Discrete Applied Mathematics 9 (1984) 251-261
North-Holland

APPLICATIONS OF RAMSEY THEORY

Fred S. ROBERTS*
Department of Mathematics, Rutgers University, New Brunswick, NJ 08903, USA

Received 22 June 1983
Revised 14 October 1983

This paper attempts to show that Ramsey theory really does have useful applications, by presenting four applications from the literature. The applications are from the fields of communications, information retrieval in computer science, and decisionmaking.

1. Introduction

Ramsey theory is very interesting. But what good is it?

In recent years, there has been a great deal of interest in Ramsey theory. Two major books on the subject have appeared, the books by Graham [10] and by Graham, et al. [11]. There was also a special issue on Ramsey theory in the *Journal of Graph Theory* (Vol. 7, No. 1, Spring 1983). However, little has been written about the applications of the subject. In this paper, we attempt to show that Ramsey theory really does have useful applications. We present four examples from the literature to make this point. The first two applications involve communications, the third is to a problem of information retrieval in computer science, and the fourth is to a problem in decisionmaking.

We shall adopt the graph-theoretic and Ramsey-theoretic notation and terminology of Roberts [17]. In particular, $R(p_1, p_2, \dots, p_t; r)$ is the smallest integer N with the property that whenever S is a set of N elements and we divide the r-element subsets of S into t sets, $X_1, X_2, \dots, X_t$, then for some i, there is a p_i-element subset of S all of whose r-element subsets are in X_i. $R(p,q)$ is $R(p,q;2)$. Finally if $G_1, G_2, \dots, G_t$ are graphs, $R(G_1, G_2, \dots, G_t)$ is the smallest N with the property that every coloring of the edges of the complete graph K_N in the t colors $1, 2, \dots, t$ gives rise, for some i, to a subgraph that is isomorphic to G_i and is colored all in color i, that is, to a monochromatic G_i.

*This expository paper is a slightly modified version of a section of: Fred S. Roberts, *Applied Combinatorics*, a 1984 Prentice-Hall publication. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

2. Confusion graphs for noisy channels

In communication theory, a *noisy channel* gives rise to a *confusion graph,* a graph whose vertices are elements of a transmission alphabet T and which has an edge between two letters of T if and only if, when sent over the channel, they can be received as the same letter. Given a noisy channel, we would like to make errors impossible by choosing a set of signals that can be unambiguously received, that is, so that no signal in the set is confusable with another signal in the set. This corresponds to choosing an independent set in the confusion graph G. In the confusion graph G of Fig. 1 the largest independent set consists of two vertices. Thus, we may choose two such letters, say a and c, and use these as an *unambiguous code alphabet* for sending messages. In general, the largest unambiguous code alphabet has $\alpha(G)$ elements, where $\alpha(G)$ is the size of the largest independent set in G.

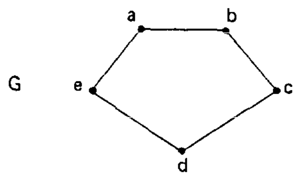


Fig. 1. Confusion graph.

To see whether we can find a better unambiguous code alphabet, we shall introduce the notion of *normal product* $G \cdot H$ of two graphs G and H. This is defined as follows. The vertices are the pairs in the Cartesian product $V(G) \times V(H)$. There is an edge between (a,b) and (c,d) if and only if one of the following holds:

(i) $\{a,c\} \in E(G)$ and $\{b,d\} \in E(H)$,

(ii) $a=c$ and $\{b,d\} \in E(H)$,

(iii) $b=d$ and $\{a,c\} \in E(G)$.

(The term normal product is used by Berge [3]; another term in use for this is *strong product.*) Fig. 2 shows a normal product.

