given in his collection (see Table 1).

(Table 1 about here)

3 A Graph-Aided Method

In this section we propose a method that combines the appealing features of Taguchi's approach and good design properties afforded by the classical approach. To ensure that the designs represented by the graphs have good properties and to avoid the enumeration of unnecessarily many graphs, the proposed method first chooses a design according to a goodness criterion. In this paper we use the minimum aberration criterion. Starting with a minimum aberration design, there are numerous ways of assigning the factors and some interactions to distinct columns so that the defining relations of the design are satisfied. Any such assignment is called *feasible.* Each feasible assignment can be represented by a graph as shown in Section 2. Such a graph is called a *feasible graph.* Two graphs are isomorphic if one can be obtained from the other by relabelling the vertices. If a complete set of nonisomorphic feasible graphs is available, then the graph representing the specified interactions, which we call the *requirements graph*, can be compared with this set of graphs. If a matched graph is found, the column assignment of factors and interactions proceeds as in Taguchi's original recipe. If no matched graph can be found, no feasible assignment exists. One can then repeat the same procedure with the second best design (of the same run size) according to the minimum aberration criterion and possibly continue with the third best, etc. until a feasible assignment is found. If none can be found, one can either increase the run size or modify the requirements. Further discussion is given in Section 3.2.

Note that the proposed method works equally well for other criteria which may be more appropriate than the aberration criterion or reflect some *a priori* knowledge on the effects. The gist of our approach is to "find the best design according to a prescribed criterion subject to the requirement that all the main effects and the specified interactions be estimable."

The method consists of two phases. In the first phase, a complete set of nonisomorphic feasible graphs is constructed (see Section 3.1). For a given 2^{n-k} design, this work is required only once. As long as these graphs are available, no user's effort is required in this phase. In the second phase, the user obtains a solution by comparing his requirements graph with the feasible graphs constructed in phase I. The detail is given in Section 3.2. Sometimes feasible graphs will be referred to simply as graphs.

3.1 Construction of Nonisomorphic Feasible Graphs

We use Example 2 to illustrate the construction method.

Example 2. Find a design for which the 6 factors A, B, C, D, E, F and the following interactions AB, BC, CD, CF, DE, EF, DF are estimable (assuming other interactions are negligible.)

First consider a 2^{6-2} minimum aberration design in which the factors are labeled 1, 2, 3, 4, 5, 6 and the defining relations are

$$I = 1235 = 2346 = 1456.$$

The alias relations of the 15 2fi's are given below,

$$12 = 35,\ 13 = 25,\ 14 = 56, 24 = 36,$$

$$34 = 26,\ 45 = 16,\ 23 = 15 = 46.$$

They form 7 groups, each of the first 6 having 2 members, the last group having 3 members. Within each group one 2fi is estimable if the others in the same group are assumed to be negligible. Therefore at most seven 2fi's are estimable. Each set of seven 2fi's can be represented by a graph. Altogether there are 192 ($=2^6 \times 3$) graphs, out of which only seven are nonisomorphic (given in the Appendix). Isomorphic graphs should be counted once since they correspond to the same interaction assignment after renaming the factors. In Section 5 we give a method for testing graph isomorphism.

The enumeration method needs improvement when a large number of graphs are being tested. For example, the 2^{10-5} minimum aberration design has 5,242,880 graphs under test. By exploiting some symmetry in the groups of aliased 2fi's, this number can be drastically reduced. This is demonstrated with a simpler example, the 2^{6-2} design in Example 2. The feasible graphs of this design can be divided into two groups, namely,

$$\textbf{group 1:}\quad \begin{array}{llll} 12, & 13 = 25, & 14 = 56, & 24 = 36, \\ & 34 = 26, & 45 = 16, & 23 = 15 = 46. \end{array}$$

$$\textbf{group 2:}\quad \begin{array}{llll} 35, & 13 = 25, & 14 = 56, & 24 = 36, \\ & 34 = 26, & 45 = 16, & 23 = 15 = 46. \end{array}$$

By switching the labels of 2 and 3, and of 1 and 5 in group 2, group 2 becomes group 1. Therefore the isomorphism test need only be carried out in group 1. This cuts by half the total number of graphs. Indeed, after group 2 is deleted, further reduction can be achieved by repeating a similar relabelling procedure for group 1. By applying this method to the 2^{10-5} design, the number of graphs under test is reduced from 5,242,880 to 32,768.

In resolution III designs there are some 2fi's that are aliased with main effects. These 2fi's are not eligible for graph construction since the main effects must be estimable. For the purpose of graph construction we call a 2fi *eligible* if it is not aliased with any main effects. Any 2fi of a resolution IV design is eligible. Among the eligible 2fi's, some may not be aliased with any main effects or other 2fi's. We call them *clear*. Clear 2fi's are estimable under the weaker assumption that 3-factor and higher order interactions are negligible. To distinguish clear 2fi's from the others, we use dashed lines for them in the constructed graphs. This graphical distinction serves a useful purpose, i.e. it allows the user to assign the more important 2fi's in the requirements set to the dashed lines in the graph.

We use a simple example to illustrate the concept. In the 2^{6-2} design defined by $I = 125 = 2346 = 13456$, 12, 15 and 25 are ineligible since each is aliased with a main effect. Among the 12 eligible 2fi's, 13, 14, 16, 35, 45, 56 are clear since they do not appear in 125 or 2346 of the defining relations; the other six appear as three aliased pairs, $23 = 46$, $24 = 36$, $34 = 26$. There are four nonisomorphic graphs for this design, each with six dashed lines (clear 2fi's) and three solid lines (eligible but not clear 2fi's).

By implementing this method and the graph isomorphism test in Sections 5 and 3.3, we can obtain graphs for many useful 2-level designs. Some of them are given in the Appendix and summarized in Table 2. In the Appendix we only consider designs that have a small or moderate number of nonisomorphic graphs. For 2^{6-2} and 2^{7-3}, we give a complete list of designs arranged in the order of the aberration criterion. For 2^{8-4} and 2^{9-5}, we only give the best and second best according to the aberration criterion. To save space, we only consider the minimum aberration designs for the remaining cases. Underneath each graph we give its degrees $\boldsymbol{d} = (d_i)$. If two graphs for the same design have identical $\boldsymbol{d}$, we also give their respective extended degrees $\boldsymbol{D} = (D_i)$ underneath $\boldsymbol{d}$. (Definitions of $\boldsymbol{d}$ and $\boldsymbol{D}$ are given in Section 5.) In most cases we can use $\boldsymbol{d}$ and $\boldsymbol{D}$ to distinguish nonisomorphic graphs.

(Table 2 about here)

Some pertinent points about the graphs.

1. A design with less aberration may have a smaller number of eligible 2fi's or of clear 2fi's (see column 4 of Table 2). For example, the minimum aberration 2^{6-2} design has seven eligible 2fi's and no clear 2fi's while the third best design according to the aberration criterion has nine clear 2fi's out of nine eligible 2fi's. Of course the latter design has the disadvantage that five of its six main effects are aliased with some 2fi's. This is the kind of trade-off users have to make.

2. For most designs our definition of graph isomorphism does not take into account the difference between solid lines and dashed lines. Only for small designs do we make the distinction. For example the graph for the 2^{4-1}_{III} design with $I = 234$ is identical to that for the 2^{4-1}_{IV} design. The former graph may be preferred since all the three 2fi's are clear. This is particularly useful when the 2fi's between factor 1 and the other three are important.

3. Certain subgraphs are of interest in design applications. Three types are considered here. A complete subgraph means that every 2fi between the factors (i.e. vertices) in the subgraph is estimable. The surface mounting experiment in Section 1 is one such example. In the last column of Table 2, we give the number of vertices in a maximal complete subgraph. The second type has the degrees $(n, 1, \ldots, 1)$, where the interactions between a specific factor (say, the line speed in the casting experiment in Section 1) and n other factors are important. The third type is the bipartite graphs. A simple example is the last graph for the 2^{6-2} design with $I = 125 = 2346$. The six factors are divided into two groups, each of three. There are nine 2fi's between groups, none within groups.

4. The graphs for the 8-run designs were given by Taguchi (1960, 1987). The graphs for the 2_{IV}^{7-3} and 2_{IV}^{8-4} designs were found by Li et al. (1990). As pointed out by Li et al., the seven graphs for the 2_{IV}^{6-2} design and the 17 graphs for the 2_{IV}^{7-3} design can be obtained as subgraphs of those for the 2_{IV}^{8-4} design.

Finally we note that, although the graph enumeration is based on the defining relations, the graphs provide a set of *explicit* solutions to the interaction assignment problem. Finding a solution directly from the defining relations can be laborious and frustrating. Use of the graphs can greatly reduce this tedious work.

3.2 Assignment of Factors and Specified Interactions

This is the second phase of the proposed method. We suggest the following steps for simultaneous assignment of factors and specified interactions. Call the set of specified interactions a requirements set.

1. Choose a 2^{n-k} design with n being the number of factors.

2. If the number of factors that appear in the specified interactions does not exceed the number of vertices in a maximal complete subgraph (last column of Table 2), assign these factors to the vertices of this subgraph and the remaining factors to other vertices in any manner. Otherwise, go to step 3.

3. Draw a graph to represent the factors and the specified interactions by vertices and edges respectively. Compare it with the complete set of feasible graphs for this design. If it is isomorphic to a graph or subgraph in the set of feasible graphs, assign the factors and interactions to the columns as indicated on the latter graph. Otherwise, go to step 4.

4. Choose the next best 2^{n-k} design according to a goodness criterion. Go to step 2.

If the design in step 1 is of resolution V, all 2fi's are estimable and there is no need to go through the whole exercise. Initially the 2^{n-k} design in step 1 should be chosen to be a best design according to some goodness criterion such as minimum aberration. This and step 4 ensure that the resulting design has good properties. If for a fixed run size, either no solution exists or only a poor solution is available, one should either increase the run size or modify the requirements set so that a satisfactory solution can be found. For example, by comparing the two graphs that do not match, one can easily determine the smallest number of 2fi's that must be sacrificed in the requirements set. The requirements set should be treated with some flexibility since in practice, except for some obvious ones, it is difficult to guess exactly which interactions are significant or important.

If the requirements graph drawn in step 3 has fewer edges than the feasible graphs, it is compared with the subgraphs of the feasible graphs. This requires an additional step of finding subgraphs with the same $\boldsymbol{d}$ and $\boldsymbol{D}$ as the requirements graph. Algorithms for testing graph isomorphism in step 3 are discussed in Section 5.

We use Example 2 to illustrate these steps.

Example 2 (continued).

1. Choose the 2^{6-2} minimum aberration design as given in Section 3.1.

2. There are 6 factors that appear in the interactions but only 4 vertices in a maximal complete subgraph.

3. A graph is drawn to represent the factors and interactions (Figure 2). Its $\boldsymbol{d}$ and $\boldsymbol{D}$ are respectively (3 3 3 2 2 1) and (8 8 8 6 4 2), which match graph no. 6 for this

design. The two are isomorphic. Therefore, by checking the column numbers in the 16-run saturated design, factors A, B, C, D, E, F are assigned to columns 7, 8, 1, 4, 14, and 2 respectively and the seven interactions AB, BC, CD, CF, DE, EF, and DF are assigned to columns 15, 9, 5, 3, 10, 12, and 6 respectively.

(Figure 2 about here)

Example 3 serves to illustrate the possibility of a no match in step 3.

Example 3. Same as Example 2 except the set of interactions is changed to {AB, AF, BC, CD, CF, DE, EF}.

1. Same as in Example 2.

2. Same as in Example 2.

3. The graph (see Figure 3) representing the factors and interactions has $\boldsymbol{d}$=(3 3 2 2 2 2) which does not match any of the 7 graphs for the 2^{6-2} minimum aberration design.

4. Choose another 2^{6-2} design with the defining relations $I = 125 = 2346 = 13456$. It is the second best according to the aberration criterion. It has four nonisomorphic feasible graphs, among which the last graph contains the graph in Figure 3 as a subgraph.

(Figure 3 about here)

3.3 Computer Implementation

The method as described in Sections 3.1 and 3.2 is "manual" but can also be automated with a computer. Manual assignment has some advantages. It encourages the investigator who is usually a nonstatistician but is familiar with the subject matter, to be more involved in the choice of design. It enhances a better understanding of the problem. If the manual method does not lead to a solution, the investigator gains enough knowledge from the process which helps him or her to decide which interactions in the requirements set can be dropped so that a solution can be obtained without increasing the run size.

On the other hand a computer implementation of the proposed method will be fast and easy to use. This is especially appealing when the number of nonisomorphic graphs is large or the graphs are too complex. Two automated versions are considered here.

If the number of nonisomorphic graphs is not too large, say, at most in the thousands, each of these graphs can be stored together with the $\boldsymbol{d}$ and $\boldsymbol{D}$ values. Each time a requirements set is submitted by the user. The algorithm codes it as a graph with $\boldsymbol{d}$ and $\boldsymbol{D}$ values. The latter graph is then compared with the stored graphs using a graph isomorphism algorithm. The rest is straightforward.

If the number of nonisomorphic graphs is too large to be enumerated or stored, a more involved algorithm is required. It first "draws" a graph for the requirements set as in the first version. Then for a given design it generates the graphs sequentially using the method of Section 3.1. Each time a graph is generated, it is "compared" with the drawn graph. If they are isomorphic, a solution is found and the search is terminated. If all the graphs are exhausted without a match, no solution exists. The algorithm will then generate another design according to the method of Section 3.2, or the user is asked to submit another design. In the worst case the amount of search can be prohibitive. If a solution exists, the amount of work is more modest since the solution is usually found long before all the graphs are exhausted. For more complex problems, this algorithm needs improvement.

4 Comparison with Taguchi's Approach and Related Methods

By building on the strength of Taguchi's graph approach(see Section 2), our proposed method has some additional advantages. First it guarantees that the selected design has good properties as dictated by the user through the goodness criterion. Second our method can identify a complete set of solutions to the problem subject to computing constraints. Therefore it either finds a solution or informs the users to increase the run size or modify the problem.

From comparing the results in Tables 1 and 2, the advantages of our approach become transparent. For the 16-run designs, we can use resolution IV designs for 7 and 8 factors whereas the designs for Taguchi's graphs have only resolution III. His Graph No.5 for the

2^{10-6} design is the only one with minimum aberration in his collection. Since only six graphs are given, it is quite likely that his approach will find no solution when it actually exists. We return to Example 1 to illustrate the difference between the two approaches.

Example 1 (continued). Our approach works as follows.

1. Choose a 2^{11-6} design with resolution IV.
2. There are six factors that appear in the set of interactions. The design in step 1 has a feasible graph (Figure 4) which has a complete subgraph with 6 vertices. Assign the six factors to columns 1, 2, 4, 8, 21, 26 and the rest is straightforward.

(Figure 4 about here)

For this problem Taguchi's graph (Figure 1b) also has a complete subgraph with six vertices. But the corresponding design has resolution III, that is, the main effects for the five factors representing columns 15, 21, 22, 25, 26 are aliased with some 2fi's. The design in step 1 does not have this shortcoming since it is of resolution IV. On the other hand, Taguchi's graph can accommodate five more 2fi's than our graph (Figure 4). As observed in Section 3.1, there is a trade-off between higher resolution and more eligible 2fi's.

Li and Chiou (1989) first pointed out that the 2^{8-4} and 2^{7-3} designs for Taguchi's linear graphs have resolution III. Li et al. (1990) found graphs for some resolution IV designs. They do not consider resolution III designs or adopt a goodness criterion (such as the aberration criterion) for comparing designs.

For the rest of the section, we discuss several other methods. A method, which has been around for a long time, is based on the table of interactions. In such a table, the column number of the interaction between any two columns of a saturated 2-level design is given in an appropriate entry. An interaction and its two factors are assigned to the respective columns according to the table. The same step is repeated for other interactions in the set. If they are all assigned to different columns, a solution is found. Otherwise it may be repeated by trial and error.

