63-3-6

IBM RESEARCH

Scale-1

Applied Research Program
Aerospace Intelligence Data System [AIDS]

VOLUME I

Progress summary

9.60

(7) $4.00
(5) 872300

(6) APPLIED RESEARCH PROGRAM
AEROSPACE INTELLIGENCE DATA SYSTEM (AIDS)

(9) QUARTERLY REPORT NO. 4 for
PERIOD ENDING May 24, 1962.

(15) Contract AF 19(628)-10

Volume I.

Submitted to

Electronic Systems Division
Air Force Systems Command
Laurence G. Hanscom Field
Bedford, Massachusetts

International Business Machines Corporation
Thomas J. Watson Research Center
Yorktown Heights, New York

(7) NA
(8) NA
(10) NA
(11) 24 May 62
(12) 109p
(13) NA
(14) NA
(16) NA
(17) NA
(18) NA
(19) NA
(20) U
(21) NA

J.C.

ABSTRACT

Progress on the Aerospace Intelligence Data System (AIDS) Research Contract (AF 19(626)-10) is presented. The one year effort is summarized and the progress and status of the fourth quarterly period of this effort is reported. The design philosophy of an adjunctive equipment group (AN/GYA-( )) to be employed in developing an on demand computer service capability with a large data processor is described. A technical paper discusses evaluating search time bounds for various probability distributions of the file location of stored information and several search strategies. The theoretical basis is established, in this report, for a procedure of sampling files in order to estimate (with confidence levels obtained) the number of filed items having some specified characteristics in common. A methodology (computer oriented, using citation data) for establishing "clusters" of related items is reported.

APPLIED RESEARCH PROGRAM

AEROSPACE INTELLIGENCE DATA SYSTEM (AIDS)

QUARTERLY REPORT NO. 4

SECTION I

Introduction and Summary

Applied research work performed under Contract AF 19(626)-10 is guided by Work Statement dated 20 March 1961 as revised on 8 August 1961. Four technical areas of effort are identified therein, for which applied research effort of either a reporting, or of an execution nature is prescribed. These technical areas and corresponding categories of effort are identified and reported on in this quarterly report. The objective of this contractual effort, as stated in the work statement, is: "The AIDS Applied Research Program ... has the overall goal of supporting the progress towards a SAC operational sub-system by reporting on results of non-numerical data automation techniques and equipment development now in progress in industry during the time period of AIDS. Selected implementation of these results will assure an optimum data handling equipment complex and utilization of the most efficient techniques in the SAC operational intelligence data handling subsystem."

Inasmuch as this report represents the fourth report of a continuing research effort, a summary of the past effort and accomplishments on each task, since the project's inception, is provided in addition to reports on the work of the last quarter.

The general organization of this report is as follows:

Volume 1

(1) One year review and last quarter progress and status summary

(2) Appendix A: The General Design Philosophy of the AN/GYA-( )

(3) Appendix B: Optimum Search Strategies

(4) Appendix C: Sampling for Co-Occurrence

(5) Appendix D: An Experimental Investigation of "Clustering"

Volume 2

(1) Test of the Datacom Model 408-2 (to be submitted upon completion of tests)

SECTION II

Summary Report

Task I - Automatic Print Reading

General Comment:

This task has consisted of a monitoring effort on progress being made on the task of automatic methods of converting printed text into machine processable form.

Results:

A complete bibliography of "Character Recognition System Patents" was provided in the second quarterly report (Volume II). This bibliography additionally included many patents that pertained to elements of character recognition systems as well as patents pertaining to allied fields such as facsimile, curve following, signal analysis, code and perforated record analysis.

Two research efforts on new and promising approaches to character recognition were reported in some technical detail. Verbal presentations and demonstration of a semi-automatic method of rapid and economical conversion of textual material to machine processable form by virtue of a developed "Stenowriter" system were provided.

Task II - Storage and Addressing

General Comment:

This work included conceptual and design efforts for exploiting

the high storage densities and unique addressing features of the Air Force's AN/GSQ-16 equipment in conjunction with a large scale data processor, for the purpose of providing: "on demand" computer service to intelligence analysts, greatly increased assistance to analysts in formulating queries, and development of methods of semi-automatic or automatic pre-processing of messages or text into machine processable form. Sub-tasks include:

(1) Photostore Disc storage of "quick demand" programs

(2) Searching of formatted files

(3) Processing of unformatted file materials

Task II - Sub-task 1: Disc Storage of Quick Demand Programs

Results:

Three alternative engineering approaches for connecting a Photostore (AN/GSQ-16) to an IBM 7090, involving a minimum of computer interference or "down-time", have been examined with the result that recommendation that one of these designs -- a direct data channel connection -- be accomplished.

The general design philosophy of a recommended effort (consummated in contract AF 30(602)-2754) for the AN/GYA-( ) Computer Storage Integration Group has been evolved,, and is outlined in appendix A of this report.

Task II - Sub-task 2: Searching of Formatted Files

General Comment:

The problem of searching files of information to answer specific requests has been examined from a general theoretical approach with results being applicable to many equipment and file-content configurations.

Results:

(a) A fundamental mathematical relation expressing minimum average file search time in terms of the probability of location of the "sought-for" file material has been found.

(b) An important new technique for solving the foregoing mathematical expression for minimum average file search time has been developed.

(c) As a result of (a) and (b), optimal search strategy for a random access file mechanism having material filed within it, in an ordered manner, has been determined. Similarly, for this case the corresponding minimum average search time has been determined.

(d) A mathematical method of sampling files for the

possible co-occurrence of specified terms or events has been developed which should have considerable utility in "on-demand" multiprocessing information retrieval systems which employ rapid access, large capacity peripheral files.

Progress and Status this Quarter:

Dr. Eugene Wong has continued his development of a mathematical basis for determining optimum search strategies. He has examined in detail the problem of determining efficient techniques for locating items stored in tabular (ordered) format in computer storage units. These are reported in Appendix B along with suggested avenues of research leading to practical applications of these "bounds".

Dr. C. Abraham has further developed the mathematical basis for sampling a file for items having a co-occurrence of say a descriptive term or associated event. This work is of particular significance for "on-demand" multi-processing computer complexes where some "feedback" to the analysts making the queries must be provided concerning the estimated magnitude of file response to be obtained. Such "feedback" based on "sampling" rather than on exhaustive file search, can obviate unnecessarily time consuming searches and

allow the analyst to narrow the search if he so desires by adding qualifying search terms with logical conditions pertaining to their relationships. This recent work is described in Appendix C along with associated work on the correlation task.

Task II - Sub-task 3: Processing of Unformatted File Material

General Comment:

Examination of this problem has led directly to the central problem of information retrieval; that is, "what is the best form into which one should translate (or characterize) unformatted source material, for machine searching?" and secondarily, but important in an economic sense, "How can unformatted text be converted (or processed) to this form?".

Work supported by AF 49(638)-1062 has provided asymptotic expressions for machine search time for a large file of retrievable items, searchable by a query made up of terms in conjunctive and disjunctive form, and an asymptotic expression for machine search time. This search time, for a large file of retrievable items, is proportional to the square of the average number of retrievable items in the file that pertain to each description term (non-negated) that occurs in the query (reference AF OSR-TN-61-2).

Dr. Bohnert of the Experimental Systems Department, IBM Research, has been studying with some success transformations between pseudo-English language sentence structure and equivalent symbolic logic expressions. Additionally, J. Andress of this Department has completed a powerful program-tool for generating sentences from stated rules of grammar (to be presented at the International Congress of Linguists, MIT, 28 August 1962, by Lees, Matthews and Andress). This program should allow penetrating analysis by linguists of the adequacy of hypothesized grammatical rules, through examination of large quantities of sentences generated by these rules.

Progress and Status this Quarter:

As a result of the above cited current and pending investigations (including a statistical study of texts by the IBM San Jose group) which are attacking fundamental aspects of English to English translation, a minimum of effort was expended on this task of the project effort.

Task III - Multiple Interrogation

General Comment:

The basis of this effort is the premise that large computer complexes which are intended in part to aid intelligence analysts in

accomplishing their functions, should be tailored to provide a convenient, timely, and useable service rather than serving entirely as large scale "batched problem" processors whose "throughput" rather than general utility is considered paramount. Two major divisions of non-hardware effort are identifiable:

(1) Programming research for multiprocessing

(2) Query formulation programming

Each of these "software" efforts are intended eventually to be integrated with, and employ the Computer Storage Integration Groups AN/GYA-( ) as outlined in Appendix A.

Results:

An overall technical plan has been prepared for a combined hardware and software program leading to a demonstrable capability of mixed simultaneous operation of a routine computer task and a user query formulation and file search (see Quarterly Report No. 2 Volume III and Appendix A of this report).