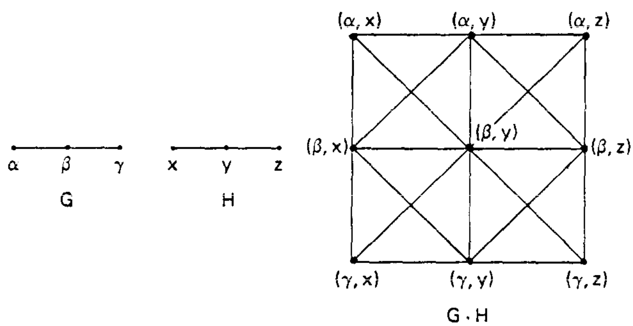


Fig. 2. The normal product of two graphs.

We can find a larger unambiguous code alphabet by allowing combinations of letters from the transmission alphabet to form the code alphabet. For example, suppose that we consider all possible ordered pairs of elements from the transmission alphabet T, or strings of two elements from T. Then under the confusion graph of Fig. 1, we can find four such ordered pairs, aa, ac, ca, and cc, none of which can be confused with any of the others. In general, two strings of letters from the transmission alphabet can be confused if and only if they can be received as the same string. In this sense, strings aa and ac cannot be confused, since a and c cannot be received as the same letter. We can draw a new confusion graph whose vertices are strings of length two from T. This graph has the following property: Strings xy and uv can be confused if and only if one of the following holds:

(i) x and u can be confused and y and v can be confused,
(ii) $x=u$ and y and v can be confused,
(iii) $y=v$ and x and u can be confused.

In terms of the original confusion graph G, the new confusion graph is the normal product $G\cdot G$.

If G is the confusion graph of Fig. 1, we have already observed that one independent set or unambiguous code alphabet in $G\cdot G$ can be found by using the strings aa, ac, ca, and cc. However, there is a larger independent set that consists of the strings aa, bc, ce, db, and ed. What is the largest independent set in $G\cdot G$? The following theorem can be used to help answer this question.

Theorem 1 (Hedrlín [13]). *If G and H are any graphs, then*

$$\alpha(G\cdot H)\leq R(\alpha(G)+1,\alpha(H)+1)-1.$$

Proof. Let $N=R(\alpha(G)+1,\ \alpha(H)+1)$. Suppose that $\alpha(G\cdot H)\geq N$. We reach a contradiction. Let I be an independent set of $G\cdot H$ with N vertices. Suppose that (a,b) and (c,d) are two distinct vertices in I. Since I is independent,

either (a) $a\neq c$ and $\{a,c\}\notin E(G)$,
or (b) $b\neq d$ and $\{b,d\}\notin E(H)$.

Consider a complete graph with vertex set the N vertices of I. Color an edge (a,b) to (c,d) of this graph blue if (a) holds and red otherwise. This is a coloring of the edges of the complete graph K_N in two colors, blue and red. By choice of N, either there is a blue clique C with $\alpha(G)+1$ vertices or a red clique D with $\alpha(H)+1$ vertices. In the former case, note that $(a,b)\in C$ and $(c,d)\in C$ implies that (a) holds, and hence $\{a\colon a\in V(G)$ and $(a,b)\in C$ for some $b\}$ is an independent set of G with $\alpha(G)+1$ vertices. This is a contradiction. In the latter case, $\{b\colon b\in V(H)$ and $(a,b)\in D$ for some $a\}$ is an independent set of H with $\alpha(H)+1$ vertices, again a contradiction. We conclude that $\alpha(G\cdot H)\leq N-1$. □

As a corollary of this result, note that if $G=Z_5$ is the confusion graph of Fig. 1 then

$$\alpha(G\cdot G) \leq R(3,3) - 1 = 5.$$

Hence, we have found a largest independent set here.