A graph version of the interaction table was suggested by Kacker and Tsui (1990). They constructed an interaction graph in which half of the columns in the interaction table were represented by its vertices and the remaining columns by its edges according to the relations in the table. Their suggested method of column assignment for factors and interactions is, apart from a few elaborate rules, very similar to the one for the interaction table. Since both are based on trial and error, there is no guarantee that a solution can be found or that the impossibility of solution can be declared when none exists. Like our method, it is quite simple and use of graphs encourages user's participation. As remarked by Kacker and Tsui (1990), the interaction graph method is difficult to implement for more than 16 runs. Because of visual complexity any graph method becomes impractical when the number of edges in a graph is large, say more than 20. Such is the case for many 32-run designs and most 64-run designs.

By modifying a method of Greenfield (1976), Franklin and Bailey (1977) proposed a method for 2-level designs, which does not use graphs. Franklin (1985) extended it to p^r-level designs, p being a prime. It is an iterative method and therefore does not have the appeal of simplicity. Like linear graphs and interaction graphs, it cannot ensure good properties of the design generated. On the other hand it is automatic, fast and can handle higher order interactions. It would be interesting to compare this method and the automated methods in Section 3.3.

5 An Algorithm for Testing Graph Isomorphism

First we review some graph-theoretic terms. A *graph* G is defined by a set $V(G)$ of elements called *vertices*, a set $E(G)$ of elements called *edges*, and a relation of incidence, which associates with each edge either one or two vertices called its end. Two vertices u and v of G are said to be adjacent if $uv \in E$. The *degree* $d(u)$ of a vertex u is the number of edges incident with u. A vertex of degree 0 is called an *isolated vertex*. A graph H is said to be a *subgraph* of a graph G if $V(H) \subset V(G)$ and $E(H) \subset E(G)$. A graph is *complete* if every pair of vertices is adjacent.

Two graphs G_1 and G_2 are isomorphic if G_2 can be obtained from G_1 by relabelling the vertices of G_1. The graph isomorphism problem is that of finding a good algorithm for determining whether two given graphs are isomorphic. A quantity associated with a graph is a graph-invariant if it is invariant under an isomorphic map. The degree sequence $\boldsymbol{d}=(d_1, \ldots, d_v)$, where d_i are the degrees in descending order of the v vertices of a graph, is a graph-invariant. For each vertex i, define its *extended degree* to be $D_i = \sum d_{ij}$, where d_{ij} are the degrees of all vertices adjacent to i. For a graph we use $\boldsymbol{D} = (D_i)$ to denote its extended degrees. Then the set $\{(d_i, D_i)\}_{i=1}^{v}$ arranged in descending order of d_i is a graph-invariant which better discriminates nonisomorphic graphs than $\boldsymbol{d}$.

For a review on efficient algorithms for testing graph isomorphism, see Read and Corneil (1977) and Hoffman (1982). However not all the existing methods are suitable for our propose. For most practical experiments the corresponding graphs have a moderate number of vertices and edges. Therefore implementing a sophisticated general purpose algorithm is expensive and may not work effectively for such graphs. Instead we adopt the following simple algorithm.

For any two graphs G_1 and G_2, we first test whether the degrees and extended degrees $\{(d_i, D_i)\}$ of G_1 match those of G_2. If not, the two are not isomorphic. If the first test is passed, we use the following method to complete the test. Let V_{1i} (and resp. V_{2i}) be the set of vertices in G_1 (and resp. G_2) with the same (d_i, D_i). Test the isomorphism of V_{1i} and V_{2i} by first labelling the vertices and edges in V_{1i} and then permuting the vertices in V_{2i} until one permutation is found, whose set of edges matches that of the labelled V_{1i}. If no such permutation can be found, the two graphs are not isomorphic and the test is aborted. In the worst case it takes $\prod(n_i!)$ permutations, where n_i is the size of V_{1i} (and also of V_{2i}). But the average number of permutations is much smaller.

Acknowledgements: The revision benefits from the insights and constructive suggestions of the Associate Editor. The referees' comments are also helpful. Research was supported by the Natural Sciences and Engineering Research Council of Canada, General Motors of Canada Limited, and the Manufacturing Research Corporation of Ontario.

6 References

Box, G.E.P., Hunter, W. G. and Hunter , J. S. (1978), *Statistics for Experimenters.*, New York. John Wiley.

Franklin, M. F. (1985), "Selecting Defining Contrasts and Confounded Effects in p^{n-m} Experiments," *Technometrics*, **27**, 165-172.

Franklin, M. F. and Bailey, R. A. (1977), "Selection of Defining Contrasts and Confounded Effects in Two-Level Experiments," *Applied Statistics*, **26**, 321-326.

Fries, A. and Hunter, W. G. (1980), "Minimum Aberration 2^{k-p} Designs," *Technometrics*, **22**, 601-608.

Greenfield, A. A. (1976), "Selection of Defining Contrasts in Two-level Experiments," *Applied Statistics*, **25**, 64-67.

Hedayat, A.S. and Pesotan, H. (1992), "Two-level factorial designs for main effects and selected two-factor interactions," *Statistica Sinica*, to appear.

Hoffman, C.M. (1982), *Group-theoretic Algorithms and Graph Isomorphism*, Springer-Verlag, Berlin.

Kacker, R. N. and Tsui, K (1990), "Interaction Graphs: Graphical Aids for planning Experiments," *Journal of Quality Technology*, **22**, 1-14.

Li, C.C. and Chiou, K.M. (1989), "Some explanation and evaluation of the linear graphs of Taguchi's orthogonal arrays $L_8(2^7)$ and $L_{16}(2^{15})$," *Journal of the Chinese Institute of Industrial Engineers*, **6**, 43-47, (in Chinese).

Li, C. C., Washio, Y., IIda, T. and Tanimoto, S. (1990), "New Linear Graphs for Orthogonal Array $L_{16}(2^{15})$," *Journal of the Chinese Institute of Industrial Engineers*, **7**, 17-23, (in Chinese).

Phadke, M.S. (1989), *Quality Engineering Using Robust Design*, Prentice Hall.

Read, Ronald C. and Corneil, Derek G. (1977), "The Graph Isomorphic Disease," *Journal of Graph Theory*, **1**, 339-363.

Shoemaker, A., Tsui, K. L. and Wu, C. F. J. (1991), "Economical Experimentation Methods for Robust Design," *Technometrics*, to appear.

Taguchi, G. (1959), "Linear Graphs for Orthogonal Arrays and Their Application to Experimental Design, with the Aid of Various Techniques," *Rep. Stat. Appl. Res.*, JUSE, **6**, No. 4, 1-43.

Taguchi, G. (1960), "Table of Orthogonal Arrays and Linear Graphs," *Rep. Stat. Appl. Res.*, JUSE, **6**, No. 5, 1-52.

Taguchi, G (1987), *Systems of Experimental Design* (2 vols.). Translated and published in English by UNIPUB, New York.

Wu, C. F. J., Mao, S. S. and Ma, F. S. (1990), "SEL: A Search Method Based on Orthogonal Arrays," in *Statistical Design and Analysis of Industrial Experiments* (S. Ghosh, Ed.), Marcel Dekker, 279-310.

graph number	*design type*	resolution	*graph number*	*design type*	resolution
L_8					
(1,2)	2^{4-1}	IV			
L_{16}					
(1)	2^{5-1}	V	(2)	2^{7-3}	III
(3,4,6)	2^{8-4}	III	(5)	2^{10-6}	III
L_{32}					
(1)	2^{11-6}	III	(2)	2^{13-8}	III
(9)	2^{14-9}	III	(3,4,6)	2^{15-10}	III
(7,11,12,13)	2^{16-11}	III	(5)	2^{17-12}	III
(8)	2^{18-13}	III	(10)	2^{20-15}	III
L_{64}					
(5)	2^{17-11}	III	(1)	2^{23-17}	III
(2,3)	2^{26-20}	III	(4,9)	2^{27-21}	III
(6,7)	2^{29-23}	III	(10)	2^{31-25}	III
(8)	2^{32-26}	III			

Table 1: **Taguchi's linear graphs and their corresponding designs.** The graph numbers are from Taguchi (1987).

Design	Generators	# of nonisomorphic graphs	# of eligible 2fi's (# of clear 2fi's)	# of vertices of complete subgraphs
8 run				
2_{IV}^{4-1}	$I = 1234$	2	3(0)	3
2_{III}^{4-1}	$I = 234$	1	3(3)	2
2_{III}^{5-2}	$I = 124 = 135$	1	2(0)	2
16 run				
2_{IV}^{6-2}	$I = 1235 = 2346$	7	7(0)	4
2_{III}^{6-2}	$I = 125 = 2346$	4	9(6)	4
2_{III}^{6-2}	$I = 125 = 1246$	1	9(9)	2
2_{III}^{6-2}	$I = 125 = 346$	1	7(5)	3
2_{IV}^{7-3}	$I = 1235 = 2346 = 1347$	17	7(0)	3
2_{III}^{7-3}	$I = 1235 = 1246 = 347$	15	8(2)	4
2_{III}^{7-3}	$I = 12345 = 236 = 1237$	5	8(4)	3
2_{III}^{7-3}	$I = 1235 = 1246 = 127$	3	6(0)	3
2_{III}^{7-3}	$I = 125 = 1246 = 247$	1	7(6)	3
2_{IV}^{8-4}	$I = 1235 = 2346 = 1347$ $= 1248$	26	7(0)	3
2_{III}^{8-4}	$I = 125 = 2346 = 1347$ $= 1248$	23	7(1)	3
2_{III}^{9-5}	$I = 1235 = 2346 = 1347$ $= 1248 = 169$	35	6(0)	3
2_{III}^{9-5}	$I = 125 = 2346 = 1347$ $= 248 = 139$	14	6(0)	3
2_{III}^{10-6}	$I = 1235 = 2346 = 1347$ $= 1248 = 169 = 34t_0$	22	5(0)	3
2_{III}^{11-7}	$I = 1235 = 2346 = 1347$ $= 1248 = 169 = 34t_0 = 24t_1$	10	4(0)	3
2_{III}^{12-8}	$I = 1235 = 2346 = 1347$ $= 1248 = 169 = 34t_0 = 24t_1$ $= 14t_2$	4	3(0)	3
2_{III}^{13-9}	$I = 1235 = 2346 = 1347$ $= 1248 = 169 = 34t_0 = 24t_1$ $= 14t_2 = 23t_3$	2	2(0)	2

Design	Generators	# of nonisomorphic graphs	# of eligible 2fi's (# of clear 2fi's)	# of vertices of complete subgraphs
32 run				
2^{7-2}_{IV}	$I = 23456 = 13457$	2	18(15)	6
2^{8-3}_{IV}	$I = 23456 = 13457$ $= 12458$	7	20(13)	6
2^{9-4}_{IV}	$I = 23456 = 13457$ $= 12458 = 12359$	66	21(8)	6
2^{10-5}_{IV}	$I = 23456 = 13457$ $= 12458 = 12359 = 1234t_0$	1676	21(0)	6
64 run				
2^{9-3}_{IV}	$I = 12347 = 13568 = 34569$	2	33(30)	8
2^{10-4}_{IV}	$I = 12347 = 13568 = 34569$ $= 1235t_0$	3	39(33)	8
2^{11-5}_{IV}	$I = 3457 = 12348 = 1269$ $= 2456t_0 = 1456t_1$	14	44(33)	8

Table 2: **Summary of graphs for 2^{n-k} designs.** The (10+i)-th factor is represented by t_i. The number of clear 2fi's is given in brackets in column 4.

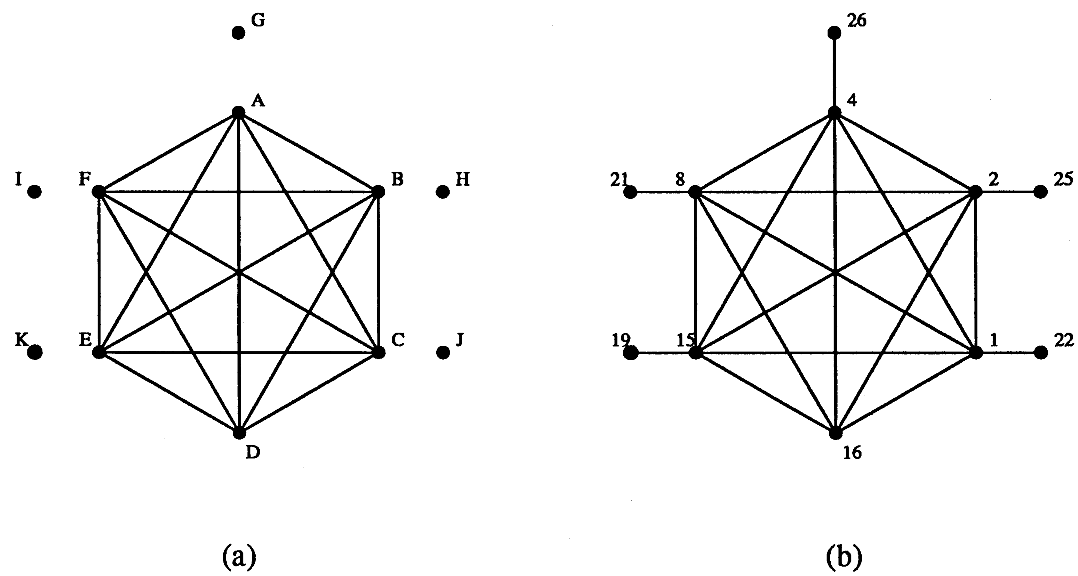


Figure 1: Graph a represents the specified interactions in Example 1.
Graph b is found to match the requirements.

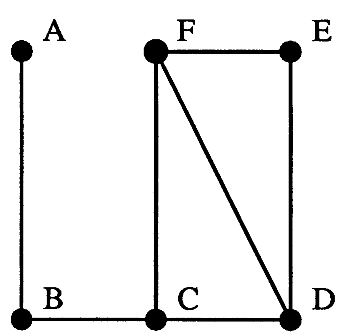


Figure 2: Graph represents the specified interactions in Example 2.

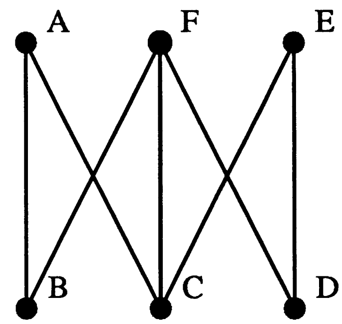


Figure 3: Graph represents the specified interactions in Example 3.

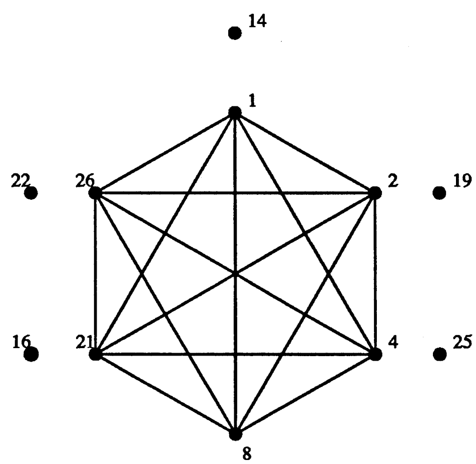


Figure 4: This graph gives a different solution than Taguchi's Figure 1b.

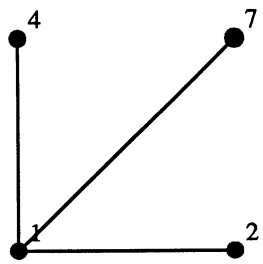
Appendix

Nonisomorphic feasible graphs for some 2^{n-k} designs.