The outline of the programming approach and structure that is being pursued is as follows:

(1) development of an advanced supervisory control program with complete accounting lists of the status of each program being worked on (or requested) and

(2) quick reference to program segments stored on the photostore as an adjunct to use of the main core memory for operating program storage. In addition, the first form of a query formulation procedure and program, has been nearly completed and de-bugged. This program stresses the interplay between the computer and the user (at a suitable display console). "One-shot" query preparation and extensive waiting for an answer will be avoided by computer provision of partial responses and possible alternative question sequences which are posed to the user as he develops his query in a knowledgeable and natural manner.

Progress and Status this Quarter:

The query simplification program in a simple outline form has been completed and is currently being de-bugged. The program in its initial form presents a sequence of display instructions to the person making the query (as follows):

Display Title	Comment
(1) Enter Field Name	User is asked to select an appropriate "Field" or sub-topic from the total computer stored data

	Display Title	Comment
(2)	Specify Parameter Match Criterion	The user is given the opportunity to specify query match conditions such as "equal to", "greater than", "less than", or "less than and equal to", "greater than and equal to" and "not equal to"
(3)	Enter search parameters	An opportunity is here provided to insert terms, names, and characteristics that pertain to the query. This is equivalent to the "inclusive or" retrieval of all Field items containing these terms or characteristics.
(4)	Expand, Narrow, or End Query	This provides an additional opportunity to expand the file search by adding additional subject fields and requesting search on an "inclusive or" or a "conjunctive (logical intersection)" basis.

This work will continue to be expanded upon this program outline with the demonstration of its utility as the primary goal.

The Advanced Supervisory Control Program (overall executive control program) to control the multiprogrammed operations on the AN/GYA-( )-IBM 7090 Integrated equipment has been outlined in scope and some detailed programming on key portions of it has begun.

Task IV - Correlation and Classification Techniques

General Comment:

The general intelligence problems characterized by the following questions are being examined: "What attributes or characteristics are really important with respect to the item or problem in which one is interested?" or "Do these particular measured characteristics signify this (or that) overall class of actions or structures?", or "If this person or thing pertains to something of possible interest are these other people or things also involved?"

Results:

Classification: Early in this program, basic work was accomplished that led to an FSD developed FORTRAN program for determining "into which of two categories or classes an item (having up to 40 measured characteristics) should be placed. This work has been extended to obtain methods of determining a confidence measure for such classification action even though it is based on an extremely

small sample of the population classes into which one is attempting to place the new item.

Correlation: Three experiments were completed in the process of developing techniques of "clustering" or determining fundamental relationships among items (people, documents, R & D activity, suspected military installations, etc.) on the basis of measurable characteristics of these items which do not obviously nor directly yield a possible or probable "connection" between them. In addition, as reported in Quarterly Report No. 3 (Volume I), Dr. Abraham reviewed and mathematically analyzed the appropriateness of use of various "association measures".

A report of an experiment that delineates and assesses a new concept of dissemination of information based upon categorizing groups of report recipients on the basis of a high probability of common reading interest, was reported in Quarterly Report No. 1.

Progress and Status this Quarter:

Miss Reisner has continued the work on methodologies of "clustering", that is, grouping of items together on some basis of of common relationship. Since the co-authorship relationship by itself appeared to provide a highly fragmented clustering effect, experimental effort was changed to the use of citations (literature)

so that results could be correlated more obviously with known facts. Of particular significance is the devising of a "path-tracing" algorithm to determine "connectivity" of items of interest. Appendix D of this report describes the work of this quarter on "clustering".

Other:

In November 1961 a Facilities Contract was executed with the Electrada Corporation for an off-line editing and message composing keyboard and display console. This equipment was delivered to the Thomas J. Watson Research Center on 9 July 1962 where it is presently undergoing test. Volume II of this fourth quarter report will be submitted to cover results of these tests.

Appendix A

Overall Design Philosophy of the AN/GYA-( ) Computer Storage Integration Group

Introduction:

Large scale computer installations typified by the IBM 7090 have been profitably exploited by many organizations in the processing of scientific, engineering, business, and other information. Until recently only a small fraction of the total personnel in a typical organization have been able to directly utilize these machines, albeit in a highly efficient manner. The advent of symbolic programming, source problem programming languages, and compilers, have extended the power of these processing complexes to the service of many more people. However, in the realm of information processing where human mental activity needs continuous assistance, rather than intermittent assistance or replacement, computers are not used to the extent that the potential of some of their characteristics would indicate should be possible. Primary deterrents to more complete exploitation of machine information processors would appear to be the essentially unilateral and discontinuous time-batching method by which men are forced to communicate and operate with computers, even when complete familiarity with the language and procedures of this communication are known. This is of course the result of unwillingness of most reasonable managements to allow "tying up" of the totality, or even large portions of these powerful processors, either on a time or equipment basis, while one man "thinks" about his next request or response to the machine. The need for discontinuous but "on demand" use of a computer is keenly felt by programmers when they attempt to "de-bug" new programs. This need is further appreciated by scientific and engineering

personnel who frequently would like a more continuous interplay with the machine during the processing of a problem even though this inter-reaction is interspersed with pauses in machine activity on that problem to allow human thought and possible redirection of machine activity (see for example reference 1). Similarly, in mixed machine and human information processing, typified by military intelligence analyst activity, there would appear to be a great need for this discontinuous but "on demand" service by a computer complex.

It is extremely doubtful, except for operational situations involving a few major specific functions, that complete dedication of a large processor to this "on demand" service is warranted. This would be particularly true if the valuable services now provided to an organization on an efficient basis are seriously curtailed. It is to the obtainment of a compromise equipment configuration, useful in an intelligent information processing environment where the "bread and butter" "batched processing" tasks of the large scale processor are affected only slightly and "on demand" service is also conveniently and efficiently provided, to which the development program on the AN/GYA-( ) is directed.

Postulation of Processing Environment:

The postulation of an intelligence information problem environment determines that existing processing needs include operation on numerical computational problems, logical and lexical operations on information in language form, and retrieval of information items from large scale storage, as well as problems having mixtures of all three processing categories. Indeed, if

processing problems could be uniquely categorized then specialized hardware dedicated to each type of problem would be a feasible and possibly economical approach to the goals of this endeavour. However, the following type of problem is presented as an example of a desirable type of service that must be provided: an analyst, (let us say a political specialist), enters into the computer complex ten names that have come to his attention by reference to some recent literature. He asks that all available news items (perhaps in abstracted form) relative to these people be provided. The processor replies, via a suitable display that approximately 500 newspaper articles are involved. The analyst decides, after some thought, that he would be satisfied initially to have all items which are source dated after 1 January 1960. He is informed of the number of items in the file response, or at least if this number is greater say than that which can be conveniently handled by the display screen. Let us say it isn't, and he requests "display", rather than hard copy print out. After examining the screen display he decides that two of these people appear to have similar knowledge on a subject of interest to him. He requests from a particular file the biographical data on these men and selects from this as well as from the previous news items a set of criteria that can be used to determine possible relationships between these people and others who may be noted in the file. He may select for example specific "location-time intervals" from the biographical data and then ask the computer to retrieve from a "location-time" organized event file, all items listed under the intervals that he selected, specifying perhaps some logical

relationship of co-occurrence of an action or event within each retrieved "location-time" interval. At this point some internal computer sampling procedure should be accomplished by the computer in order to place a bound on the computer "tie-up" on any specific request. After the computer informs the user of the estimated magnitude of items to be handled as a result of his request, the user may instruct the computer to proceed with a clustering matrix or "graph tracing" operation which then provides him with a grouping of names, each group having a decreasing degree of "connectivity" by virtue of their relationship reported in the event file. At this point the analyst requests a file time and CPU time accounting of his use of the complex and decides to ask for a print-out so that he can submit a "batched" request for complete biographical and news item print out of either the pertinent document references or the actual source material, so that he may assess the reasons or "common purposes" that connect the groups of people (if such a common purpose exists).

The above description, although, fictitious, is believed realizable and typical of service that is needed in many human information processing environments. The question of course now arises "So it can be done, what happens to the processing of the daily organizational reports; or worse, the payroll?" The answer to this question involves consideration of processor throughput in terms of the number of concurrent "on demand" users and the statistical characteristics of their queries in terms of processing load, file search load, input/output operation load, and so forth.

We obviously don't know and presently have no reasonable means of determining quantitative assessments of these factors until such a service is provided at least in development form. Accordingly we have arbitrarily set some initial criteria for our research and development program to provide "bounds" on what is done to assess this problem area.

(1) The characteristics of the central processor (IBM 7090), at least initially, will be unmodified per se.