Going beyond strings of length 2, we can seek strings of length k from the transmission alphabet, and seek independent sets in the graph $G^k = G\cdot G\cdot \cdots \cdot G$, where there are k terms in the product. We obtain larger and larger unambiguous code alphabets this way, but at a cost in efficiency: We use longer strings. This observation led Shannon [22] to compensate by considering the number $\sqrt[k]{\alpha(G^k)}$ as a measure of the capacity of the channel to build an unambiguous code alphabet of strings of length k, and to consider the number $c(G) = \sup_k \sqrt[k]{\alpha(G^k)}$. The number $c(G)$ is called the *capacity of the graph* or the *zero-error capacity of the channel.* Computation of the capacity of a graph is a difficult problem. Indeed, even the capacity of the graph $G = Z_5$ which we discussed above was not known precisely until Lovász [15] showed that it equals $\sqrt{5}$. Meanwhile, as of this writing, $c(Z_7)$ remains unknown. For some bounds on $c(G)$, see Lovász [15], Haemers [12], Schrijver [21], and Rosenfeld [18].

3. Design of packet switched networks

Stephanie Boyles and Geoff Exoo (personal communication) have found an application of Ramsey theory in the design of a packet switched network, the Bell System signaling network. We describe the application in this subsection.[1]

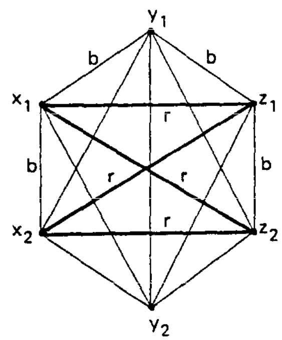


Fig. 3. Links are colored red (r), blue (b), or white (unlabelled).

Consider a graph in which vertices represent communications equipment joined by communications links or edges. The graph is assumed to be complete, that is, every pair of vertices is joined by a link. In some applications, vertices are paired up, and we would like to guarantee that in case of outages of some links, there will

[1] The author thanks Drs. Boyle and Exoo for bringing this application to his attention and for permission to present it here.

always remain at least one link joining every paired set of vertices. For instance, consider the graph shown in Fig. 3. The vertices labeled x_1 and x_2 are paired, the vertices labeled y_1 and y_2 are paired, and the vertices labeled z_1 and z_2 are paired. Outages occur at intermediate facilities such as microwave towers, trunk groups, etc. An outage at such a facility will affect all links sharing this facility. Let us color the intermediate facilities and hence the corresponding links. Fig. 3 shows such a coloring. Note that in case the red intermediate facility goes out, there will be no operative links between the pair of vertices x_1 and x_2 and the pair of vertices z_1 and z_2. This corresponds to the fact that the four edges $\{x_i, z_j\}$ form a monochromatic (red) Z_4. In general, designing a network involves a decision as to the number of intermediate facilities and which links will use which intermediate facilities. We would like to design the network so that if any intermediate facility is destroyed, there will remain at least one link for each paired set of vertices. If the vertex pairing may change after the network is constructed, we want to avoid all monochromatic Z_4's.

It turns out that $R(Z_4, Z_4) = 6$. (See Faudree and Schelp [8] or Rosta [19,20].) Thus, if there are just two intermediate facilities, there is a network with 5 vertices which has an assignment of links to intermediate facilities so that there is no monochromatic Z_4. Chung and Graham [6] show that $R(Z_4, Z_4, Z_4) \geq 8$. Thus, there is a network with three different intermediate facilities and 7 vertices and no monochromatic Z_4.

As we have said, designing a network involves a decision as to the number of intermediate facilities and which links will use which intermediate facilities. Intermediate facilities are expensive, and it is desirable to minimize the number of them. Thus, one is led to ask the following. If we have a network of n vertices, what is the least number of colors or intermediate facilities so that there is some network of n vertices and some coloring of edges (assignment of links to intermediate facilities) with no monochromatic Z_4. In other words, what is the least r so that if there are r Z_4's, $R(Z_4, Z_4, \ldots, Z_4) > n$? If $n = 6$, as in our example of Fig. 3, then since $R(Z_4, Z_4) = 6$, and $R(Z_4, Z_4, Z_4) \geq 8$, we have $r = 3$. We need three intermediate facilities. Boyles and Exoo point out that for their purposes, it is enough to estimate the number r using a result of Erdös (see Graham, et al. [11]) that a graph of n vertices always contains Z_4 if it has at least $\frac{1}{2}n^{3/2} + \frac{1}{4}n$ edges. If the $\binom{n}{2}$ edges of an n-vertex graph are divided into r color classes, the average class will have $\binom{n}{2}/r$ edges, and so, by the pigeonhole principle, some class will have at least $\binom{n}{2}/r$ edges. We want to be sure that no class has $\frac{1}{2}n^{3/2} + \frac{1}{4}n$ edges, so we must pick r so that