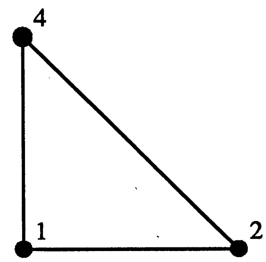
In each graph the number for a vertex denotes the column number of the corresponding factor. The column number of an interaction, represented by a line in the graph, can be obtained from the column numbers of its two vertices through the interaction table. A dashed line represents a clear 2-factor interaction, i.e. not aliased with any main effects or other 2fi's. (More important 2fi's in the requirements set should be assigned to dashed lines.) Below each graph is its degrees and extended degrees $\binom{d_1,\cdots,d_v}{D_1,\cdots,D_v}$, where d_i are arranged in descending order. If there is only one graph with a given $\boldsymbol{d}=(d_i)$, D_i are not given. In these graphs isolated vertices are omitted. Note that they correspond to factors which do not appear in the specified interactions. We do not give them since they can be obtained from the table next to each design. In the table the column numbers of the n factors are given. Any column number in this table that does not appear in the graph can be assigned to any factor that does not appear in any of the specified interactions.

2^{4-1} : I=1234

Factor #	1	2	3	4
Col #	1	2	4	7



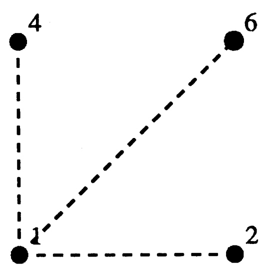
3 1 1 1



2 2 2

2^{4-1} : I=234

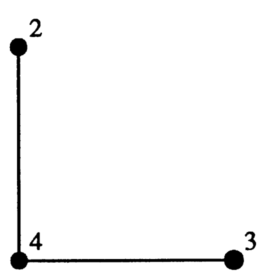
Factor #	1	2	3	4
Col #	1	2	4	6



3 1 1 1

2^{5-2} : I=124=135

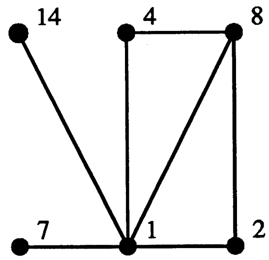
Factor #	1	2	3	4	5
Col #	1	2	4	3	5



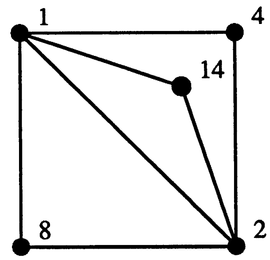
2 1 1

2^{6-2} : I=1235=2346

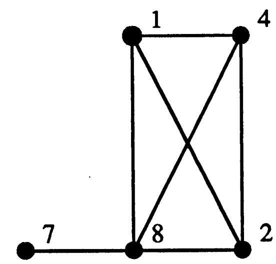
Factor #	1 2 3 4 5 6
Col #	1 2 4 8 7 14



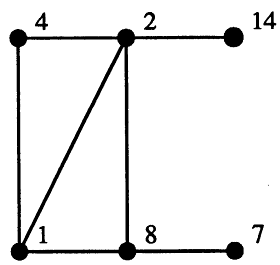
532211



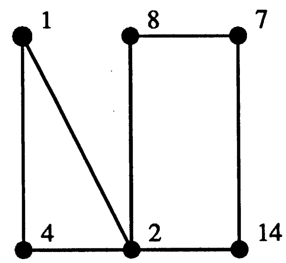
442220



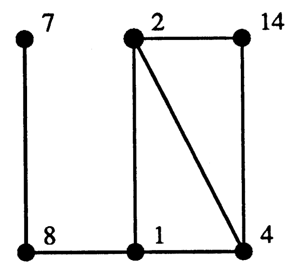
433310



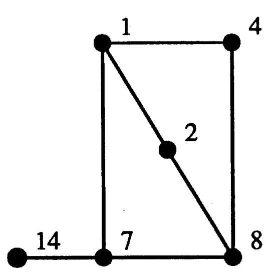
433211



422222



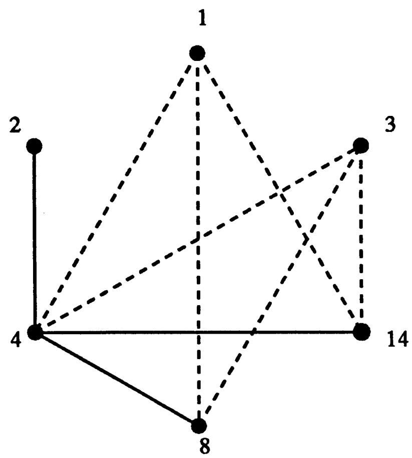
333221
888642



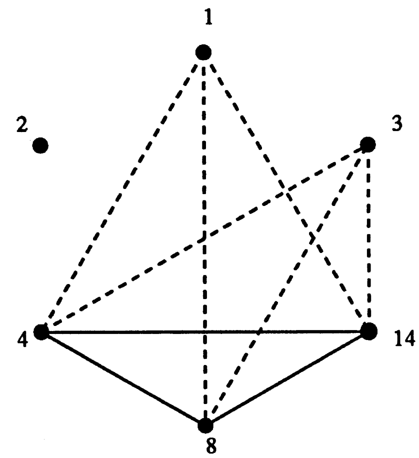
333221
777663

2^{6-2} : I=125=2346

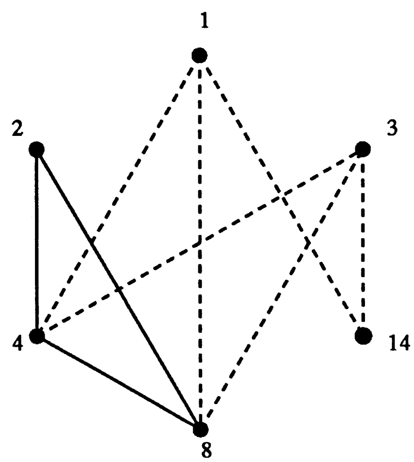
Factor #	1 2 3 4 5 6
Col #	1 2 4 8 3 14



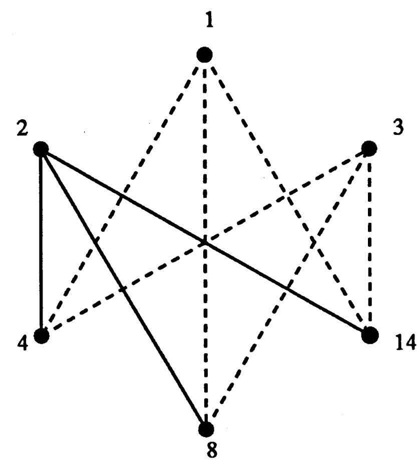
5 3 3 3 3 1



4 4 4 3 3 0



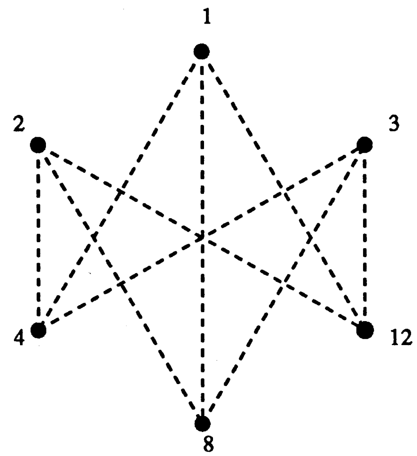
4 4 3 3 2 2



3 3 3 3 3 3

2^{6-2} : I=125=346

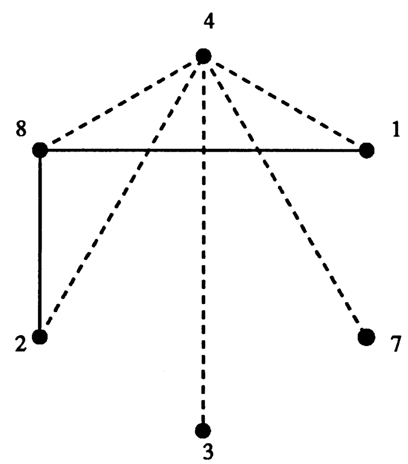
Factor #	1 2 3 4 5 6
Col #	1 2 4 8 3 12



3 3 3 3 3 3

2^{6-2} : I=125=1246

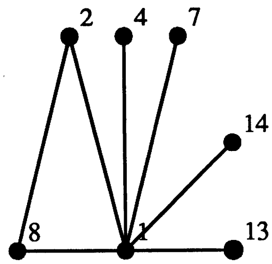
Factor #	1 2 3 4 5 6
Col #	1 2 4 8 3 7



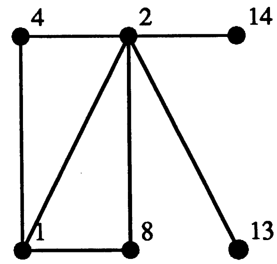
5 3 2 2 1 1

2^{7-3} : I=1235=2346=1347

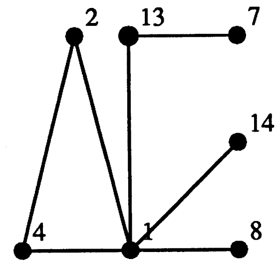
Factor #	1	2	3	4	5	6	7
Col #	1	2	4	8	7	14	13



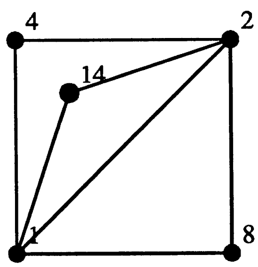
6 2 2 1 1 1 1



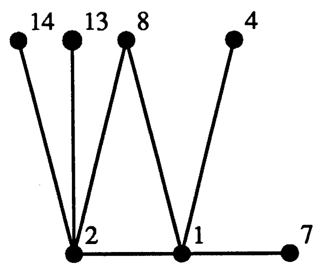
5 3 2 2 1 1



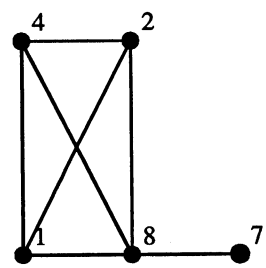
5 2 2 2 1 1 1



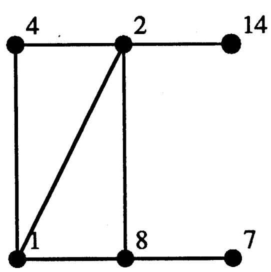
4 4 2 2 2



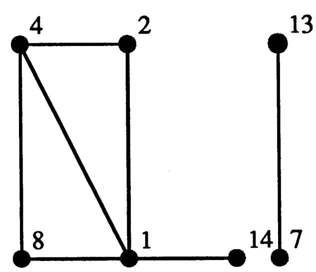
4 4 2 1 1 1 1



4 3 3 3 1

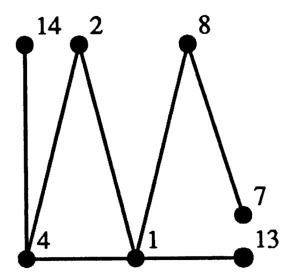


4 3 3 2 1 1



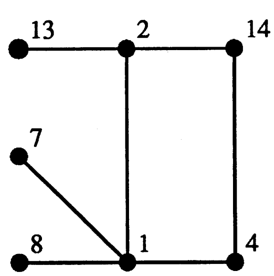
4 3 2 2 1 1 1

8 8 7 7 1 4 1



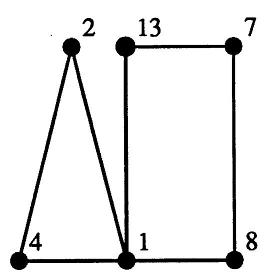
4 3 2 2 1 1 1

8 7 7 5 2 3 4

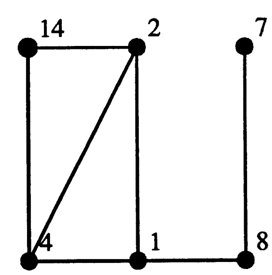


4 3 2 2 1 1 1

7 7 6 5 4 4 3



4 2 2 2 2 2



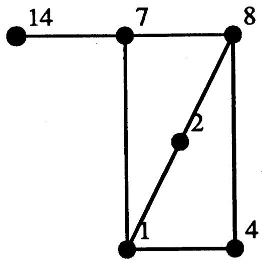
3 3 3 2 2 1

8 8 8 4 6 2

2^{7-3} : I=1235=2346=1347

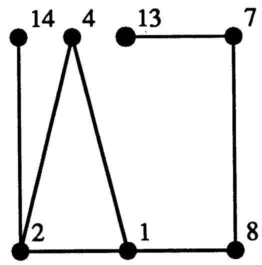
(continued)

Factor #	1 2 3 4 5 6 7
Col #	1 2 4 8 7 14 13



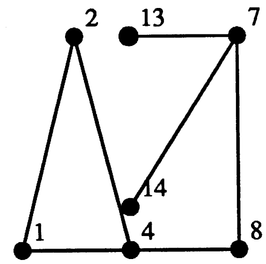
3 3 3 2 2 1

7 7 7 6 6 3



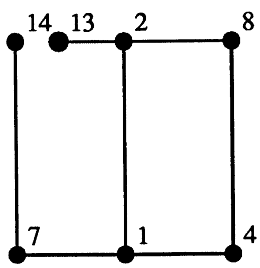
3 3 2 2 2 1 1

7 6 6 5 3 3 2



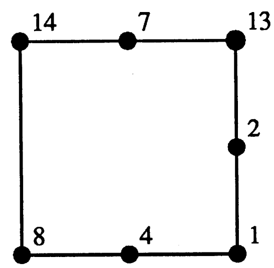
3 3 2 2 2 1 1

6 4 5 6 5 3 3



3 3 2 2 2 1 1

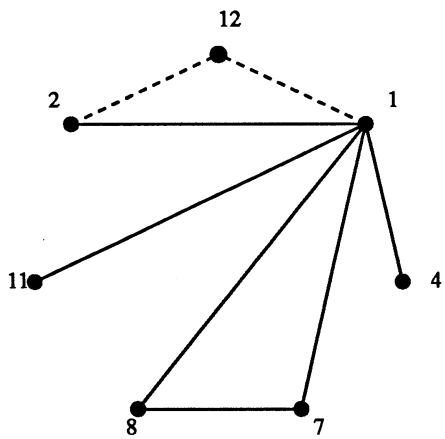
7 6 5 5 4 2 3



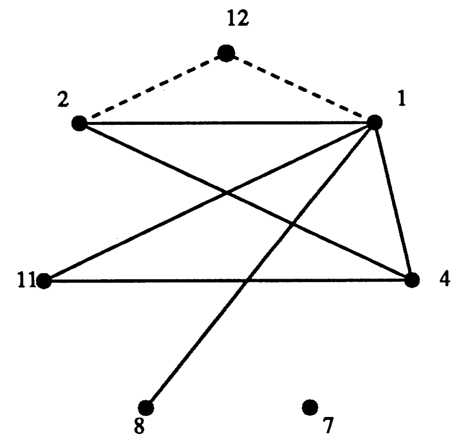
2 2 2 2 2 2 2

2^{7-3} : I=1235=1246=347

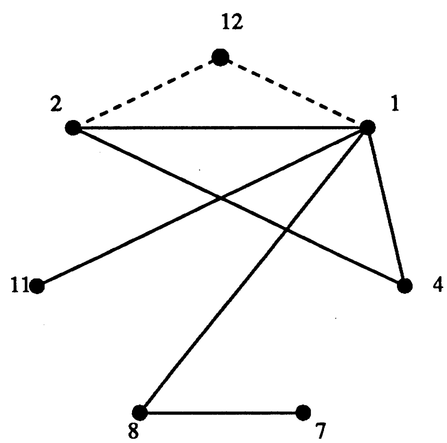
Factor #	1	2	3	4	5	6	7
Col #	1	2	4	8	7	11	12