(2) "De-bugged" programs only will be allowed within the processor during the "on-demand" service experimentation. (Later in the program, hardware modifications for memory allocation and protection will be examined in order to provide greater flexibility in this respect.

(3) More than one concurrent "on demand" user will be hypothesized for the development program in order to provide for generalizing the usefulness of experimental results.

(4) Assessment of expected impact on "bread and butter" program operation in terms of "on demand" users will be a goal of the program.

Processor "Throughput" Discussion:

Obviously any specified computer equipment configuration has limits as to the number of "on demand" programs that it can execute in a specified length of time. The portion of the computer complex that fixes such a limit will depend on the predominant characteristics of the processing that the problem-mix requires. The following possibilities exist as unique or joint occurrences in a

central processor type computer.

Predominant Characteristics of Processing Required	Possible Computer - System Limitations
1. Extensive computations	Basic Core Memory and CPU cycle time of a central computer.
2. High Input/Output (I/O) data flow rates.	I/O transfer cycle time availability.
3. Frequent references to peripheral storage devices.	External-storage average item access time and availability of peripheral storage capacity.
4. Extensive transfer within programs to sub-routines and other non-contiguous program segments.	High-speed memory capacity or storage average item - access time of external storage.
5. Extensive non-contiguous data-block branching.	High-speed memory capacity or average item access time of external storage.
6. Many, I/O device to I/O device, data transfer operations.	I/O data transfer rate or I/O transfer cycle time availability.

The above list pertains to a single central processor of a serial processing type (with overlap of input/output and CPU operations allowed). The list for distributed systems, that is, processors with multiple or distributed processors (or component processing units) would be different, with such limiting factors appearing as "intra-computer switching and communication rates" or

"supervisory program operation rates".

Although quantitative assessments can not be realistically made without experimental trial of the actual operation of an "on demand" computer service, qualitative assessments are necessary to produce a design approach for such experimentation. Review of the above list, keeping in mind an intelligence environment for operation, has allowed the following guide lines to be drawn:

(1) Extensive logical operations or computations: This problem characteristic is more typical of scientific problems and would not be expected as a predominant characteristic of the "mix" of intelligence processing problems (such problems will occur of course).

(2) High Input/Output (I/O) data flow rates: If the "on-demand" processing service were to be provided to a multiplicity of high data rate devices such as radar sets or multiplexed communication data channels, then I/O transfer cycle time could become saturated. Generally speaking, if "on demand" computer service is to be provided primarily to human beings the superimposed data transfer rates required for communications from these human beings would be negligibly small. It is true however that unless some measure of internal control is provided, a few "on demand" queries for information could unnecessarily cause a saturation of available I/O channels if they required total file contents to be scanned by the central processor unit for matching query criteria. This is a problem that must be circumvented in an "on demand" system design.

(3) Frequent references to peripheral storage devices: This problem-processing characteristic is typical of information retrieval application, input-data transformation to machine processable form, and processing that involves frequent references to pre-computed data tables or tables of constants. All of these processing characteristics would be typical of intelligence and other intellectual information processing environments.

(4) Extensive transfers within programs to sub-routines and other non-contiguous program segments: This problem does not appear seriously in the normal "batched processing" use of a computer (except in some compiler operations) since large contiguous segments or even total programs are usually brought into main high speed core memories for rapid reference. In this case, overlapping operations of several programs can be efficiently accomplished, and the occasional reference to a special macro-program or sub-routine via I/O operations need not produce great loss in computer throughput. If however, "on demand" service to many users each with different program requirements is considered the capacity of even a large high speed core memory may be exceeded. Teager and Morse (see reference 2) working toward similar "on demand" computer service goals in an engineering and scientific environment have noted that greater high speed memory capacity could be effectively provided for processing purposes, if the amount of high speed memory required to contain the programs could be reduced. Their preliminary assessments showed that over 20% of the machine time of a particular computing complex goes to programs which take

either greater than 16000 words of memory (core) or require more than 15 minutes running time. Their analysis further indicated that on the average only three or four programs can be run simultaneously in a 32K word core-memory machine, and if the alternative of processing each program on a priority sequence basis were followed, unacceptable delays would be encountered by individual users.

(5) Extensive non-contiguous data-block branching: This type of problem arises in many instances of processing data, each item of which requires a processing reference to a data item not contained in a contiguous block of data held in the high speed core memory. Again, processing of information that is heavily "cross linked" with other items in a total file, linquistic operations such as synonym and word-phrase linking, and retrieval of information based on "chain" storage where connected or relevant items are connected by "see also" or "see next" linkages, all lead to extensive data branching. Here again, as in the frequently branching or the highly segmented type of program problem, limitation will be imposed either by the ability of the high speed core memory to hold all pertinent data for all the programs, or by the time consumed in finding information in external storage.

(6) Many, I/O device-to-I/O device, data transfer operations: An example of this type of processing problem might be the transfer of large quantities of digitally stored information from say tape units, through the CPU for an editing process, to an output printer. This type of operation is a prevalent one in large processing complexes and frequently is assigned to relatively

independent smaller scale peripheral data processes. In any event many of these operations occur in problems susceptible to "batched" processing operations, and are frequently handled that way.

Computer Through-Put Considerations:

In reviewing the foregoing categories of probable processing characteristics of problems in an intelligence operation, it would appear that categories (3), (4) and (5) require special attention in the design of an "on demand" computer service. Item (3), frequent references to peripheral storage, in particular may be effected not only by the problem environment (through problem characteristics (4) and (5)) but also by methods of achieving "on demand" service from the computer complex. Since the amount of high speed core memory appeared to be an important possible limiting factor on the number of contiguous programs (and associated data blocks) that can be operated upon "concurrently" (on a dense time multiplexed basis); methods that could effectively augment the available high speed memory were explored. Direct supplementation of the high speed core memory of course is possible and if core memory augmentation factors of up to two were all that were required, this would be perfectly feasible by engineering addition to the computer of a parallel memory unit. However, high speed read/write memories are the most expensive form of computer memories for storage of sparsely used data, on the order of one dollar per bit in cost in an operational set-up. An alternative was to seek a peripheral storage device whereby the complete program repertoire of the computer installation would be contained, and needed portions thereof would be

available for quick transfer as needed into the main core memory. Now the amount of storage capacity needed for the routine program repertoire of say an IBM 7090 in a scientific computing center may be as low as one to two million bits; this would be the case where a great majority of the operating programs are maintained by the various users, external to the computer center, and where no "on demand" service is provided. On the other hand, cursory review of the total published library of programs for the IBM 704 system (704 Share Abstracts, May 15, 1960) representing programs developed for a great variety of customer applications in the scientific area indicates that a versatile program library may fall in the ten million bit category, or greater. It would seem logical to assume that an "on demand" computer service should provide for the use of a wide variety of program routines and that peripheral storage capacity on the order of 10^7 bits should be a developmental criterion if program repertoire storage only, exclusive of other functions, were to be considered.

The IBM 7090 can execute on the order of 2×10^5 program instructions per second (on a non-data-limited basis). Therefore any scheme of effectively augmenting high speed memory capacity so that multiple programs can be executed on a dense time shared basis; must consider the effect of retrieval and "read-in" of the program instructions. Unfortunately, extensive statistical data on the relationship between program segment size and the average number of program transfers outside each such program segment, is not available. From Research on instruction mixes for the

IBM 7094 program one would expect that the probability density (per word) of a sequential operating instruction being outside a program segment whose size equals two instructions, to be about 0.55 and this probability should eventually reach a value close to the reciprocal of the program size for program segments that approach average maximum program size. Review of the above referenced Share program abstracts for the IBM 704 indicates that special sub-routines or essentially self-contained programs (with exception of compilers) average about 200 instruction words each. Assuming that primarily the terminal instruction in these routines involves transfer to an instruction outside a 200 word average set or segment, then the probability density per word of encountering a non-included instruction would be on the order of 0.01 for a 200 word program segment size.

With this probability of requiring retrieval of another program segment from a peripheral storage device (assuming that each additional segment needed does not also reside in the high speed memory) then a program segment access time of about a millisecond (per segment) would be required for a computer that averages on the order of 10^5 instructions "consumed" per second. This estimate is predicated on the program-segment retrieval operation which does not appreciably lower the average useful program instruction execution rate and that the computer need not wait for one program-segment in order to continue execution of other program instructions. Several practical factors tend to reduce the rate at which a program's total instructions are

"consumed". First, programming makes extensive use of highly efficient reiterated "loops" or short sequences of instructions and this reduces the effective rate at which the computer moves through a programs' total of instructions. Second, a certain amount of within program "housekeeping" functions involving say transfer of data within memory, etc. may further reduce the effective rate of total program instruction consumption.