$$\binom{n}{2}\Big/ r < \tfrac{1}{2}n^{3/2} + \tfrac{1}{4}n.$$

4. Information retrieval

Yao [25,26] uses Ramsey theory in the study of information retrieval.[2] Suppose a table or a file has n different entries, chosen from a *key space* $M = \{1, 2, \ldots, m\}$, whose elements are called *keys*. We wish to find a way to store all subsets S of n elements from M in a table so that it is easy to answer queries of the form: Is x in S? A rule for telling us how to store the n-element subsets S of M is called a *table structure* or an (m, n) *table structure*. The simplest table structure is called a *sorted table structure:* We just list all elements of S in increasing order. For instance, if $m = 3$ and $n = 2$, a sorted table structure is shown in Fig. 4. The second table structure in Fig. 4 is called *cyclic*. Note that if we have the sorted table structure of Fig. 4, if we want to know if x is in S, we need to ask two questions. However, in the cyclic structure, we need to ask only one question, since by the cyclic nature of the table structure, the first entry in the row corresponding to S determines the second entry. A variant of the sorted table structure is the *permuted sorted table structure*. Here, we fix a permutation σ of $\{1, 2, \ldots, n\}$, and list elements of S in order according to this permutation. For instance, the third table structure of Fig. 4 is a permuted sorted table structure corresponding to the permutation which interchanges the first and second elements. Again, to determine if x is in S, two questions are needed with this table structure.

Fig. 4. Three table structures for storing 2 keys from a three-element key space $M = \{1, 2, 3\}$.

Sorted table structure		Cyclic table structure		Permuted sorted table structure	
set S	corresponding table	set S	corresponding table	set S	corresponding table
$\{1,2\}$	1 | 2	$\{1,2\}$	1 | 2	$\{1,2\}$	2 | 1
$\{2,3\}$	2 | 3	$\{2,3\}$	2 | 3	$\{2,3\}$	3 | 2
$\{1,3\}$	1 | 3	$\{1,3\}$	3 | 1	$\{1,3\}$	3 | 1

The computational complexity of information retrieval depends on the table structure and the search strategy, that is, the kinds of questions asked. It is measured by the number of queries needed to determine if $x \in S$ in the worst case. For instance, for a sorted table structure, the number of queries required is $\lceil \log_2(n+1) \rceil$ if a binary search tree is used. Let the complexity $f(n, m)$ be defined to be the minimum complexity over all conceivable (m, n) table structures and search strategies.

Theorem 2 (Yao). *For every n, there exists a number $N(n)$ so that*

$$f(n, m) = \lceil \log_2(n+1) \rceil \quad \textit{for all } m \geq N(n).$$

[2] See Chandra, et al. [5] for a different use of Ramsey theory in information exchange.

It follows from Theorem 2 that for m sufficiently large, using a sorted table structure is the most efficient method as far as information retrieval is concerned. There are two crucial ideas in proving this result:

Lemma 1. *If $m \geq 2n-1$ and $n \geq 2$, then for a permuted sorted table structure, $\lceil \log_2(n+1) \rceil$ probes are needed to determine if $x \in S$ in the worst case by any search strategy.*

Lemma 2. *Given n, there is a number $N(n)$ with the following property. If $m \geq N(n)$ and we are given an (m, n) table structure, then there is a set K of $2n-1$ keys so that the tables corresponding to the n-element subsets of K form a permuted sorted table structure.*

Theorem 2 follows from these lemmas. For given an (m, n) table structure and search strategy and a number $m \geq N(n)$, find the set K of Lemma 2. Then by Lemma 1, $\lceil \log_2(n+1) \rceil$ probes are needed in the worst case, just restricting the problem to subsets of K. Thus, the complexity is at least $\lceil \log_2(n+1) \rceil$, so $f(n, m) \geq \lceil \log_2(n+1) \rceil$. But we know that binary search on a sorted table structure has complexity $\lceil \log_2(n+1) \rceil$. Thus, $f(n, m) = \lceil \log_2(n+1) \rceil$.