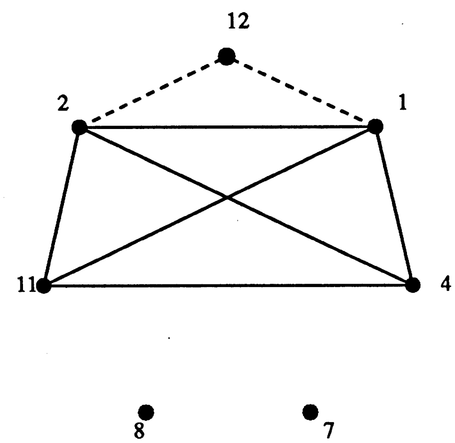
6 2 2 2 2 1 1



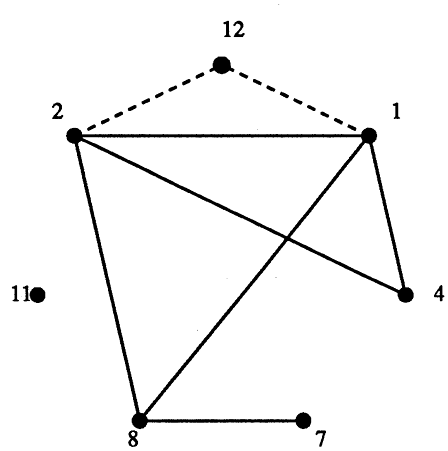
5 3 3 2 2 1 0



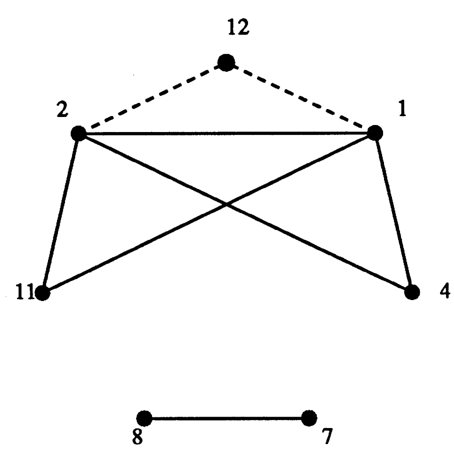
5 3 2 2 2 1 1



4 4 3 3 2 0 0



4 4 3 2 2 1 0



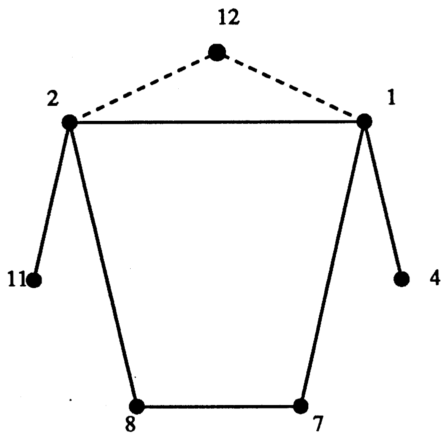
4 4 2 2 2 1 1

10 10 8 8 8 1 1

2^{7-3} : I=1235=1246=347

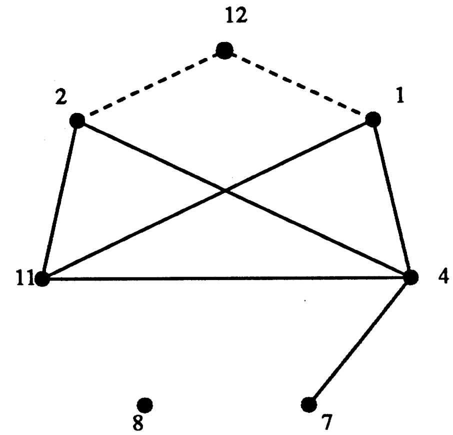
(continued)

Factor #	1	2	3	4	5	6	7
Col #	1	2	4	8	7	11	12

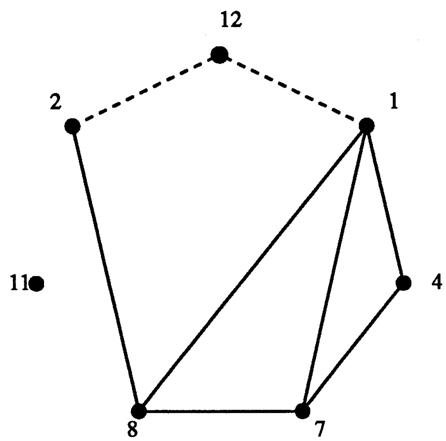


4 4 2 2 2 1 1

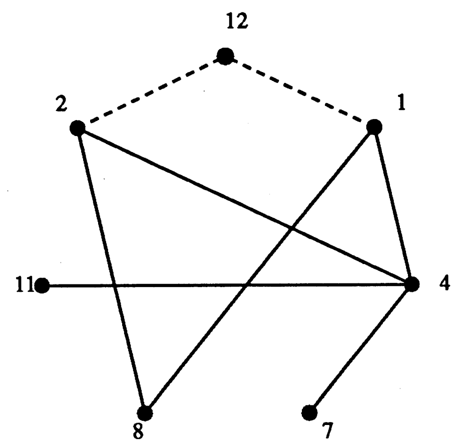
9 9 6 6 8 4 4



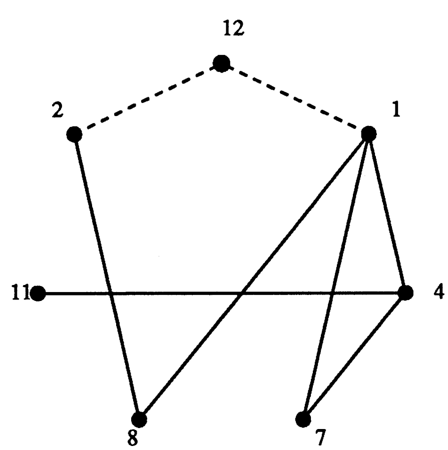
4 3 3 3 2 1 0



4 3 3 2 2 2 0

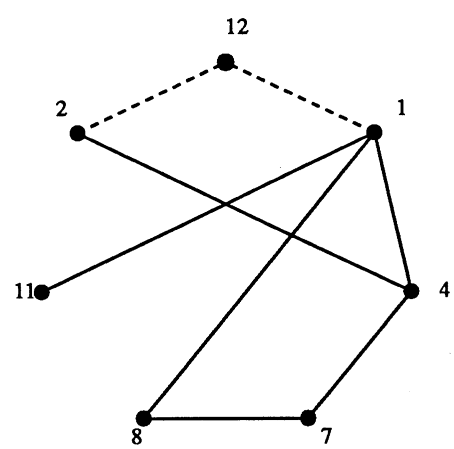


4 3 3 2 2 1 1



4 3 2 2 2 2 1

9 7 4 6 7 6 3



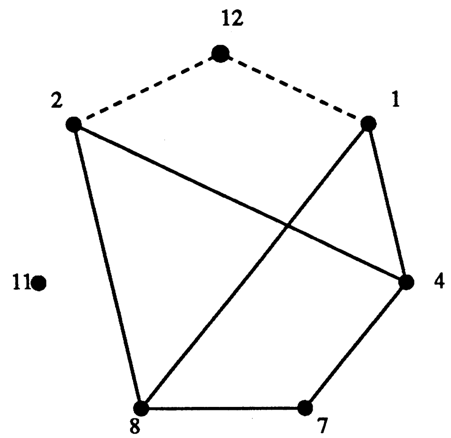
4 3 2 2 2 2 1

8 8 5 6 5 6 4

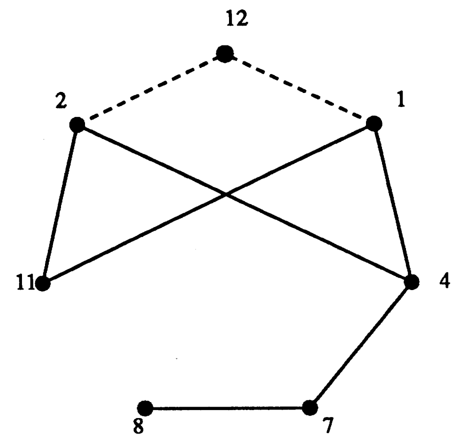
2^{7-3} : I=1235=1246=347

(continued)

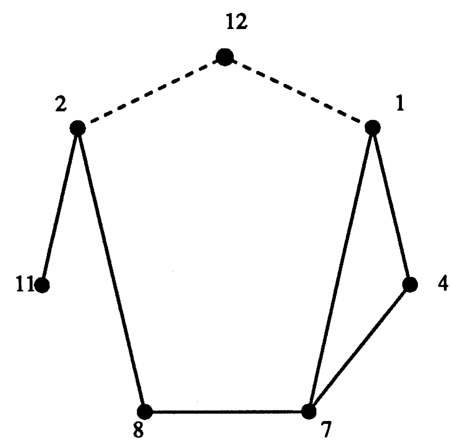
Factor #	1 2 3 4 5 6 7
Col #	1 2 4 8 7 11 12



3 3 3 3 2 2 0



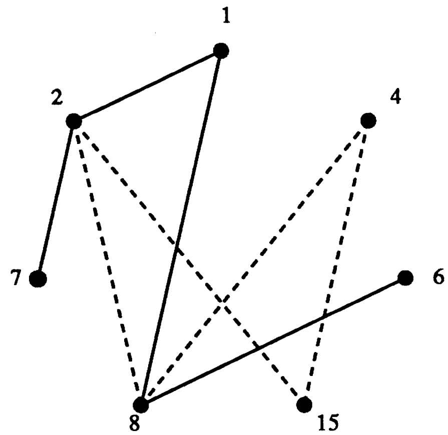
3 3 3 2 2 2 1
7 7 8 4 6 6 2



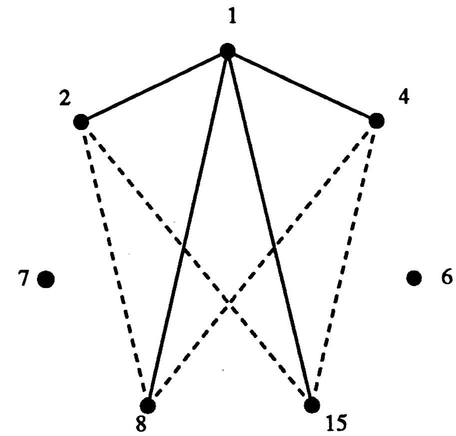
3 3 3 2 2 2 1
7 5 7 6 6 6 3

2^{7-3} : I=12345=236=1237

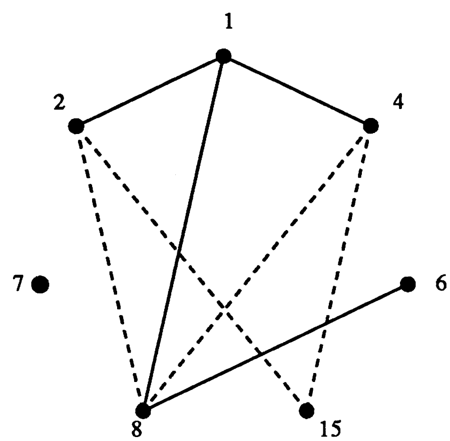
Factor #	1 2 3 4 5 6 7
Col #	1 2 4 8 15 6 7



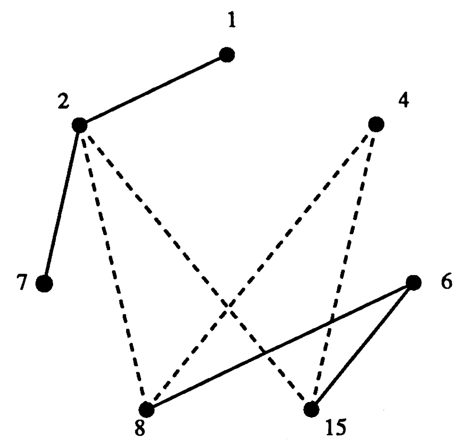
4422211



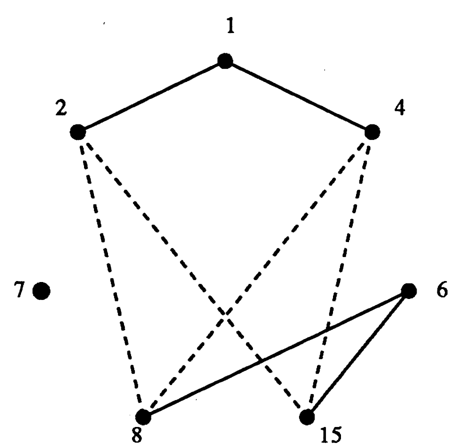
4333300



4333210



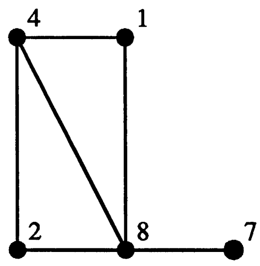
4332211



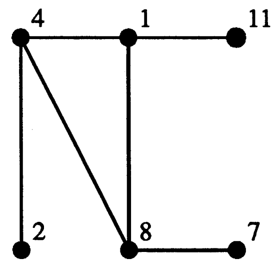
3333220

2^{7-3} : I=1235=1246=127

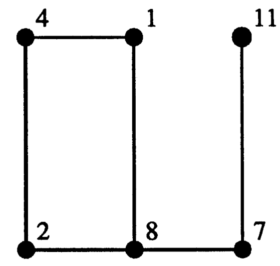
Factor #	1 2 3 4 5 6 7
Col #	1 2 4 8 7 11 3



4 3 2 2 1



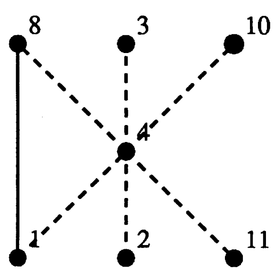
3 3 3 1 1 1



3 2 2 2 2 1

2^{7-3} : I=125=1246=247

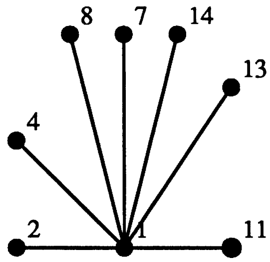
Factor #	1 2 3 4 5 6 7
Col #	1 2 4 8 3 11 10



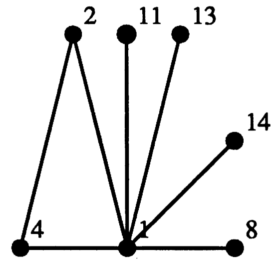
6 2 2 1 1 1 1

2^{8-4} : I=1235=2346=1347=1248

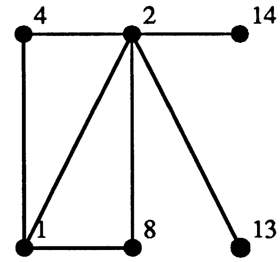
Factor #	1	2	3	4	5	6	7	8
Col #	1	2	4	8	7	14	13	11