These factors may cause a program instruction "consumption rate" (not instruction execution rate) to drop to about one fifth or even as much as one tenth of the basic instruction execution rate of the computer (based on average 2 cycles per instruction). If, and the speculative nature of this cannot be overemphasized, the above assumptions are approximately correct, then the program instruction "consumption rate" would be about 20,000 per second, as averaged over the total program "mix" of a computer complex. A program segment length, selected to be about 256 instruction words in length would lead to the probability density per word that a transfer of instruction sequence outside a segment will occur, of about 0.01. This estimate is based upon sparse data, and attempts to obtain further information on this will be made.

The fraction of machine capacity that should be dedicated to "on demand" service must be determined. At this time it is impossible to predict the effect of segmentation of program data, but if one arbitrarily sets the limit that less than 10% of the program instruction consumption rate be dedicated to the "on demand" service then the following constraints should hold as an approximation:

(1) $\tau_s = \left[P(L) \cdot I \cdot K_s\right]^{-1}$

where: K_s = fraction of program instruction consumption rate allocated to program segment retrieval.

$P(L)$ = probability density per program instruction that an "outside" segment instruction will be required. (Segments, L words long)

I = average program instruction word "consumption rate" (per second)

τ_s = average time required to obtain (in a main core memory position) program segments of length L from peripheral storage.

(2) $\tau_s = \tau_1 + \tau_2 + \tau_3$: average total time to retrieve a program segment.

where: τ_1 = average time required to address and locate a desired program segment in a peripheral storage device.

τ_2 = average time to read out a selected program segment into a buffer storage.

τ_3 = average time to read selected program segment from buffer to main high speed core memory.

Combining (1) and (2)

(3) $$\tau_1 = \frac{1}{P(L) \cdot I \cdot K_s} - (\tau_2 + \tau_3)$$

assume the following:

K_s = 0.1

L = 256 words (36 bits each) of program instruction words.

$P(L) \approx .01$ transfers per program word "consumed".

$I \approx$ 20,000 instructions "consumed" per second.

$\tau_2 \approx$ 0.009 seconds (1×10^{-6} seconds/bit* x 256 words x 36 bits/word).

$\tau_3 \approx$ 0.008 seconds (256 words x 30×10^{-6} seconds/word**).

Then τ_1 (average time to locate a desired program segment in the peripheral storage unit) becomes from (3): $\tau_1 \approx$ 33 milliseconds

* 1.0 bit per microsecond is an assumed high speed peripheral storage data transfer rate.

**This assumes an average of 14 machine cycles per 36 bit word transferred via an I/O device.

Obviously, if τ_1 and τ_3, that is the time to address and locate a program segment and the time to read it out of buffer storage can be overlapped, then (3) becomes (assuming $\tau_3 < \tau_1$):

$$\tau_1 = \frac{1}{P(L) \cdot I \cdot K_s} - \tau_2$$

and in this case τ_1 becomes:

τ_1 = 41 milliseconds

Alternatively, in this latter case, if the addressing time (τ_1) is maintained at 33 milliseconds then the fraction of machine cycles used for "on demand" service can be upped from 10% to 14%.

If auxiliary program storage alone functionally dictated the desired characteristics of an auxiliary storage device for use in providing an "on demand" user service, the aforementioned storage characteristics could be satisfied by technology which is already in an advanced development or operational status (say by magnetic drum storage devices). One should expect however that the developmental efforts toward providing such service should anticipate that each user will eventually be allowed considerable flexibility with respect to the terms (at least) if not the language which he uses to communicate with the computer and its stored information, without imposing additional processing load on the existing computer complex. This implies large dictionary storage and language processing capabilities. For this reason the basic technique and equipment employed in the USAF AN/GSQ-16 language processing system were selected as a key functional element of the equipment complex needed to achieve the goals of this program. This equipment (referred to as the Photostore) currently has a storage capacity of 60 million bits of information. Item retrieval times on the order 30 milliseconds are now achieved. The basic technique potentially is capable of greatly expanded capacity and access times on the order of 10 milliseconds.

Additionally, this equipment has operationally demonstrated a useful capability to translate between languages, a desirable characteristic if successfully exploited for solution of man-machine communication problems.

The AN/GYA-IBM 7090 Relationship:

Two basic methods of connecting the photostore to an IBM 7090 as an auxiliary program segment store (as well as a possible lexical processor) were examined. In each approach, hardware methods of program relocation and memory protection were assumed to be possible of achievement, even though they were not included in the connection method study. Because the photostore is an asynchronous and independently operable device some data buffer-storage was required in the path of interconnection between the photostore and the IBM 7090. A maximum possible length of program segmentation of one thousand words was assumed and a minimum buffer size of two thousand (36 bit) computer words was assumed (on the basis of providing an alternate program segment choice to the IBM 7090). Further consideration of possible uses of the photostore-buffer combination in conditional table-look-up processing of data (such as this combination serves in the AN/GSQ-16) may make a larger buffer desirable, at least for experimentation purposes.

The two methods of interconnection which were examined for the foregoing purposes were (1) Direct Data Channel Connection and (2) Shared Memory Connection. The first of these connections was finally considered as the best, although each method of connection

is briefly discussed below.

Direct Data Channel Connection - This method of connection was discussed in Volume III of the second quarterly report on AF19 (626)-10. In review, an IBM 7607 Model II Data Channel would be modified to transmit to the IBM 7090, a maximum of one 36 bit word every 6.54 microseconds (based upon no other I/O channel activity.) With three I/O channels on a complex using I/O cycle times on a maximum basis, the slowest rate of transfer of data retrieved from the photostore would be about one 36 bit word every 54 microseconds. A "nominal" figure of 1.2 bits per microsecond (one word every 14 machine cycles) transfer rate can be assumed. This modified data channel connection is referred to as the Direct Data Channel Connection. Of considerable significance, with respect to this connection, is the fact that in transferring a block of data or program words from the photostore to the high speed working memory, only one machine cycle per 36 bit word is required (even though this may occur every 14 machine cycles).

Shared Memory Connection - Engineering design has been accomplished that allows connection of an additional 7302 core memory to an IBM 7090, that effectively provides a 64 thousand word total (36 bit word) high speed core memory for the IBM 7090. With this added memory attachment the necessary hardware modifications to the IBM 7090 are provided such that at the choice of the operator (or the program) the computer can operate using either, or both, of the memories. If one now considers the sentence analyzer memory, of the AN/GSQ-16 Lexical Processor (a smaller

core memory having electrical characteristics similar to those of the IBM 7090 core memory) as being used as this additional added memory, then the IBM 7090 and the Photostore would communicate via the placement of data look-up addresses (entries) or retrieved data in this common additional memory.

The two methods of connection briefly outlined above, each has particular advantages and disadvantages, particularly when examined from the point of view of each of the various intelligence data processing requirements such as lexical processing, information retrieval, and "on demand" service. Since the latter objective is a primary goal of this R & D program the following factor strongly influenced the final decision to exploit the Direct Data Channel connection. In the Shared Memory Connection, operation on program words retrieved from the Photostore would require in many cases the transfer of program instruction words from the shared memory into the main core memory of the IBM 7090. This operation would require the employment of a minimum of six machine cycles and possibly up to 12 machine cycles for each instruction word transferred. This appeared to be a serious "overhead" to pay in terms of useful machine cycles. In addition to the foregoing, the future utility of a connecting design which involved a "non-standard" and technically intimate connection between two specific memory structures was considered suspect, whereas the Direct Data Channel Connection offered greater flexibility of "tie-in" with variations of structure in the IBM 7000 series computers (such as the higher speed IBM 7094 Data Processing System which is compatible with the IBM 7090 Data Processing System).