We shall omit the proof of Lemma 1, referring the reader to Yao [26]. We shall present the proof of Lemma 2.

Proof of Lemma 2. Let us note that a set $S = \{j_1, j_2, \ldots, j_n\}$ of n keys is stored in the table structure in some order. If $j_1 < j_2 < \cdots < j_n$ and j_i is stored in the u_ith box of the table, then the set S corresponds to the permutation $u_1, u_2, \ldots, u_n$ of the integers $1, 2, \ldots, n$. For instance, in the cyclic table structure of Fig. 4, if $S = \{1, 3\}$, then $j_1 = 1$, $j_2 = 3$, $u_1 = 2$, and $u_2 = 1$. In a permuted sorted table structure, each set of n keys corresponds to the same permutation $u_1, u_2, \ldots, u_n$. Given an (m, n) table structure, let $\sigma(u_1, u_2, \ldots, u_n)$ consist of all sets S of n keys whose corresponding permutation is $u_1, u_2, \ldots, u_n$. For instance, in the cyclic table structure of Fig. 4, $\sigma(1, 2)$ consists of the sets $\{1, 2\}$ and $\{2, 3\}$ and $\sigma(2, 1)$ consists of the set $\{1, 3\}$.

Let $p_i = 2n-1$, all i, let $t = n!$, and let $r = n$. Let $N(n)$ be the Ramsey number $R(p_1, p_2, \ldots, p_t; r)$. Suppose that $m \geq N(n)$ and that we divide the r-element subsets (the n-element subsets) of the key space M into $t = n!$ parts, with each consisting of the set $\sigma(u_1, u_2, \ldots, u_n)$ of all n-element subsets S of M which are stored in the permutation $u_1, u_2, \ldots, u_n$. By the definition of $R(p_1, p_2, \ldots, p_t; r)$, there is for some i, a p_i-element subset ($2n-1$ element subset) K of M all of whose n-element subsets belong to a given $\sigma(u_1, u_2, \ldots, u_n)$. This proves Lemma 2. □

To illustrate this proof, consider the table structure of Fig. 5. Here, $m = 6$ and $n = 2$. The set $\sigma(1, 2)$ is given by the elements labeled * and the set $\sigma(2, 1)$ is given by the remaining elements. Note that there is a 3-element subset $K = \{1, 2, 5\}$ all of

Fig. 5. A (6,2) table structure with elements of the set $\sigma(1,2)$ represented by *.

Set S	Corresponding table	Set S	Corresponding table
*{1,2}	1 2	*{2,5}	2 5
{1,3}	3 1	{2,6}	6 2
{1,4}	4 1	*{3,4}	3 4
*{1,5}	1 5	*{3,5}	3 5
*{1,6}	1 6	*{3,6}	3 6
*{2,3}	2 3	{4,5}	5 4
*{2,4}	2 4	{4,6}	6 4
		{5,6}	6 5

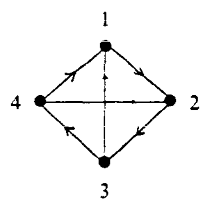
whose 2-element subsets belong to $\sigma(1,2)$.