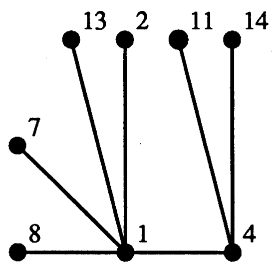
7 1 1 1 1 1 1 1



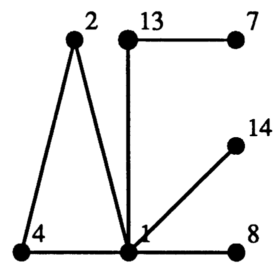
6 2 2 1 1 1 1



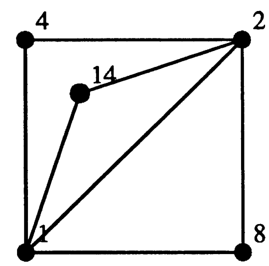
5 3 2 2 1 1



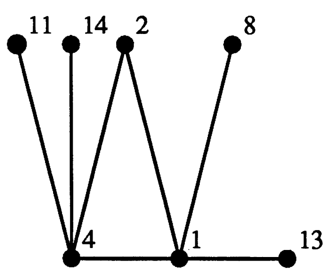
5 3 1 1 1 1 1 1



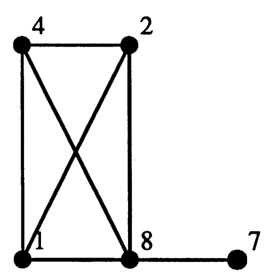
5 2 2 2 1 1 1



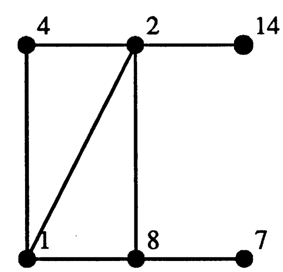
4 4 2 2 2



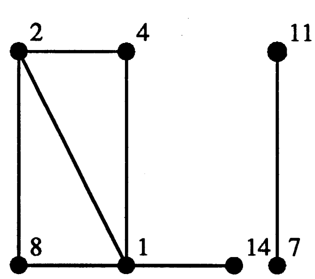
4 4 2 1 1 1 1



4 3 3 3 1

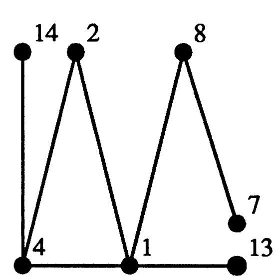


4 3 3 2 1 1



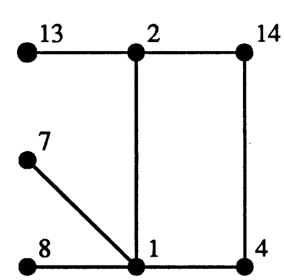
4 3 2 2 1 1 1

8 8 7 7 1 4 1



4 3 2 2 1 1 1

8 7 7 5 2 3 4



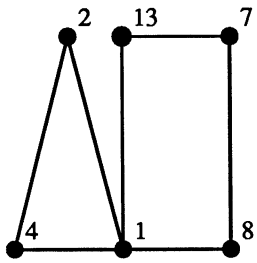
4 3 2 2 1 1 1

7 7 6 5 4 4 3

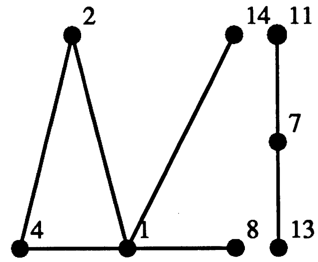
2^{8-4} : I=1235=2346=1347=1248

(continued)

Factor #	1	2	3	4	5	6	7	8
Col #	1	2	4	8	7	14	13	11

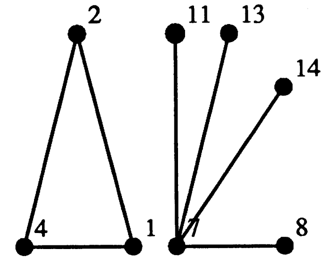


4 2 2 2 2 2



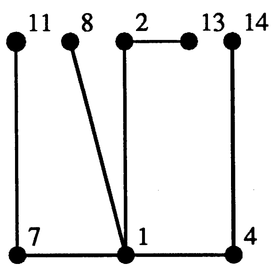
4 2 2 2 1 1 1 1

6 6 6 2 4 4 2 2



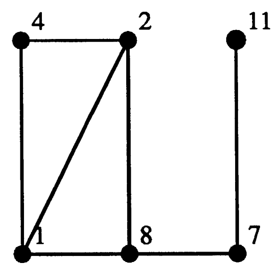
4 2 2 2 1 1 1 1

4 4 4 4 4 4 4 4



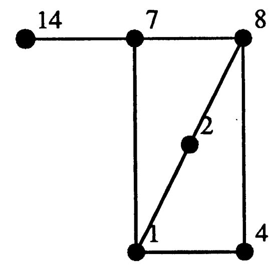
4 2 2 2 1 1 1 1

7 5 5 5 4 2 2 2



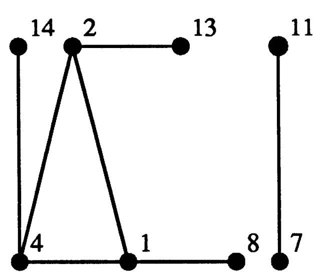
3 3 3 2 2 1

8 8 8 6 4 2



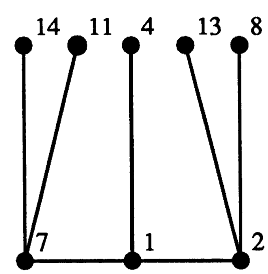
3 3 3 2 2 1

7 7 7 6 6 3



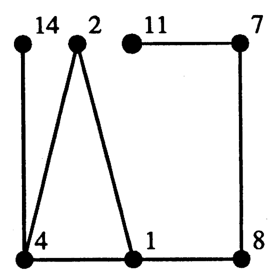
3 3 3 1 1 1 1 1

7 7 7 3 1 3 3 1



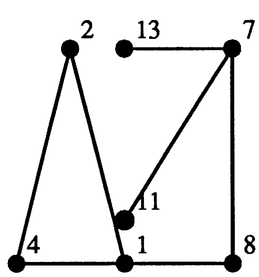
3 3 3 1 1 1 1 1

7 5 5 3 3 3 3 3



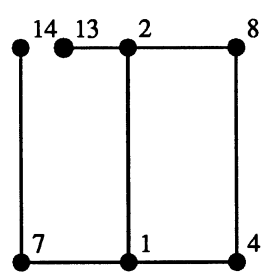
3 3 2 2 2 1 1

7 6 6 5 3 3 2



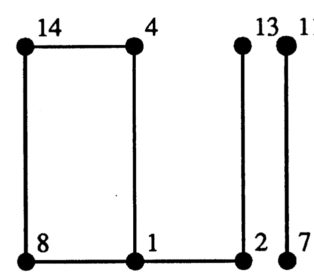
3 3 2 2 2 1 1

6 4 5 6 5 3 3



3 3 2 2 2 1 1

7 6 5 5 4 2 3



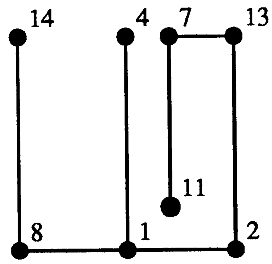
3 2 2 2 2 1 1 1

6 4 5 5 4 1 2 1

2^{8-4} : I=1235=2346=1347=1248

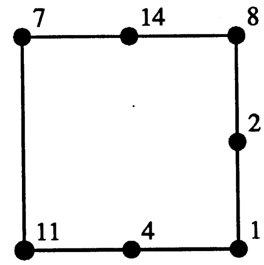
(continued)

Factor #	1	2	3	4	5	6	7	8
Col #	1	2	4	8	7	14	13	11



3 2 2 2 2 1 1 1

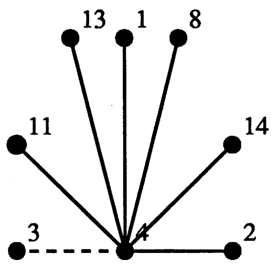
5 5 4 3 4 2 3 2



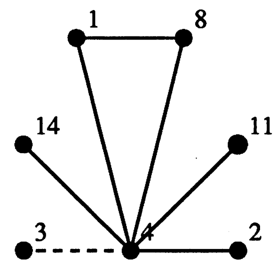
2 2 2 2 2 2 2

2^{8-4} : I=125=2346=1347=1248

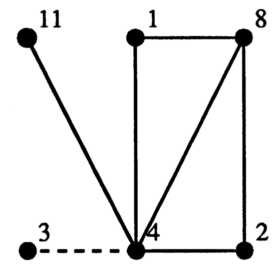
Factor #	1 2 3 4 5 6 7 8
Col #	1 2 4 8 3 14 13 11



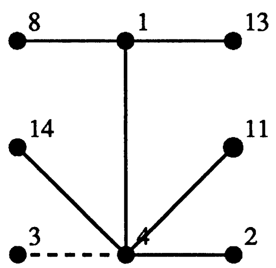
7 1 1 1 1 1 1 1



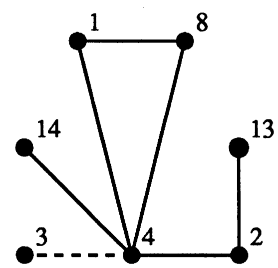
6 2 2 1 1 1 1



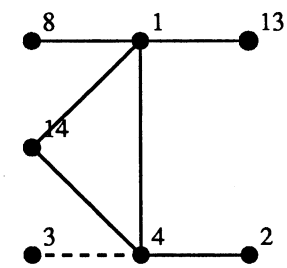
5 3 2 2 1 1



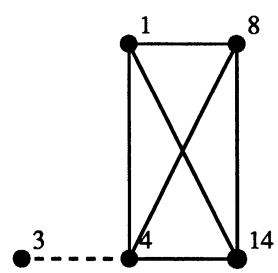
5 3 1 1 1 1 1 1



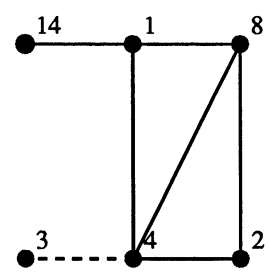
5 2 2 2 1 1 1



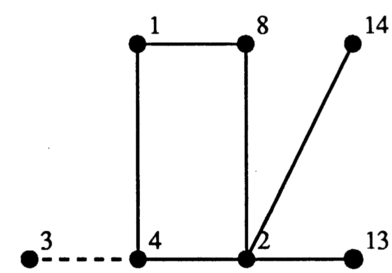
4 4 2 1 1 1 1



4 3 3 3 1

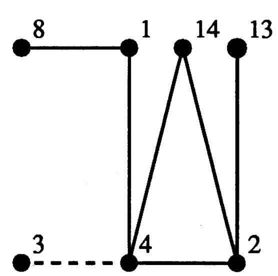


4 3 3 2 1 1



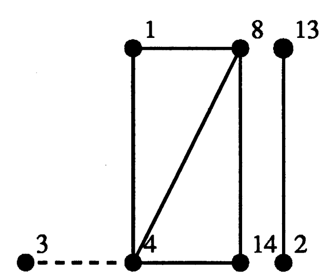
4 3 2 2 1 1 1

7 7 5 6 3 4 4



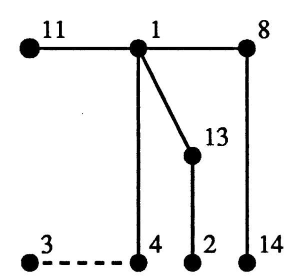
4 3 2 2 1 1 1

8 7 5 7 4 2 3



4 3 2 2 1 1 1

8 8 7 7 4 1 1



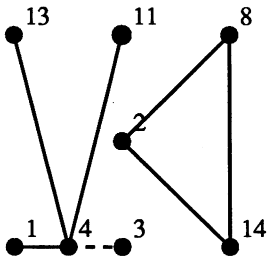
4 2 2 2 1 1 1 1

7 5 5 5 2 2 2 4

2^{8-4} : I=125=2346=1347=1248

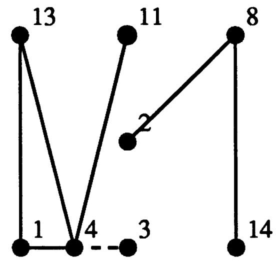
(continued)

Factor #	1	2	3	4	5	6	7	8
Col #	1	2	4	8	3	14	13	11



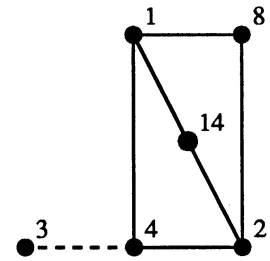
4 2 2 2 1 1 1 1

4 4 4 4 4 4 4 4

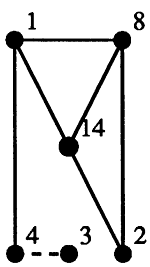


4 2 2 2 1 1 1 1

6 6 2 6 4 2 2 4

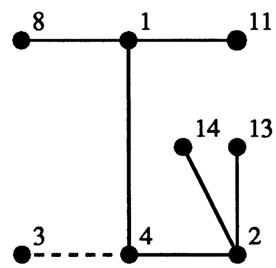


3 3 3 2 2 1



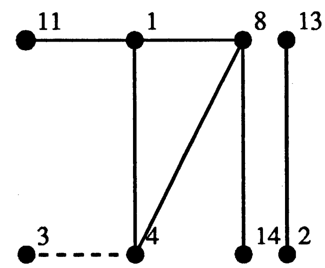
3 3 3 2 2 1

8 8 8 6 4 2



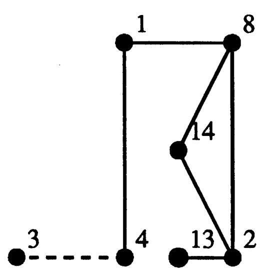
3 3 3 1 1 1 1 1

5 5 7 3 3 3 3 3



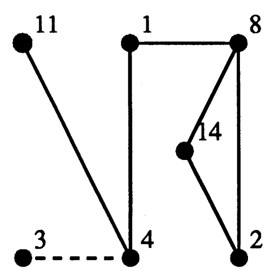
3 3 3 1 1 1 1 1

7 7 7 1 3 3 1 3



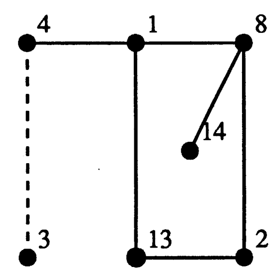
3 3 2 2 2 1 1

6 7 3 5 6 2 3



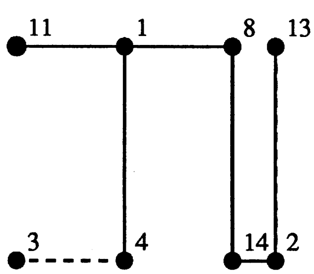
3 3 2 2 2 1 1

4 6 6 5 5 3 3



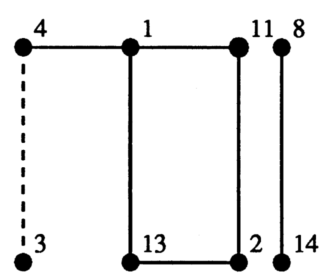
3 3 2 2 2 1 1

7 6 4 5 5 3 2



3 2 2 2 2 1 1 1

5 3 4 5 4 2 2 3

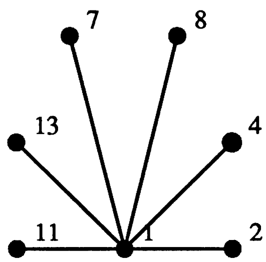


3 2 2 2 2 1 1 1

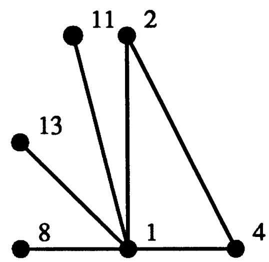
6 4 4 5 5 1 1 2

2^{9-5} : I=1235=2346=1347
=1248=12349

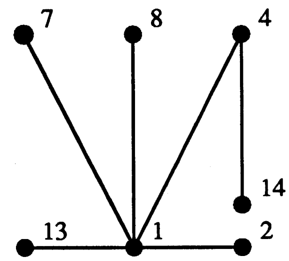
Factor #	1	2	3	4	5	6	7	8	9
Col #	1	2	4	8	7	14	13	11	15