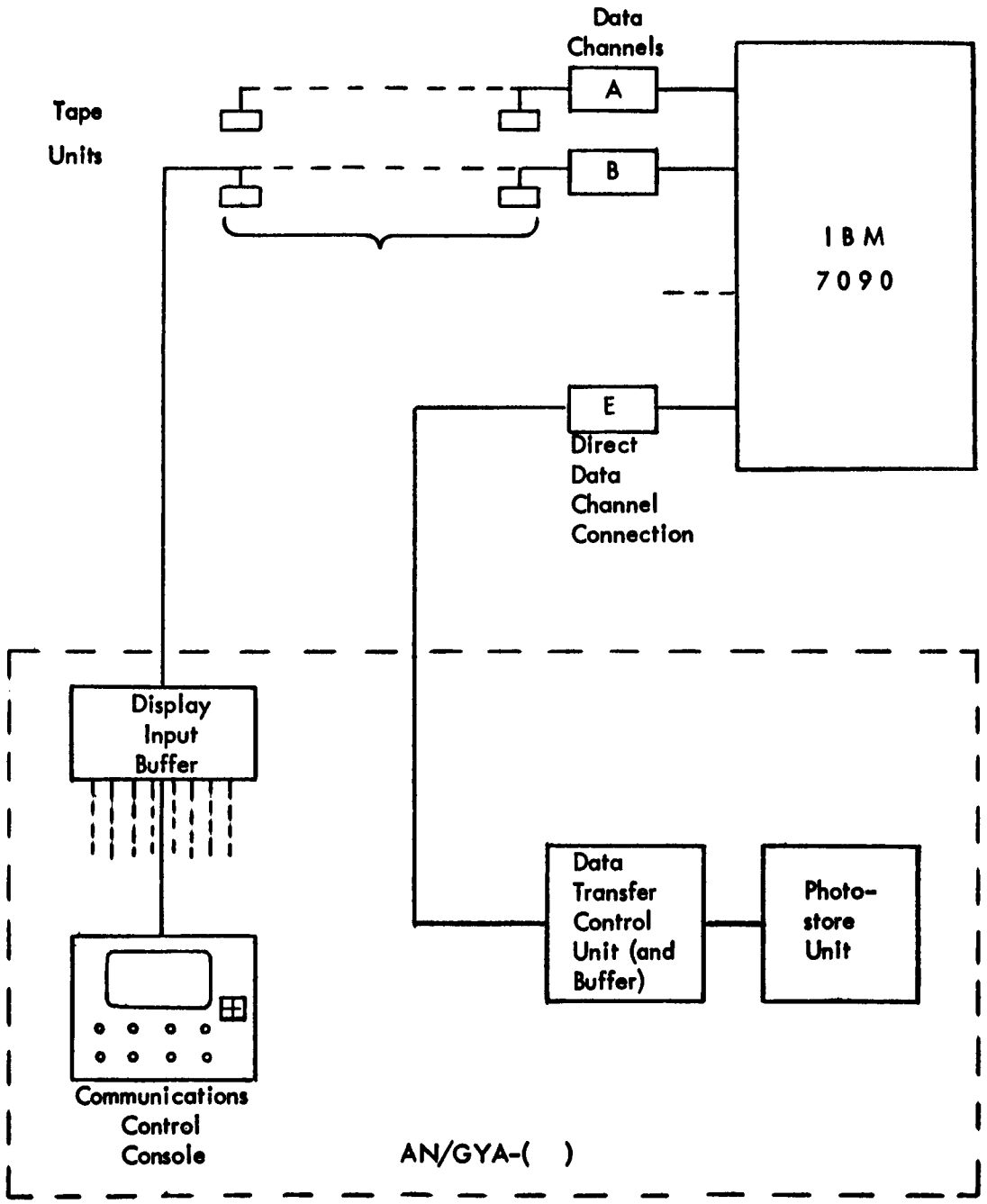
To implement the basic equipment group (the AN/GYA-( )) to reach the goals of this program, a more flexible method of communicating between the computer complex and the "on demand" user was required. Fortunately, the U. S. Air Force has previously sponsored development of a Communications Control Console, capable of being connected to an IBM 7090 tape unit connection. Although the eventual optimum characteristics of such a console for "on demand" use in a computer complex are not presently known, this device which has typewriter key-board, selected operation keys, and cathode ray tube display, should provide a good experimental capability.

The basic configuration of the AN/GYA-( ) Computer Storage Integration Group which will be developed and integrated with an IBM 7090 are shown in figure 1.

References

Reference 1: "Some Theoretical Aspects of the Mechanization of Literature Searching" by Y. Bar-Hillel, Technical Report No. 3, April 1960, Contract No. N62558-2214 ONR, ASTIA Document No. 236772 (Specifically Page 51, section 18).

Reference 2: "Real-Time, Time-Shared Computer Project" Contract No. NONR-1841 (69) DSR #8644, P. M. Morse, Principal Investigator H. M. Teager, Project Head (1st Quarterly Report, January 3, 1961).



THE COMPUTER STORAGE
INTEGRATION GROUP

Figure 1

Appendix B

Optimum Search Strategies

Introduction

We consider a store consisting of N cells, with information stored in tabular form. That is, the record r(i) stored in cell i is in the form of argument-function x_i $f(x_i)$, the file being arranged in ascending order of the argument x. Given a particular argument x, one proceeds in searching for the cell containing xf(x) by comparing x against the arguments in a sequence of cells $i_1, i_2, \ldots$ This sequence of cells is to be chosen so that the average number of comparisons needed to locate the correct cell is a minimum.

Assumptions

(1) We assume that for a comparison of x against x_i, one of the following three outcomes is possible:

$x > x_i$, $\quad x < x_i$, $\quad x = x_i$.

(2) Let ξ be a random variable denoting the location of x. We assume the à priori distribution p_k is given, where:

$$p_k = \text{Prob}\left[\xi = k\right] \qquad \sum_{k=1}^{N} p_k = 1 \qquad (1)$$

(3) Let S be the set of integers 1 through N, and let σ be any non-empty subset of S. We assume that the à posteriori probability distribution of ξ is unchanged except for normalization, i.e.,

$$\text{Prob}\left[\xi = k \mid \xi \in \sigma\right] = \frac{p_k}{P(\sigma)}, \quad k \in \sigma \tag{2}$$

where

$$P(\sigma) = \sum_{i \in \sigma} p_i \tag{3}$$

Formulation

Let $T\left[(p_k), N\right]$ formally denote the minimum average number of comparisons necessary, given the distribution (p_k) and the number of cells N. It is to be emphasized that T is not a function of p_k per se, but is rather a functional of the distribution, i.e., the whole set of p_k's. It is clear that the first step is to select a cell for the first comparison. Suppose we select cell n, then the following situation results from the comparison:

(a) There is a probability p_n that $x = x_n$ and the search terminates.

(b) There is a probability $P_{n-1} = \sum_{i=1}^{n-1} p_i$ that $x_n > x$, then x must be contained somewhere in the first n-1 cells. Now if we renumber the cells backwards starting with n-1 as the first cell, the new distribution becomes:

$$p'_k = \frac{p_{n-k}}{P_{n-1}} \tag{4}$$

(c) There is a probability $1-P_n = \sum_{i=n+1}^{N} p_i$ that $x_n < x$, and upon renumbering the new distribution becomes:

$$p''_k = \frac{p_{n+k}}{1-P_n} \tag{5}$$

Now, whatever the choice of the first comparison is succeeding choices must remain optimum for the overall choice to be optimum. Therefore, T must satisfy the following minimization functional equation.

$$T\left[(P_k),\ N\right] = \min_{1 \le n \le N} \left\{ 1 + P_{n-1}\, T\left[\left(\frac{P_{n-k}}{P_{n-1}}\right),\ n-1\right] + (1-P_n)\, T\left[\left(\frac{P_{n+k}}{1-P_n}\right),\ N-n\right]\right\} \quad (6)$$

Equation (6) is a dynamic programming equation [1] yielding as solutions the functional $T(.,.)$, and $n\left[(p_k), N\right]$ which minimizes the right hand side. As initial conditions we set $P_o\ T(.,o) = o$ and $T(.,1) = o$.

The following distributions are of particular interest:

(a) $P_k = \frac{1}{N}, \quad k = 1,\cdots,N.$

(b) $P_k = \frac{N(1-\beta)-(1+\beta)+2\beta k}{N(N-1)}, \quad -1 \le \beta \le 1 .$

(c) $P_k = \binom{N-1}{k-1} \gamma^{k-1} (1-\gamma)^{N-k}, \quad 0 < \gamma < 1 .$

(d) $P_k = A\frac{1}{k^{\alpha}}, \quad 1 < \alpha < 2 .$

The first is obviously the uniform distribution. The second is the distribution with constant slope. The third is the binominal distribution and the fourth Yule distribution. Thus far, Eq. (6) is completely solved only for (a).