We next note that if $n=2$, there is an alternative proof of Lemma 2 which gives a better value of $N(n)$. If $n=2$, then any table structure can be represented as a digraph whose vertex set is $M=\{1,2,\ldots,m\}$, and which has an arc from i to j if the set $\{i,j\}$ is stored as $\boxed{i \mid j}$. For example, the table structure of Fig. 6 yields the digraph shown in that figure. This digraph is a tournament. For $m \geq 4$, such a tournament always has a transitive triple, a triple of vertices $\{i,j,k\}$, with arcs (i,j), (j,k), (i,k). This means that all 2-element subsets of the 3-element set $\{i,j,k\}$ will be stored in a manner corresponding to the same permutation i,j,k. Thus, if we relabel the elements of M so that i becomes 1, j becomes 2, and k becomes 3, the 2-element subsets of $\{1,2,3\}$ will appear in the first sorted table structure of Fig. 4. Hence, if $n=2$, $N(n)=4$ will suffice to give us the conclusion of Lemma 2. In the example of Fig. 6, one transitive triple is $\{4,1,2\}$ and all 2-element subsets of this triple are stored in the same permutation 4, 1, 2. If we relabel the elements of M so that 4 becomes 1, 1 becomes 2, and 2 becomes 3, we have a sorted table structure in which the sets $\{1,2\}$, $\{2,3\}$, and $\{1,3\}$ are stored as in the sorted table of Fig. 4. Note that the conclusion of the lemma does not hold here; that is, for this table structure, there is no 3-element subset K and no permutation u_1, u_2 of $\{1,2\}$ so that

Fig. 6. A (4,2) table structure and the associated digraph.

Set S	Corresponding table
{1,2}	1 2
{1,3}	3 1
{1,4}	4 1
{2,3}	2 3
{2,4}	4 2
{3,4}	3 4

Associated digraph



all 2-element subsets of K are stored in the order u_1, u_2. If m were at least $R(3,3;2)=6$, we would be able to draw this conclusion.

5. The dimension of partial orders: A decisionmaking application

A digraph $D=(V,A)$ is *asymmetric* if $(u,v)\in A$ implies that $(v,u)\notin A$. An asymmetric digraph is *transitive* if $(u,v)\in A$, $(v,w)\in A$ imply that $(u,w)\in A$. (An arbitrary digraph is *transitive* if $(u,v)\in A$, $(v,w)\in A$ and $u\neq w$ imply $(u,w)\in A$.) A digraph which is both asymmetric and transitive is called a *(strict) partial order.* Patial orders arise in many contexts in decisionmaking. For instance, if V is a set of alternatives being considered, and $(u,v)\in A$ means that u is preferred to v, we get a partial order if preference satisfies the following conditions: If you prefer u to v, you do not prefer v to u; if you prefer u to v and prefer v to w, then you prefer u to w. Partial orders arise similarly if $(u,v)\in A$ means u is judged more important than v, u is judged more qualified than v, and so on. We shall use preference as a concrete example.

Suppose that $D=(V,A)$ is a digraph representing preference. If we are judging our alternatives a on the basis of one characteristic, say monetary value $f(a)$, we would have

$$(u;v)\in A \Leftrightarrow f(u)>f(v),$$

i.e., we would prefer u to v if and only if the value of u is greater than the value of v. If we judge on the basis of several characteristics, say monetary value $f_1(a)$, quality $f_2(a)$, beauty $f_3(a),\ldots,f_t(a)$, we might only express preference for u over v if we are sure that u is better than v on every characteristic. Thus, we would have

$$\begin{aligned}(u,v)\in A \Leftrightarrow\ & [f_1(u)>f_1(v)]\,\&\,[f_2(u)>f_2(v)]\\ & \&\,[f_3(u)>f_3(v)]\,\&\cdots\&\,[f_t(u)>f_t(v)]. \qquad (*)\end{aligned}$$

If A is defined using (*), then it is easy to show that (V,A) is a partial order.