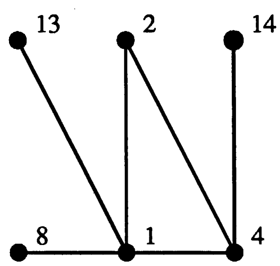
6111111



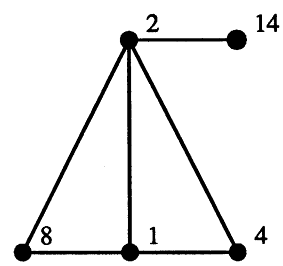
522111



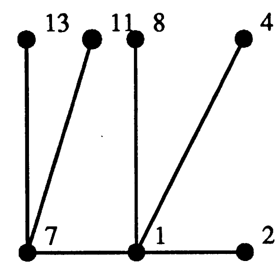
5211111



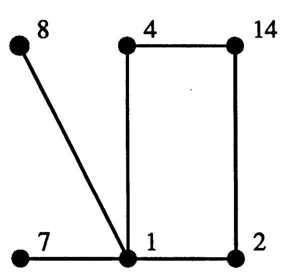
432111



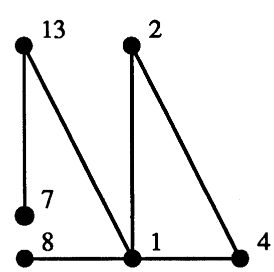
43221



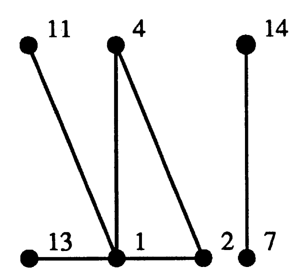
4311111



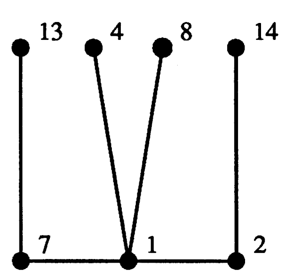
422211
666444



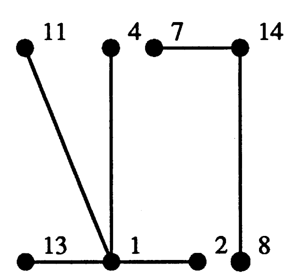
422211
766542



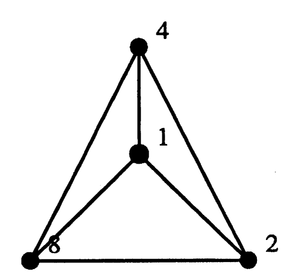
4221111
6664411



4221111
6554422



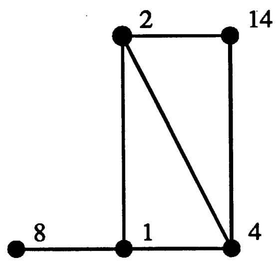
42111111



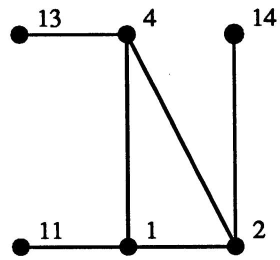
3333

2^{9-5} : I=1235=2346=1347
(continued) =1248=12349

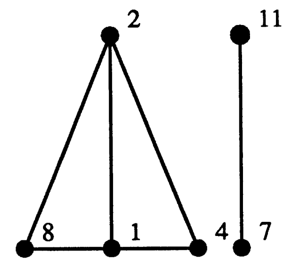
Factor #	1	2	3	4	5	6	7	8	9
Col #	1	2	4	8	7	14	13	11	15



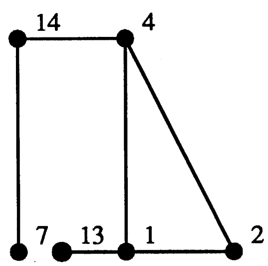
33321



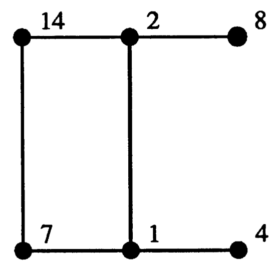
333111



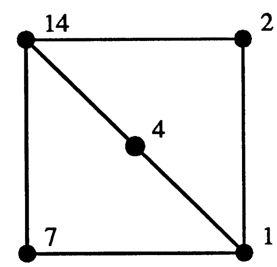
332211
776611



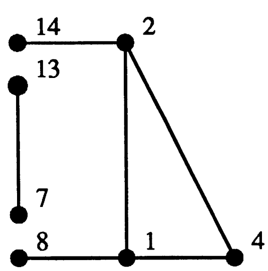
332211
766432



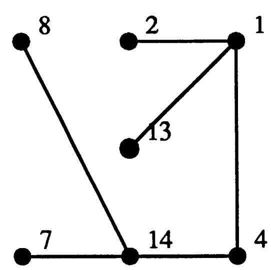
332211
665533



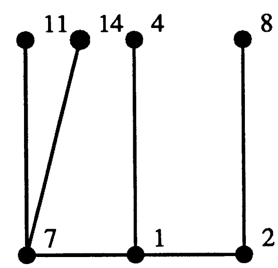
33222



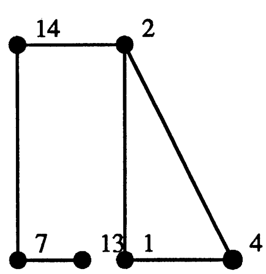
3321111
6663311



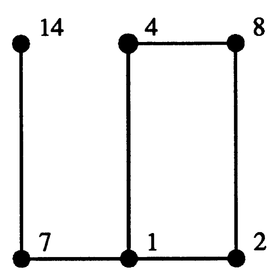
3321111
4463333



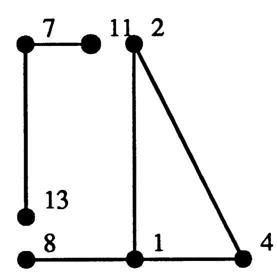
3321111
6543332



322221
555532



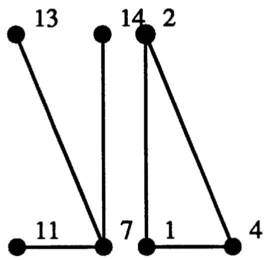
322221
655442



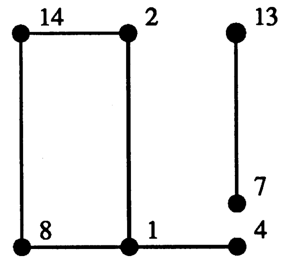
3222111
5552322

2^{9-5} : I=1235=2346=1347
(continued) =1248=12349

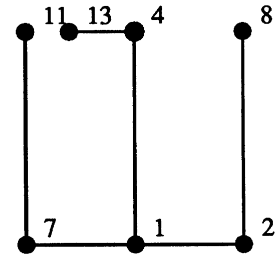
Factor #	1	2	3	4	5	6	7	8	9
Col #	1	2	4	8	7	14	13	11	15



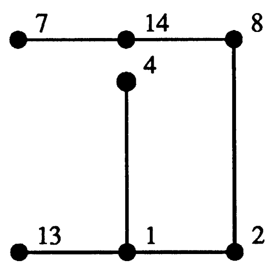
3222111
3444333



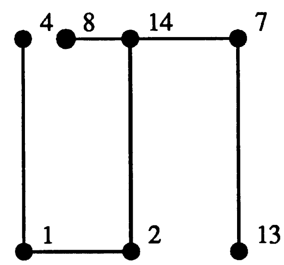
3222111
5554311



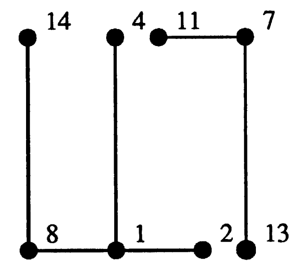
3222111
6444222



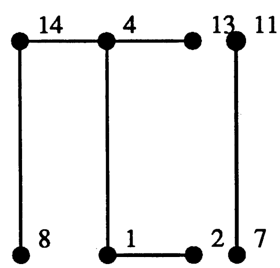
3222111
4543332



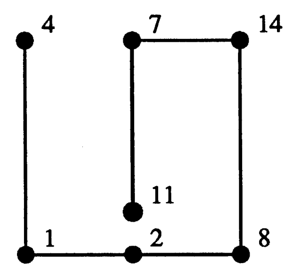
3222111
5543322



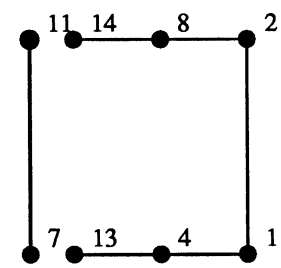
32211111
44233222



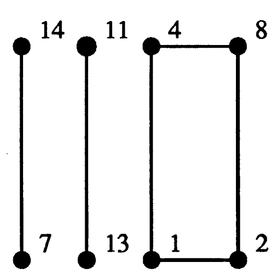
32211111
54432211



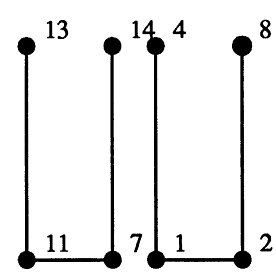
2222211



22221111
44332211



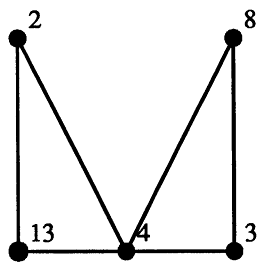
22221111
44441111



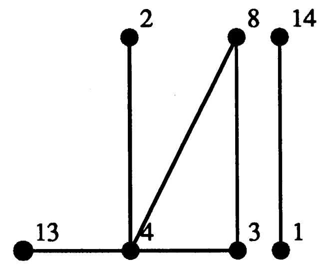
22221111
33332222

2^{9-5} : I=125=2346=1347
=248=139

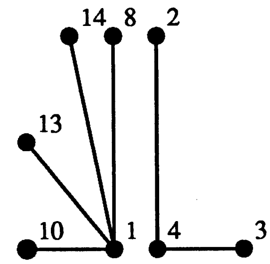
Factor #	1	2	3	4	5	6	7	8	9
Col #	1	2	4	8	3	14	13	10	5



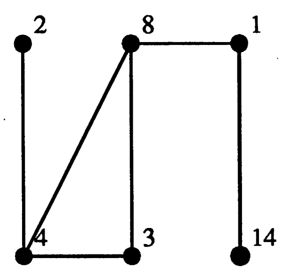
4 2 2 2 2



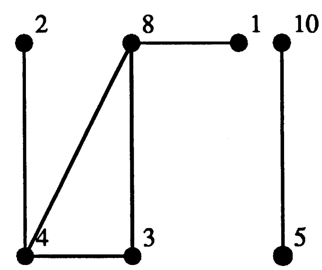
4 2 2 1 1 1 1



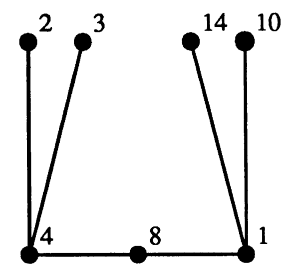
4 2 1 1 1 1 1 1



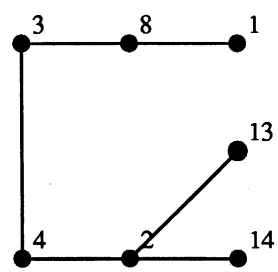
3 3 2 2 1 1



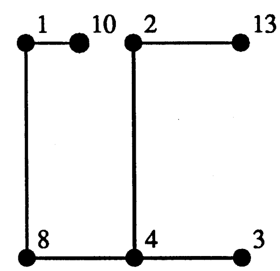
3 3 2 1 1 1 1
6 6 6 3 3 1 1



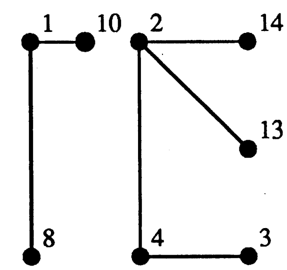
3 3 2 1 1 1 1
4 4 6 3 3 3 3



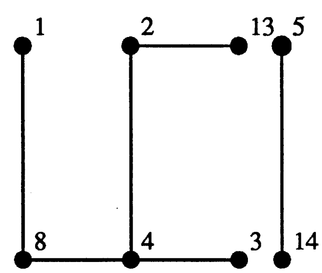
3 2 2 2 1 1 1
4 5 3 4 2 3 3



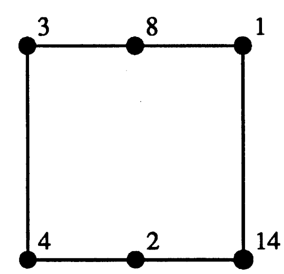
3 2 2 2 1 1 1
5 4 3 5 3 2 2



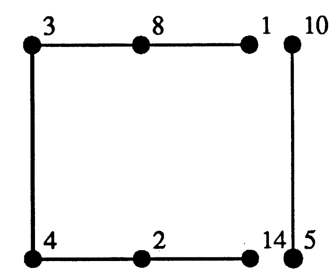
3 2 2 1 1 1 1 1
4 2 4 2 2 3 3 2



3 2 2 1 1 1 1 1
5 4 4 2 3 1 2 1



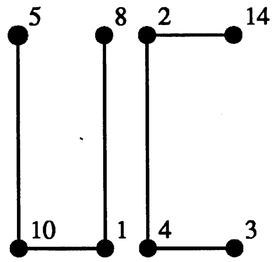
2 2 2 2 2 2



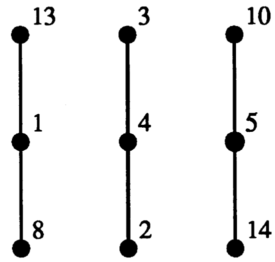
2 2 2 2 1 1 1 1
3 4 3 4 2 2 1 1

2^{9-5} : I=125=2346=1347
(continued) =248=139

Factor #	1	2	3	4	5	6	7	8	9
Col #	1	2	4	8	3	14	13	10	5



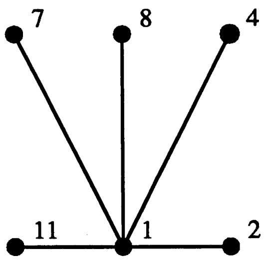
2 2 2 2 1 1 1 1
3 3 3 3 2 2 2 2



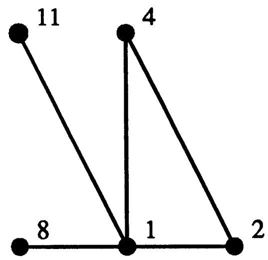
2 2 2 1 1 1 1 1 1

2^{10-6}: I=1235=2346=1347=1248 =12349=$34t_0$

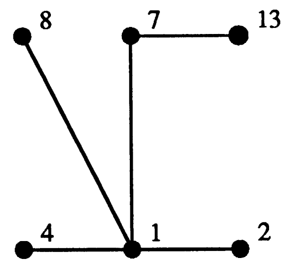
Factor #	1	2	3	4	5	6	7	8	9	t_0
Col #	1	2	4	8	7	14	13	11	15	12



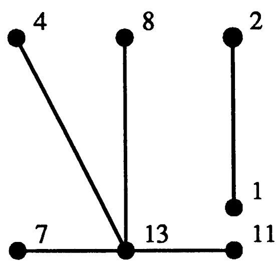
511111



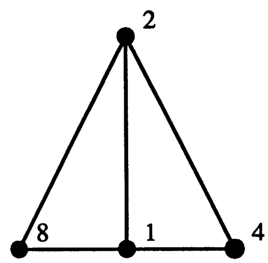
42211



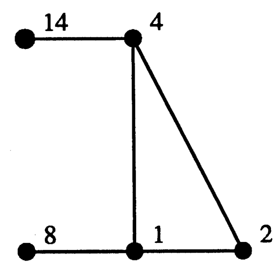
421111



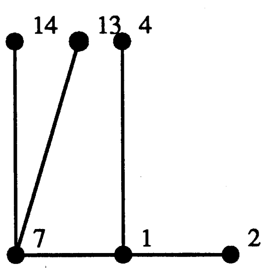
4111111



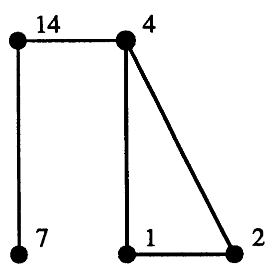
3322



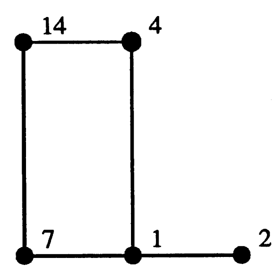
33211



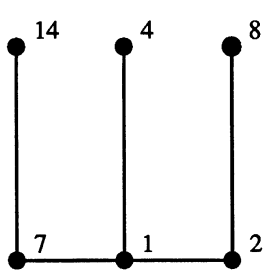
331111



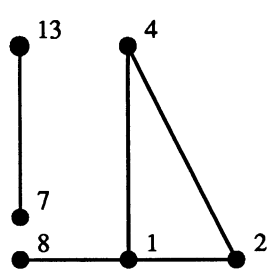
32221
65542



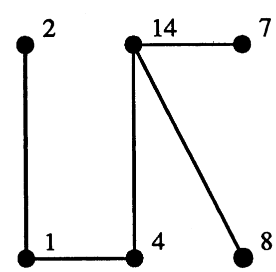
32221
55543



322111
544322



322111
555311

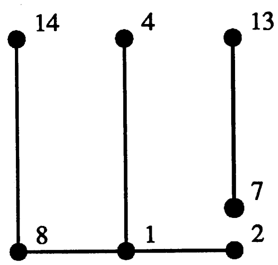


322111

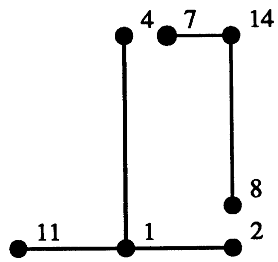
2^{10-6}: I=1235=2346=1347=1248 =12349=$34t_0$