Solution for Uniform Distribution

For $p_k = \frac{1}{N}$, Equation (b) reduces to

$$T(N) = 1 + \min_{1 \le n \le N} \left\{ \frac{n-1}{N} \; T(n-1) + (1 - \frac{n}{N}) \; T(N-n) \right\}. \quad (7)$$

If we let $f(N) = NT(N)$, then Equation (7) becomes:

$$f(N) = N + \min_{1 \le n \le N} \left[f(n-1) + f(N-n) \right]. \quad (8)$$

It can be shown that the solutions are as follows: (See Appendix 1)

$$f(2^{k+1}+2m-1) = 2^{k+1}(k-\frac{1}{2}) + 2mk + 3m+1, \quad \begin{matrix} k=0, 1, \ldots \\ m=0, 1, \ldots 2^k \end{matrix} \quad (9)$$

$$f(2^{k+1}+2m) = 2^{k+1}(k-\frac{1}{2}) + (2m+1)k + 3(m+1), \quad \begin{matrix} k=0, 1, \ldots \\ m=0, 1, \ldots 2^k-1 \end{matrix} \quad (10)$$

The policy solution n(N) which yields the minimum is not unique. In fact, the multiplicity of solutions for large N is quite large. The complete set of solution are as follows:

$$n(2^{k+1}+2m) = 2^k + j, \quad \begin{matrix} j=0, 1, \ldots 2m+1, & m \le 2^{k-1} \\ j=2m-2^k+1, \ldots, 2^k, & m \ge 2^{k-1} \end{matrix} \quad (11)$$

$$n(2^{k+1}+2m-1) = 2^k + 2j, \quad \begin{matrix} j=0, 1, \ldots, m, & m \le 2^{k-1} \\ j=m-2^{k-1}, \ldots, 2^{k-1}, & m \ge 2^{k-1} \end{matrix} \quad (12)$$

For example, consider $N = 2^4+6 = 22$.

$n(22) = 8, 9, 10, 11, \ldots, 15.$

Similarly, n(23) is given by

$n(23) = 8, 10, 12, 14, 16.$

Linear Distribution

For $p_k = \frac{2\beta}{N(N-1)} \left[k - \frac{N+1}{2} \right] + \frac{1}{N}$, Eq. (6) becomes

$$T(\beta,N) = 1 + \min_{1 \le n \le N} \left\{ P_{n-1}(\beta,N)\, T\left[f_{n-1}(-\beta,N),\ n-1 \right] + P_{N-n}(-\beta,N)\, T\left[f_{N-n}(\beta,N),\ N-n \right] \right\} \tag{13}$$

where $$P_n(\beta,N) = \frac{n}{N} \left[1 - \beta \left(\frac{N-n}{N-n} \right) \right] , \tag{14}$$

and $$f_n(\beta,N) = \frac{(n-1)\beta}{N-1+(N-n)\beta} \tag{15}$$

It can be shown that $T(\beta, N)$ as a function of β is piece wise linear, i. e.

$$T(\beta, N) = a(\beta, N) - b(\beta, N) \left| \beta \right| , \tag{16}$$

where $a(\beta, N)$ and $b(\beta, N)$ are constants over intervals of β.

It is also clear that:

$$n(\beta, N) \geq \max\left\{ n(0, N) \right\} , \quad \beta > 0 , \tag{17}$$

similarly, since $n(\beta, N) = N - n(-\beta, N)$,

$$n(\beta, N) \leq \min \left\{ n(0, N) \right\} , \quad \beta < 0 , \tag{18}$$

where maximization and minimization are taken over the set of multiplicity of solution for $\beta = 0$ (uniform distribution). From

earlier result we find that:

$$\max\left\{n\,(2^{k+1}+2m)\right\} = 2^k + 2m + 1, \quad m < 2^{k-1}, \qquad (19)$$

$$= 2^{k+1}, \quad m \geq 2^{k-1},$$

$$\max\left\{n(2^{k+1}+2m-1)\right\} = 2^k + 2m, \quad m \leq 2^{k-1}, \qquad (20)$$

$$= 2^{k+1}, \quad m \geq 2^{k-1}$$

Now, instead of seeking the optimum solution to Eq. (6), we can obtain a near-optimum (for small β) solution by using Eqs. (19) and (20) in Eq. (6). It can be shown that the near-optimum solution $To(\beta, N)$ is of the form

$$To(\beta, N) = \frac{1}{N} f(N) - |\beta| \frac{1}{N(N-1)} g(N), \qquad (21)$$

where $f(N)$ is given by Eqs. (9) and (10) and $g(N)$ satisfies the functional equation:

$$g(N) = (N-n+1)\, f(n-1) - n\, f(N-n) + g(n-1) + g(N-n). \qquad (22)$$

In Eq. (22), n is given by Eqs. (19) and (20). The function $g(N)$ for some specific values of N can easily be obtained. For example,

$$g(2^{k+1}-1) = 2^k f(2^k-1) - 2^k f(2^k-1) + 2g(2^k-1), \qquad (23)$$

whence $g(2^{k+1}-1) = 0$, since $g(1) = 0$. However, general solution of $g(N)$ have not yet been obtained.

Entropy Measure

It is often argued c.f. 2 on intuitive grounds that the entropy measure $-\sum_{k=1}^{N} p_k \log_2 p_k$ represents approximately the

mimimum average number of comparisons. The validity of this approximation can be examined using Eq. (6). To do this, we change assumption (1) slightly, and restrict the outcomes to a comparison to $x \leq x_i$ and $x > x_i$ thus eliminating the possible outcome $x = x_i$.

As a result, Eq (6) is modified and becomes:

$$T\left[(p_k),\ N\right] = \min_{1 \leq n \leq N} \left\{ 1 + P_n\, T\left[\left(\frac{P_n - k + 1}{P_n}\right),\ n\right] + (1 - P_n)\, T\left[\left(\frac{P_n + k}{1 - P_n}\right),\ N - n\right] \right\}. \tag{24}$$

Now, if we substitute for $T[\cdot, \cdot]$ in Eq. (24)

$$T\left[(p_k),\ N\right] = -\sum_{k=1}^{N} p_k \log_2 p_k, \tag{25}$$

Eq. (24) becomes:

$$-\sum_{k=1}^{N} p_k \log_2 p_k = \min_{1 \leq n \leq N} \left\{ 1 - \sum_{k=1}^{n} p_k \log_2 p_k + P_n \log_2 P_n - \sum_{k=n+1}^{N} p_k \log_2 p_k + (1 - P_n) \log_2 (1 - P_n) \right\} \tag{26}$$

or

$$\min_{1 \leq n \leq N} \left\{ 1 + P_n \log_2 P_n + (1 - P_n) \log_2 (1 - P_n) \right\} = 0 \tag{27}$$

Equation (27) is satisfied if and only if

$$P_n = 1/2 \tag{28}$$

In general, there is no integer n which satisfied Eq. (28). Furthermore, even if there is an integer n which satisfies Eq. (28), in general Eq (25) is still not a solution due to the recursive nature

of Eq. (24). In order for Eq. (25) to be the optimum solution, not only must there be an n such that

$$\sum_{k=1}^{n} p_k = \sum_{k=n+1}^{N} p_k = 1/2$$

but there must also be integers ℓ and m such that

$$\sum_{k=1}^{\ell} p_k = \sum_{k=\ell+1}^{n} p_k = \sum_{k=n+1}^{m} p_k = \sum_{k=m+1}^{N} p_k = 1/4$$

and so on. For large N and under some regularity conditions on p_k, it is very likely that for the first few comparisons partitioning into equally probable subsets can be accomplished. However, it is not clear that in what precise sense can Eq. (25) be regarded as an approximate solution.

Discussion and Conclusion

From a purely theoretical point of view, the formulation of the search problem as a functional equation involving minimization is quite attractive. First, it has led to the complete solution in one simple case and a detailed examination of some approximate solution in a second case. Secondly, this approach has permitted a precise analysis of the entropy measure as an approximate solution for the minimum average number of comparisons. Nevertheless, from the point of view of application, the purely theoretical approach has serious limitations. First, the formulation requires that the a priori distribution be known. An accurate knowledge of the distribution is probably difficult in most practical situations. However, it should be noted that this difficulty is an inherent one and not due to the particular formulation chosen. It is always possible to devise a strategy which is distribution free, such as binary search or serial search, in which case it almost always means that our information concerning what has been stored is not fully utilized. A second limitation is that implementation of an optimum strategy would almost surely involve some computation time. If the computation time becomes significant in comparison with the average search time, then it is doubtful that the total time required will be a minimum.

Despite the above reservations, the importance of a rigorous theoretical approach should not be underestimated. Its principal usefulness is in establishing a standard of performance against which any practical strategy of search can be measured. In this regard, the most important open problem is to establish both upper and lower bounds for the minimum mean search time (or minimum number of comparisons). It has been shown that the entropy measure is in some sense an approximate solution. However, in the general case where the outcome to each comparison is "greater than", "equal to" or "smaller than", the entropy measure is neither an upper nor a lower bound.

Future Research

As the design of system for the integrated complex progresses, a number of immediate storage and searching problems has arisen or can be anticipated. A few of these are briefly formulated below.