The converse problem is of importance in preference theory. Suppose that we are given a partial order $D=(V,A)$. Can we find functions $f_1, f_2, \ldots, f_t$, each f_i assigning a real number to each a in V, so that (*) holds? It is not hard to prove that for every partial order (with V finite), we can find such functions for sufficiently large t. (The proof uses Szpilrajn's [23] extension theorem. See Baker et al. [2].) The smallest t such that there are t such functions is called the *dimension* of the partial order.[3] This notion is originally due to Dushnik and Miller [7], and has been widely studied. See Baker et al. [2], Kelly and Trotter [14] and Trotter and Moore [24] for surveys, and Roberts [16] for some applications.

[3] Strictly speaking, the dimension of the partial order is usually defined to be the smallest t such that the partial order is the intersection of t linear orders. However, our definition of dimension agrees with the more common one except for dimensions 1 and 2: The so-called (strict) weak orders can have dimension 1 by our definition, but not by the more common definition (see Baker, et al. [1]).

The dimension of many important partial orders has been computed. Here we shall study the dimension of one very important class of partial orders, the interval orders. To get an interval order, imagine that for each alternative a that you are considering, you do not know its exact value, but you estimate a range of possible values, given by a closed interval $J(a)=[\alpha(a), \beta(a)]$. Then you prefer a to b if and only if you are sure that the value of a is greater than the value of b, that is, if and only if $\alpha(a)>\beta(b)$. It is easy to show that the corresponding digraph gives a partial order, i.e., it is asymmetric and transitive. (In this digraph, the vertices are a family of closed real intervals, and there is an arc from an interval $[a,b]$ to an interval $[c,d]$ if and only if $a>d$.) Any partial order that arises this way is called an *interval order*. The notion of interval order is due to Fishburn [9].

In studying interval orders, which are somehow one-dimensional in nature, it came as somewhat of a surprise that their dimension as partial orders could be arbitrarily large. That is the content of the main theorem of this section. It implies that if preferences arise in the very natural way that defines interval orders, we might need very many dimensions or characteristics to explain preference in the sense of equation (*).

Theorem 3 (Bogart, Rabinovitch, Trotter [4]). *There are interval orders of arbitrarily high dimension.*

Proof. Suppose that $I(0,n)$ is the interval order defined by taking all closed intervals $[a,b]$ with a,b integers between 0 and n inclusive, and by taking an arc from $[a,b]$ to $[c,d]$ if and only if $a>d$. We shall show that given $t\geq 2$, there is a number $N(t)$ so that if $n\geq N(t)$, $I(0,n)$ has dimension greater than t. In particular, let $N(t)=R(p_1,p_2,\ldots,p_t;r)-1$ with $p_1=p_2=\cdots=p_t=4$ and $r=3$. Now suppose that $n\geq N(t)$ and that $I(0,n)$ has dimension less than or equal to t. Then there are functions $f_1,f_2,\ldots,f_t$ so that (*) holds. Now consider the set of all integers between 0 and n and consider the 3-element subsets $\{u,v,w\}$. Suppose that $u<v<w$. Then neither $([u,v],[v,w])\in A$ nor $([v,w],[u,v])\in A$. It follows by (*) that there are i and j so that

$$f_j([v,w])\geq f_j([u,v]) \quad \text{and} \quad f_i([u,v])\geq f_i([v,w]).$$

Place the triple $\{u,v,w\}$ in the ith class, $i=1,2,\ldots,t$, if i is the smallest integer so that $f_i([u,v])\geq f_i([v,w])$. Since $n\geq N(t)$, we have $n+1\geq R(4,4,\ldots,4;3)$. Thus, we know that for some i, there is a 4-element subset $\{x,y,z,t\}$ of $\{0,1,\ldots,n\}$ all of whose 3-element subsets are in the ith class. Thus, if $x<y<z<t$, we have

$$f_i([x,y])\geq f_i([y,z]) \quad \text{and} \quad f_i([y,z])\geq f_i([z,t]).$$

Hence, $f_i([x,y])\geq f_i([z,t])$. But $([z,t],[x,y])\in A$, so we should have

$$f_i([z,t])>f_i([x,y]).$$

Hence, we have reached a contradiction, which implies that $I(0,n)$ has dimension larger than t. □

References

[1] K.A. Baker, P.C. Fishburn and F.S. Roberts, Partial orders of dimension 2, interval orders, and interval graphs, Paper P-4376, The RAND Corporation, Santa Monica, CA (1970).