(continued)

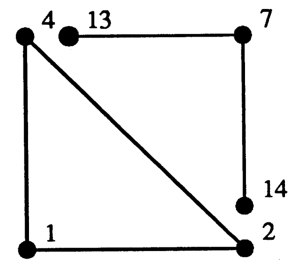
Factor #	1	2	3	4	5	6	7	8	9	t_0
Col #	1	2	4	8	7	14	13	11	15	12



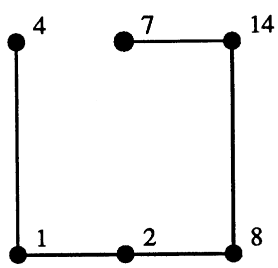
3211111
4433211



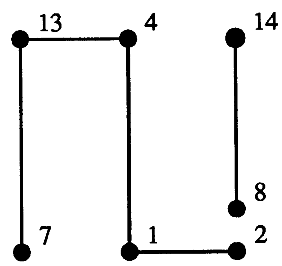
3211111
3233322



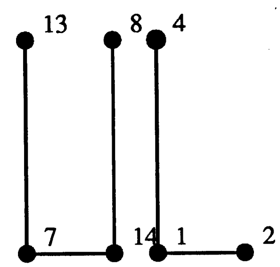
222211
444222



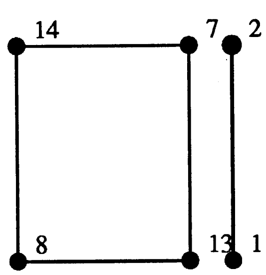
222211
443322



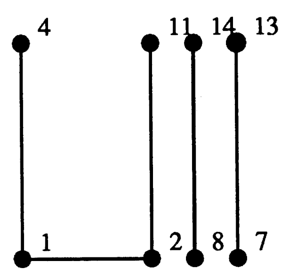
2221111
4332211



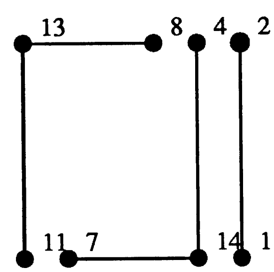
2221111
3322222



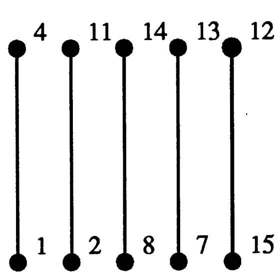
222211



22111111
33221111



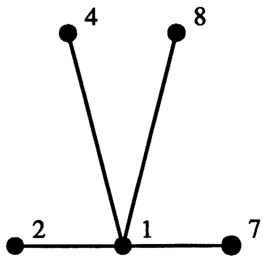
22111111
22222211



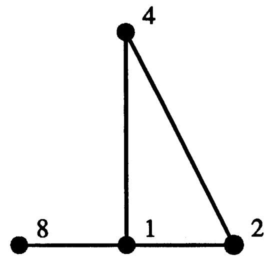
1111111111

2^{11-7}: I=1235=2346=1347=1248 =12349=$34t_0$=$24t_1$

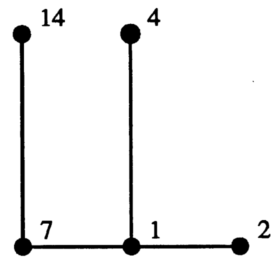
Factor #	1	2	3	4	5	6	7	8	9	t_0	t_1
Col #	1	2	4	8	7	14	13	11	15	12	10



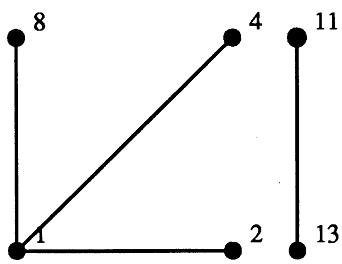
41111



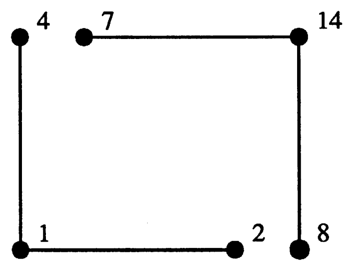
3221



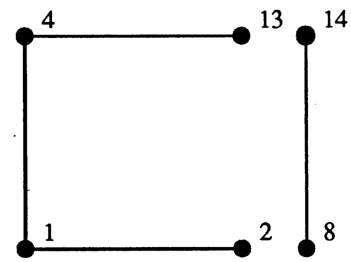
32111



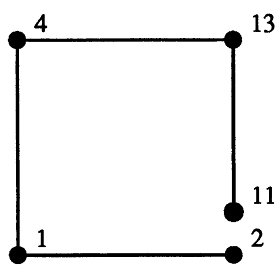
31111



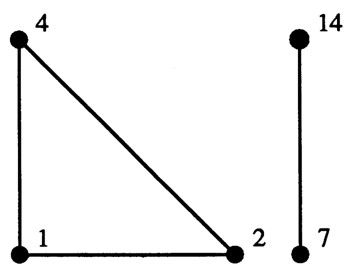
221111
222222



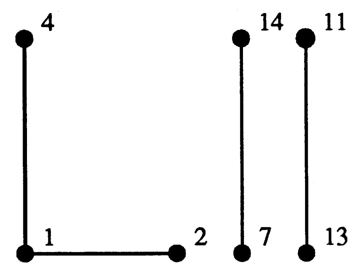
221111
332211



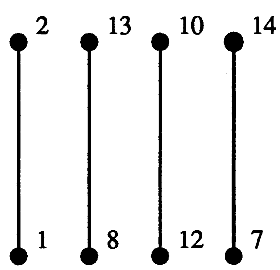
22211
43322



22211
44411



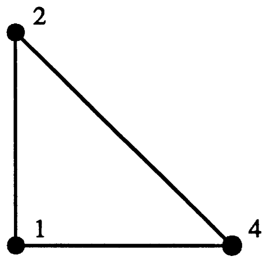
2111111



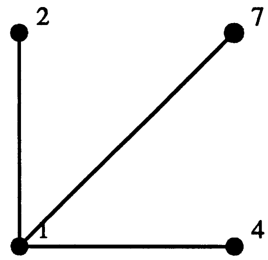
11111111

2^{12-8}: $I=1235=2346=1347=1248$
$=12349=34t_0=24t_1=14t_2$

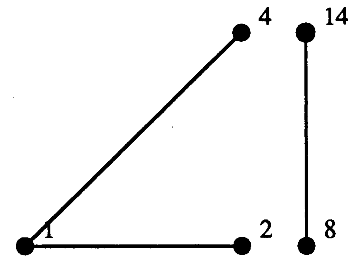
Factor #	1	2	3	4	5	6	7	8	9	t_0	t_1	t_2
Col #	1	2	4	8	7	14	13	11	15	12	10	9



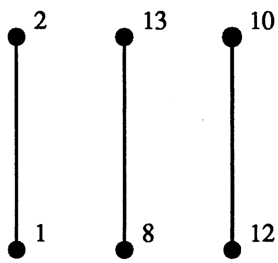
322000



311100



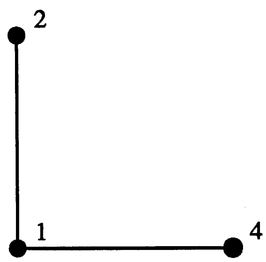
211110



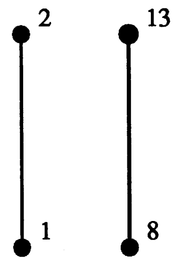
111111

2^{13-9}: $I=1235=2346=1347=1248$
$=12349=34t_0=24t_1=14t_2=23t_3$

Factor #	1	2	3	4	5	6	7	8	9	t_0	t_1	t_2	t_3
Col #	1	2	4	8	7	14	13	11	15	12	10	9	6



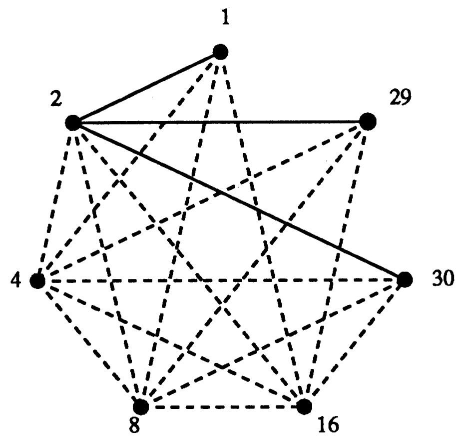
211



1111

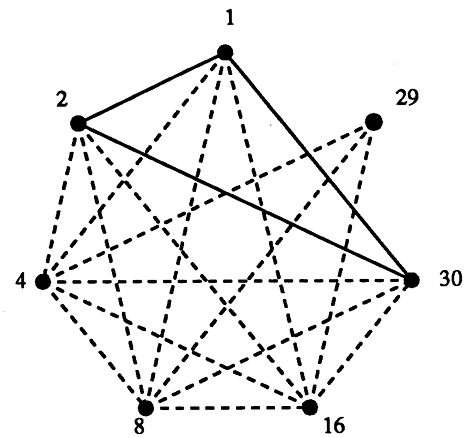
2^{7-2} : I=23456=13457

Factor #	1	2	3	4	5	6	7
Col #	1	2	4	8	16	29	30



6666444

all 2fi's except 16, 17, 67

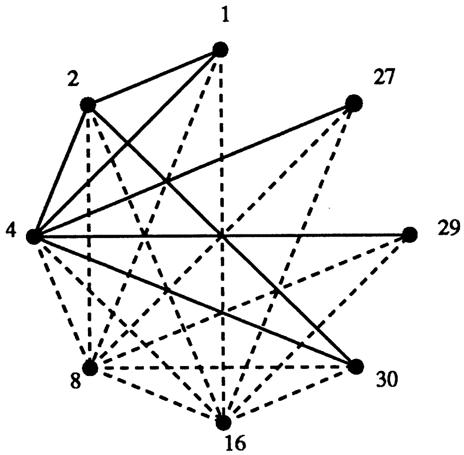


6665553

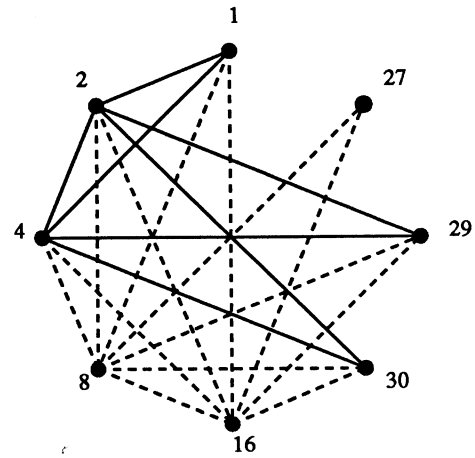
all 2fi's except 12, 26, 27

2^{8-3} : I=23456=13457=12458

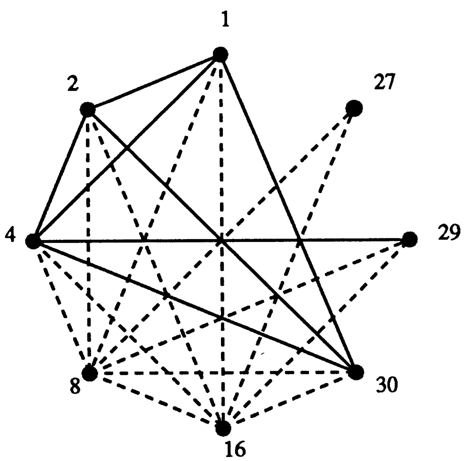
Factor #	1 2 3 4 5 6 7 8
Col #	1 2 4 8 16 29 30 27



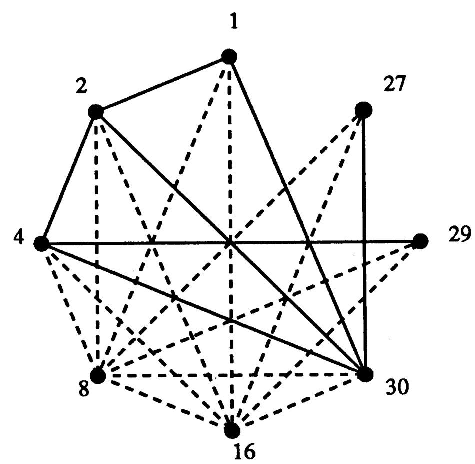
7 7 7 5 4 4 3 3



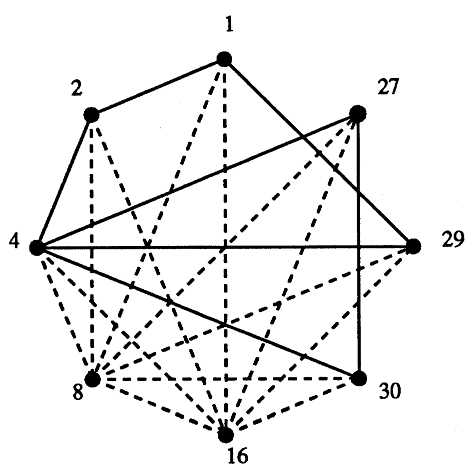
7 7 6 6 4 4 4 2



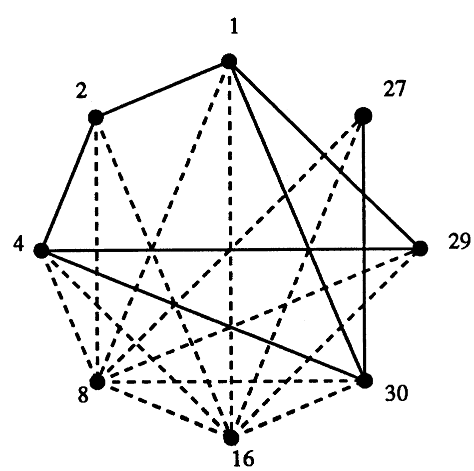
7 7 6 5 5 5 3 2



7 7 6 5 5 4 3 3



7 7 6 4 4 4 4 4



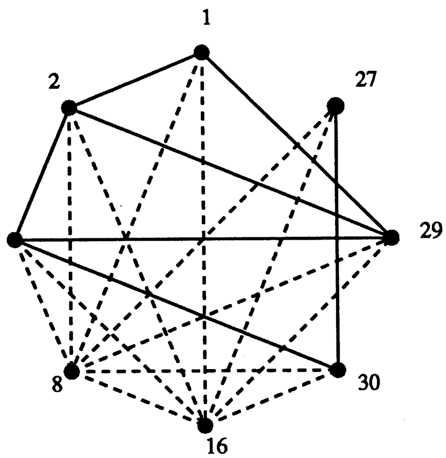
7 7 5 5 5 4 4 3

33 33 27 27 27 24 24 19

2^{8-3} : I=23456=13457=12458

(continued)