A. The storage apportionment problem - it frequently occurs that a large amount of information has to be stored, some of which are to be retrieved more frequently than others. The question then arises as to what proportion of the information should be stored in a rapid-access, high-cost memory (e.g. core) and what fraction

should be stored in a slow-access, low-cost memory (e. g. tape) to achieve a favorable balance of cost and speed. Although the general problem is open, a great deal can be inferred from existing results. The following example will serve to illustrate the approach that can be used. Suppose that a dictionary containing N entries is to be stored in a combination of core storage and tape. Assume that the entries are arranged in decreasing order of frequency of use such that the j^{th} entry has a frequency of use $\sim \frac{1}{j}$ (Yule Distribution or Zipf's Law). The problem is now to determine the average time required per look-up and the total cost of storage, if n of the N entries are stored in core and the remainder on tape. Under some simplifying assumptions, this problem can be solved. Assume that the n entries in core are sorted so that a binary search can be used. It is estimated that for each comparison cycle of a binary search, approximately 15 machine cycles or 30 μsec 7090 time is required. Assume that the information on tape is again sorted with a record length of k entries. This immediately implies that an additional core storage for k entries is required. It is assumed that entries are read off the tape in blocks of k, and a binary search is then used on the k entries. Each entry is assumed to be 5 machine words long (180 bits). The approximate average time per

lookup is easily calculated to be:

$$T \cong .03 \log_2 n \frac{\sum_{j=1}^{n} \frac{1}{j}}{\sum_{j=1}^{N} \frac{1}{j}} + (7.3 + .5k + .03 \log_2 k)\left(\frac{N-n}{2k}\right)\frac{\sum_{j=n+1}^{N} \frac{1}{j}}{\sum_{j=1}^{N} \frac{1}{j}}$$

in milliseconds, where a tape reading time of (7.3 + .1 x number of wds/rec) milliseconds has been assumed. If we use the approximation

$$\sum_{j=1}^{n} \frac{1}{j} \approx \ln n + \gamma \quad (\gamma \approx .58 \text{ Euler's constant}),$$

then the expression for T simplifies to be

$$T \cong \frac{.03 \log_2 n (\ln n + \gamma)}{\ln N + \gamma} + \left(\frac{N-n}{2k}\right) \frac{(7.3 + .5k + .03 \log_2 k)}{\ln N + \gamma} \frac{(\ln N - \ln n)}{}$$

The total cost of storage is approximately

$$C = \beta_1 (n + k) + \beta_2 (N - n),$$

where β_1 and β_2 are the costs per 180 bits of core storage and tape storage respectively. To reduce T, one can increase either n or k or both, of course, at a greater cost. Since T is nonlinear with respect to both n and k, while C is linear, speed is not simply proportional to cost. The choice of n and k must be carefully considered. As this example indicates, the large number of problems in the area of apportionment and selection of storage are amenable to theoretical analysis and should be pursued.

B. A second area where work is required is when the cost of maintaining a sorted file is too high. However, this does not mean that completely random storage has to be resorted to. Between a sorted file(where order is almost complete)and a completely random file, there is a wide spectrum of policies of filing and searching which preserve most of the structure and order inherent in the information to be stored at a fraction of the cost of maintaining a sorted file. The open addressing system described in detail by Peterson (Ref. 2) is an example of such a policy. Another example is where items to be stored can be labeled by an originating date (e.g., news items, periodicals). A simple policy would be to store the items in the order of accession date. A knowledge of the distribution of the gap between the originating date and the accession date can be used in designing search strategies which permit rapid access to the file.

REFERENCES

(1) R. Bellman, Dynamic Programming , Princeton, Princeton University Press, 1957.

(2) W. W. Peterson, "Addressing for random-access storage", IBM Journal of Research and Development, vol. 1 (1957), pp. 130-146.

Optimum Solutions for Uniform Distribution

In this appendix, we shall prove that the solutions of Eq. (8) are given by Eqs. (9) through (12). The proof proceeds in three stages. First, we shall show that the right hand side of Eq. (8) is minimized by a specific set of values of n. Next, the solution for f(N) will be given. Finally, the multiplicity of the policy solution n(N) will be derived.

A, We begin by proving the following theorem:

Theorem 1, given the conditions $f(0) = f(1) = 0$, the equation

$$f(N) = N + \min_{1 \leq n \leq N} \left[f(n-1) + f(N-n) \right] , \tag{8}$$

is satisfied if $n = n^*(N)$ where

$$n^*(4m-2) = n^*(4m-1) = n^*(4m) = n^*(4m+1) = 2m, \quad m = 1, 2, \ldots \tag{A.1}$$

Proof: It is easily seen that the theorem is equivalent to the following set of equations for m ranging over all positive integers:

$$f(4m-2) = 4m - 2 + f(2m-2) + f(2m-1), \tag{A.2a}$$

$$f(4m-1) = 4m - 1 + f(2m-1) + f(2m-1), \tag{A.2b}$$

$$f(4m) = 4m + f(2m-1) + f(2m), \tag{A.2c}$$

$$f(4m+1) = 4m+1 + f(2m-1) = f(2m+1), \tag{A.2d}$$

The proof proceeds by induction. First, it is easily shown by enumerating all possibilities in Eq. (8) that Eq. (A. 2) holds for m = 1. Now assume Eq. (A. 2) to hold for m = 1, 2, . . . ,K, it will be shown that the validity of Eq. (A. 2) for m = K + 1 follows. The main part of

the proof of the theorem will need the following Lemma.

Lemma: The validity of (A. 2) for $m = 1, 2, \ldots, K$, implies the following inequalities:

$$f(n+1) - f(n) \geq f(n-1) - f(n-2), \quad n = 2, 3, \ldots, 4K, \tag{A. 3a}$$

$$f(n+1) > f(n), \quad n = 1, 2, \ldots, 4K, \tag{A. 3b}$$

$$f(2n) - f(2n-1) > f(2n+1) - f(2n), \quad n = 1, 2, \ldots, 2K. \tag{A. 3c}$$

If we subtract (A. 2c) from (A. 2d) and (A. 2a) from (A. 2b), we see that

$$f(4m+1) - f(4m) > f(4m-1) - f(4m-2)$$

$$> f(4m-3) - f(4m-4) > \cdots\cdots\cdots \geq 0, \quad m = 1, \ldots, K.$$

Similarly,

$$f(4m) - f(4m-1) > f(4m-2) - f(4m-3) = f(4m-4) - f(4m-5)$$

$$> f(4m-6) - f(4m-7) \cdots\cdots > 0 \qquad m = 2, \ldots, K. \tag{A. 5}$$

Thus, Eqs. (A. 3a) and (A. 3b) follow immediately from Eqs. (A. 4) and (A. 5). Equation (A. 3c) can be proved by indication. If Eq. (A. 2) is valid for $m = 1, \ldots, K$, then

$$f(2m) - f(2m-1) > f(2m+1) - f(2m), \qquad m = 1, 2, \ldots, K$$

implies

$$f(4m-2) - f(4m-3) > f(4m-1) - f(4m-2)$$

and

$$f(4m) - f(4m-1) > f(4m+1) - f(4m), \qquad m = 1, 2, \ldots, K$$

Therefore, the fact that $f(2) - f(1) > f(3) - f(2)$ implies Eq. (A. 3c).

Now $f(4K + 2)$ can be written by Eq. (8) as

$$f(4K+2) = 4K+2 + \min_{1 \leq n \leq 4K+2} \left\{ f(n-1) + f(4K+2-n) \right\} \tag{A.6}$$

By symmetry and and Eq. (A. 3b), Eq. (A. 6) can be rewritten

$$f(4K+2) = 4K+2 + \min_{2 \leq n \leq 4K+1} \left\{ f(n-1) + f(4K+2-n) \right\} \tag{A.7}$$

$$= 4K+2 + \min \left\{ \min_{1 \leq n \leq K} \left[f(2n-1) + f(4K+2-2n) \right], \min_{1 \leq n \leq K} \left[f(2n) + f(4K+1-2n) \right] \right\}$$

Now by Eq. (A. 3a), Eq. (A. 7) becomes

$$f(4K+2) = 4K+2 \min \left\{ \left[f(2K+2) + f(2K-1) \right], \left[f(2K) + f(2K+1) \right] \right\}$$

$$= 4K + 2 + f(2K) + f(2K + 1). \tag{A.8}$$

Now, Eq. (A. 2a) becomes valid for $m = 1, 2, \ldots, K + 1$, and Eq. (A. 3a) is valid for $n = 1, \ldots, 4K + 1$.

Similarly, by the use of Eqs. (A. 3a) and (A. 3b) $f(4K + 3)$ can be written

$$f(4K + 3) = 4K+3 + \min \left\{ 2f(2K+1), f(2K) + f(2K + 2) \right\}. \tag{A.9}$$

Now it follows from (A. 3c) that

$$f(2K + 2) - f(2K + 1) > f(2K + 3) - f(2K + 2),$$

and it follows from Eq. (A. 3a) that

$$f(2K + 3) - f(2K + 2) \geq f(2K + 1) - f(2K).$$

Therefore,

$$f(2K + 2) + f(2K) > 2f(2K + 1),$$

and

$$f(4K + 3) = 4K + 3 + 2f(2K + 1), \qquad \text{(A. 10)}$$

thus extending Eq. (A. 2b) to m = K + 1.