[2] K.A. Baker, P.C. Fishburn and F.S. Roberts, Partial orders of dimension 2, Networks 2 (1972) 11-28.

[3] C. Berge, Graphs and Hypergraphs (American Elsevier, New York, 1973).

[4] K.P. Bogart, I. Rabinovitch and W.T. Trotter, A bound on the dimension of interval orders, J. Combin. Theory 21 (1976) 319-328.

[5] A.K. Chandra, M.L. Furst and R.J. Lipton, Multi-party protocols, Proc. 15th Annual ACM Symp. Theory of Computing (Assoc. Comput. Mach., New York, 1983) 94-99.

[6] F.R.K. Chung and R.L. Graham, On multicolor Ramsey numbers for complete bipartite graphs, J. Combin. Theory (B) 18 (1975) 164-169.

[7] B. Dushnik and E.W. Miller, Partially ordered sets, Amer. J. Math. 63 (1941) 600-610.

[8] R.J. Faudree and R.H. Schelp, All Ramsey numbers for cycles in graphs, Discrete Math. 8 (1974) 313-329.

[9] P.C. Fishburn, Intransitive indifference with unequal indifference intervals, J. Math. Psychol. 7 (1970) 144-149.

[10] R.L. Graham, Rudiments of Ramsey Theory, CBMS Regional Conf. Series in Math. 45 (Amer. Math. Soc., Providence, RI, 1981).

[11] R.L. Graham, B.L. Rothschild and J.H. Spencer, Ramsey Theory (Wiley, New York, 1980).

[12] W. Haemers, On some problems of Lovász concerning the Shannon capacity of a graph, IEEE Trans. Inf. Theory 25 (1979) 231-232.

[13] Z. Hedrlín, An application of the Ramsey theorem to the topological product, Bull. Acad. Pol. Sci. 14 (1966) 25-26.

[14] D. Kelly and W.T. Trotter, Dimension theory for ordered sets, in: I. Rival, ed., Ordered Sets (Reidel, Boston, 1982) 171-211.

[15] L. Lovász, On the Shannon capacity of a graph, IEEE Trans. Inf. Theory 25 (1979) 1-7.

[16] F.S. Roberts, What if utility functions do not exist?, Theory and Decision, 3 (1972) 126-139.

[17] F.S. Roberts, Applied Combinatorics (Prentice-Hall, Englewood Cliffs, NJ, 1984).

[18] M. Rosenfeld, On a problem of C.E. Shannon in graph theory, Proc. Amer. Math. Soc. 18 (1967) 315-319.

[19] V. Rosta, On a Ramsey type problem of J.A. Bondy and P. Erdös, I, J. Combin. Theory (B) 15 (1973) 94-104.

[20] V. Rosta, On a Ramsey type problem of J.A. Bondy and P. Erdös, II, J. Combin. Theory (B) 15 (1973) 105-120.

[21] A. Schrijver, A comparison of the Delsarte and Lovász bounds, IEEE Trans. Inf. Theory 25 (1979) 425-429.

[22] C.E. Shannon, The zero-error capacity of a noisy channel, IRE Trans. Inf. Theory 2 (1956) 8-19.

[23] E. Szpilrajn, Sur l'extension de l'ordre partiel, Fund. Math. 16 (1930) 386-389.

[24] W.T. Trotter and J.I. Moore, Characterization problems for graphs, partially ordered sets, lattices, and families of sets, Discrete Math. 15 (1976) 361-368.

[25] A.C. Yao, Should tables be sorted? (preliminary version), Proc. 1978 IEEE Symp. Foundations of Computer Science, Ann Arbor, MI (October 1978).

[26] A.C. Yao, Should tables be sorted? J. ACM 28 (1981) 615-628.