Factor #	1	2	3	4	5	6	7	8
Col #	1	2	4	8	16	29	30	27

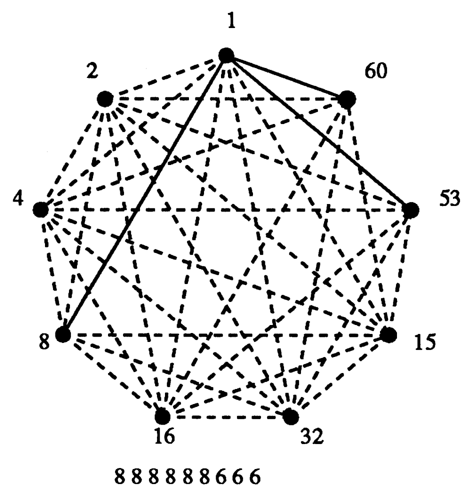


7 7 5 5 5 4 4 3

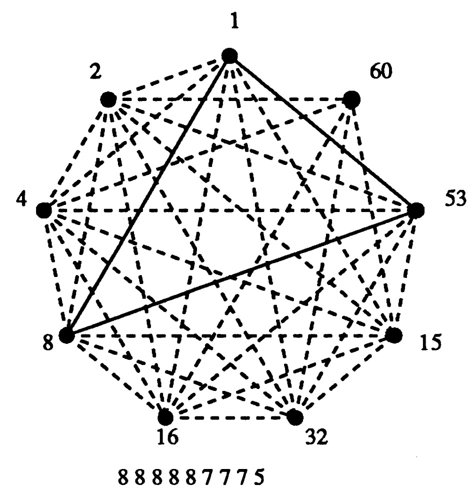
33 33 28 28 28 22 24 18

2^{9-3} : I=12347=13568=34569

Factor #	1	2	3	4	5	6	7	8	9
Col #	1	2	4	8	16	32	15	53	60



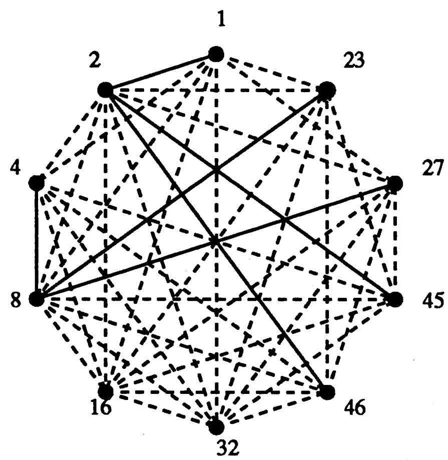
all 2fi's except 48, 49, 89



all 2fi's except 19, 49, 89

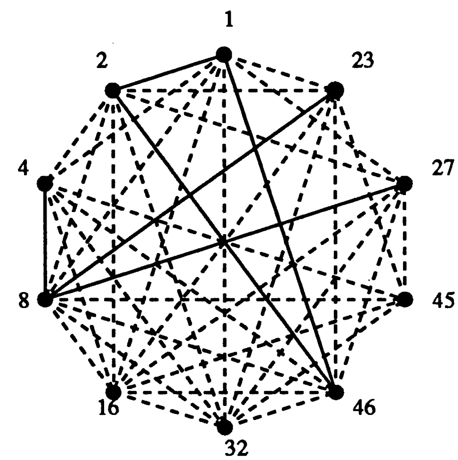
2^{10-4}: I=23467=13468
=12459=1235t_0

Factor #	1	2	3	4	5	6	7	8	9	t_0
Col #	1	2	4	8	16	32	46	45	27	23



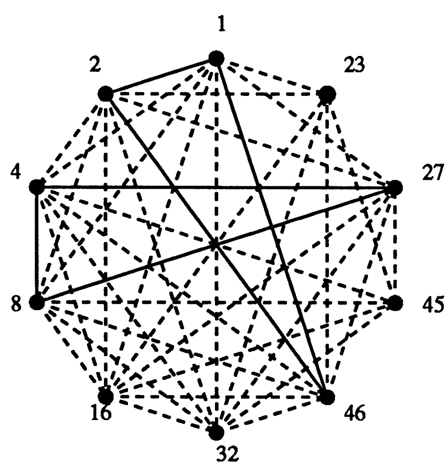
9999777777

all 2fi's except 17 18 39 3t_0 78 9t_0



9998887776

all 2fi's except 18 28 39 3t_0 78 9t_0

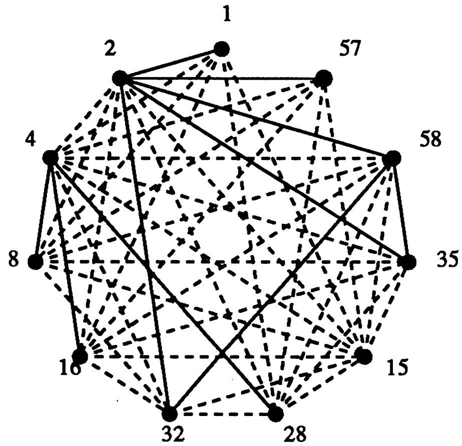


9988888866

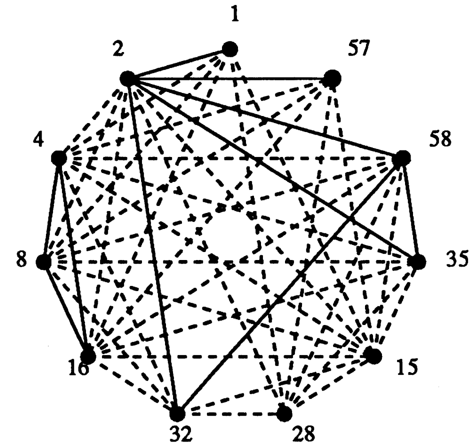
all 2fi's except 18 28 3t_0 4t_0 78 9t_0

2^{11-5}: $I=3457=12348=1269$
$=2456t_0=1456t_1$

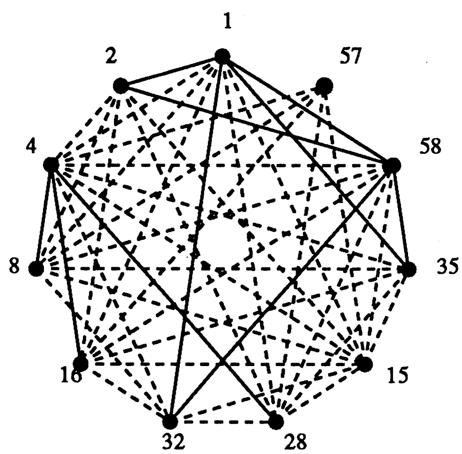
Factor #	1	2	3	4	5	6	7	8	9	t_0	t_1
Col #	1	2	4	8	16	32	28	15	35	58	57



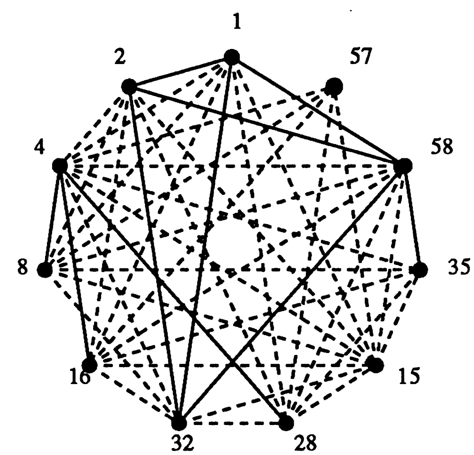
10 10 10 8 8 8 8 7 7 6 6



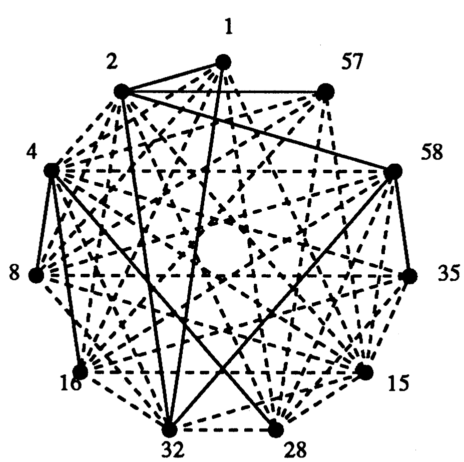
10 10 9 9 9 8 7 7 7 6 6



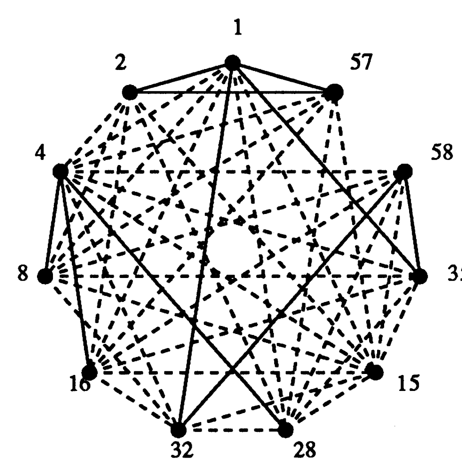
10 10 9 9 8 8 8 7 7 7 5



10 10 9 8 8 8 8 8 8 6 5



10 10 9 8 8 8 8 8 7 6 6

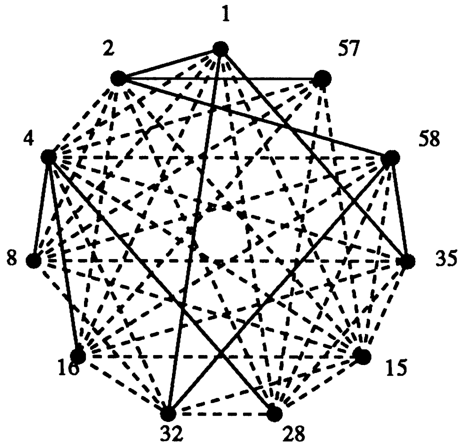


10 10 9 8 8 8 7 7 7 7 7

2^{11-5}: I=3457=12348=1269=2456t_0=1456t_1

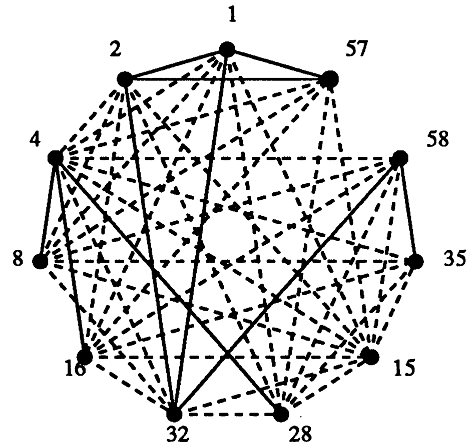
(continued)

Factor #	1	2	3	4	5	6	7	8	9	t_0	t_1
Col #	1	2	4	8	16	32	28	15	35	58	57



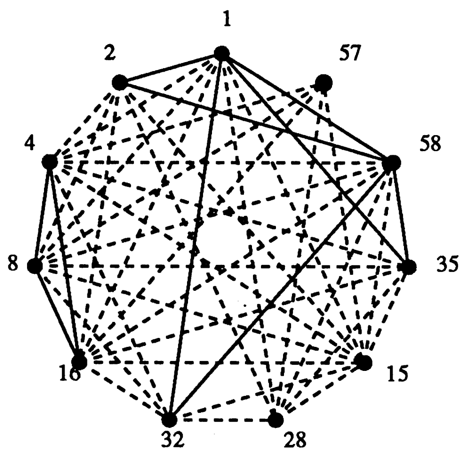
10 10 8 8 8 8 8 8 7 7 6

78 78 66 64 64 64 66 66 60 60 52

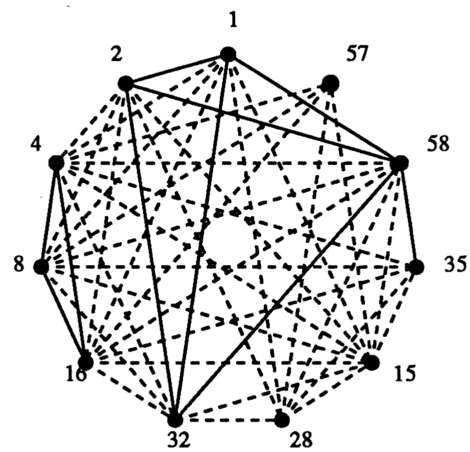


10 10 8 8 8 8 8 8 7 7 6

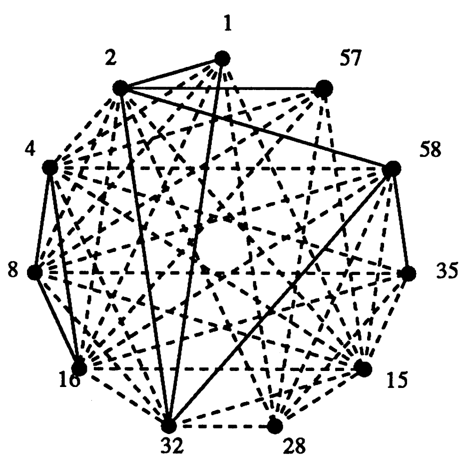
78 78 67 64 64 67 64 67 58 60 51



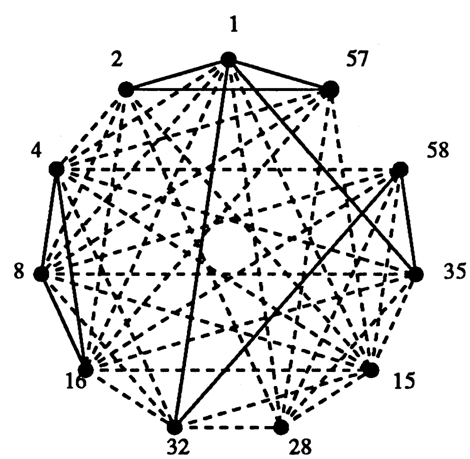
10 9 9 9 9 9 7 7 7 7 5



10 9 9 9 9 8 8 8 7 6 5



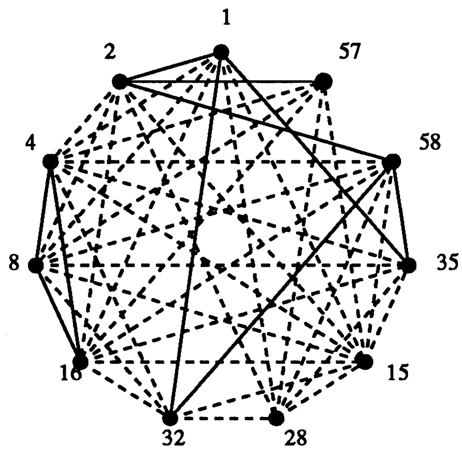
10 9 9 9 9 8 8 7 7 6 6



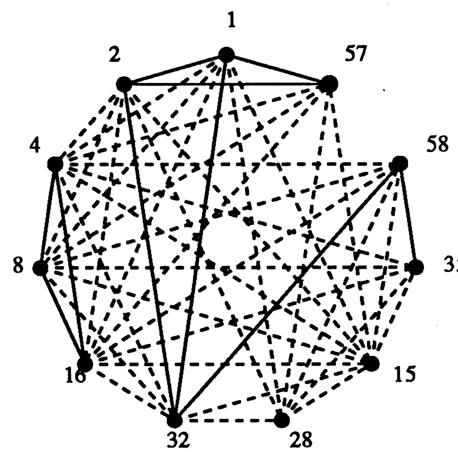
10 9 9 9 9 7 7 7 7 7 7

2^{11-5}: I=3457=12348=1269
(continued) =2456t_0=1456t_1

Factor #	1	2	3	4	5	6	7	8	9	t_0	t_1
Col #	1	2	4	8	16	32	28	15	35	58	57



10 9 9 9 8 8 8 7 7 7 6
78 72 72 72 66 66 66 60 60 54 52



10 9 9 9 8 8 8 7 7 7 6
78 72 72 72 67 67 67 54 58 60 51