By following a procedure nearly identical to the above, it can be shown that

$$f(4K + 4) = 4K + 4 + f(2K + 1) + f(2K), \qquad \text{(A. 11)}$$

and

$$f(4K + 5) = 4K + 5 + f(2K + 1) + f(2K + 3). \qquad \text{(A. 12)}$$

By induction, Eq. (A. 2) is valid for all integral m, thus proving the theorem.

B, Thus far, it has only been shown that certain choices of n yield the minimum for the right hand side of Eq. (8). Nothing has been said about what form f(N) might take. This will be taken care of by the following theorem:

Theorem 2: Equation (8) is satisfied <u>if and only if</u>

$$f(2^{k+1} + 2m - 1) = 2^{k+1}(k - \tfrac{1}{2}) + 2mk + 3m + 1, \quad k = 0, 1, \ldots, \quad m = 0, 1, \ldots 2^k, \qquad \text{(B. 1a)}$$

$$f(2^{k+1} + 2n) = 2^{k+1}(k - \tfrac{1}{2}) + (2m+1)\,k + 3m + 3, \quad k = 0, 1, \ldots, \quad m = 0, 1, \ldots, 2^k - 1. \qquad \text{(B. 16)}$$

Proof: The "only if" (or uniqueness) part of theorem 2 is trivial and will be taken care of first. Suppose there are two solutions $f_1(N)$

and $f_2(N)$. If $f_1 > f_2$, then f_1 cannot be a solution of Eq. (8). Similarly, f_2 cannot be a solution if $f_2 > f_1$. Thus, the solution f(N) of Eq. (8) must be unique.

To prove Eq. (B.1), we can again employ induction. That is; we can easily verify that Eqs. (B.1) is valid for k = 0, in addition, we assume Eq. (B.1) to be valid for $k = 0, 1, \ldots, K-1$. If, thereby, the validity of Eq. (B.1) for k = K follows, then Eq. (B.1) must be valid for all k. The complete proof is a simple exercise in substituting Eq. (B.1) in Eq. (A.2) with elementary manipulations, and will not be carried out here.

C, Theorem 1 is strengthened by the following theorem:

Theorem 3: Equation (8) is satisfied by $n = n^*(N)$, i.e.,

$$f(N) = N + f\left[n^*(N) - 1\right] + f\left[N - n^*(N)\right], \qquad (C.1)$$

<u>if and only if</u>

$$n^*(2^{k+1} + 2m) = 2^k + j, \quad j = 0, 1, \ldots, 2m+1, \quad 0 \leq m < 2^{k-1} \qquad (C.2a)$$

$$j = 2m - 2^k + 1, \ldots, 2^k, \quad 2^{k-1} \leq m \leq 2^k - 1,$$

$$n^*(2^{k+1} + 2m - 1) = 2^k + 2j, \quad j + 0, 1, \ldots, m, \quad 0 \leq m \leq 2^{k-1} \qquad (C.2b)$$

$$j = m - 2^{k-1}, \ldots, 2^{k-1}, \quad 2^{k-1} \leq m \leq 2^k.$$

Proof: The "if" part of the theorem is easily proved by simply substituting Eqs. (C.2) and (B.1) in (C.1) and verify. In the process, it is also easily shown that for $2^{k+1} \leq N \leq 2^{k+2} - 1$, the only solutions for the range $2^k \leq n^* \leq 2^{k+1}$ are those given by

Eq. (C. 2). Therefore, it only remains to show that no value of n^* greater than 2^{k+1} or less than 2^k satisfies Eq. (C. 1).

Consider $N = 2^{k+1} + 2m$, $0 \leq m < 2^k - 1$. Since we know that $n^* = 2^k$ is a solution, we need only to prove that (similar results follow for $n^* > 2^{k+1}$ by symmetry)

$$f(2^k-1) + f(2^k + 2m) < f(2^k-2) + f(2^k + 2m+1) \leq f(2^k-3) + f(2^k + 2m+2) \leq \ldots \quad (C.3)$$

Inequality (C. 3) follows immediately from (A. 4)§. The proof for $N + 2^{k+1} + 2m$, $2^k - 1 \leq m \leq 2^k$, is equally simple. By using $n^* = 2^{k+1}$, it is found that

$$f(2^{k+1}-1) + f(2m) < f(2^{k+1}) + f(2m-1) \leq f(2^{k+1}+1) + f(2m-2) \leq \ldots \quad (C.4)$$

which follows from (A. 5). The proof for $N = 2^{k+1} + 2m - 1$ follows completely similar lines and will not be reproduced here.

§In fact, a stronger inequality with strictly unequal signs throughout follows from (A. 4).

Appendix C

Sampling for Co-occurrence

1. In business and military intelligence as well as in the retrieval of information from documented knowledge, it is quite often necessary to determine the number of items that co-occur in two or more lists of items. These items may be the names of individuals, titles of documents, "key words", etc. The lists of items, though finite, may often be too long to permit exhaustive matching for co-occurrence, without making the cost of this operation prohibitive. Thus, it is of interest to explore the possibility of applying sampling techniques to estimate the number of co-occurring items.

As a specific case of the problem of co-occurrence, let us suppose that we have lists of names of individuals on the basis of their professional specialities such as Mathematics, Engineering, Physics, etc. In addition, let us assume that associated with each name is a short biography. It might be of interest to obtain the number of individuals in the various lists who might have been working together on some specific project or might have been at a particular university during a specified time period. Quite often, it is only necessary to obtain an estimate of this number than its exact value. In any man-machine information retrieval system, it may often prove to be advantageous to obtain a reasonable estimate of the number of co-occurrences before a display by cathode ray tube or any other hardware, since such displays may be very expensive and one might want to reduce the number of items to be displayed by adding new requirements. Thus in the specific problem mentioned

previously, if the number of people who worked on a particular project are too numerous, and the properties they should have in common other than the one specified are known, then by requiring these further properties to be satisfied, the size of the collection of co-occurring items can be reduced until at such stage display is economically feasible. Thus at each stage, we can obtain an estimate of the number of items and stop sampling when the desired size of display has been obtained.

Having given a rather brief account of the reasons for sampling to determine co-occurrences, we shall consider the statistical aspects of the sampling problem. There are two situations with which we shall be concerned. They are:

1. The finite populations are unstratified and the sampling is simple random sampling.

2. The finite populations are stratified and the sampling is stratified sampling. Stratification is economical in such cases as alphabetical listings.

This problem had received some attention earlier. Leo Goodman (2) discussed simple random sampling and derived the unbiased estimator for the number of items common to two or more lists of items. The approximate variance of the estimator for small sampling fractions had been indicated. We shall obtain the exact moments of the unbiased estimator. This will enable us to discuss the skewness and excess of the distribution of the estimator. Using the probability distribution of the estimator we can show computationally that the unbiased estimator is indeed the maximum likelihood estimator. We shall define unbiased esimator for stratified sampling and indicate several interesting aspects of this type of estimation.

As shown by Goodman, this is one situation where insufficient statistics give minimum variance and unbiasness. As he indicates, sometimes the estimator has unreasonable values and adjustments have to be made so that the nearest reasonable estimate is used. This naturally will introduce bias.

We shall discuss the sampling problem with emphasis on pair wise co-occurrence but shall indicate natural extensions to more than a pair of populations.

2. Sampling Techniques

Assumptions:

1) There are r populations $U_1, U_2, \ldots, U_r$ with $N_1, N_2, \ldots, N_r$ units respectively. The units do not occur more than once in each population.
2) Samples of units will be drawn from $U_1, U_2, \ldots, U_r$, without replacement.
3) Sampling can be done in one or more stages.
4) Sampling from the populations or from the strata is random sampling.

A. Simple Random Sampling

Definition 1: Simple random sampling is a method of selecting n units out of a population of N units such that every one of the $\binom{N}{n}$ samples has an equal chance of being chosen. Sometimes this type of sampling is referred to as unrestricted random sampling in statistical literature. Assumptions 1) and 2) hold.

Sampling Scheme: Simple random samples $S_1, S_2, \ldots, S_r$ of sizes $n_1, n_2, \ldots, n_r$ are drawn from populations $U_1, U_2, \ldots, U_r$. The number of units, n_{ij} which samples S_i and S_j have in common with each other, are observed for $i \neq j$, $1 \leq i, j \leq r$.

Problems: If σ_{ij} are the number of units which U_i and U_j have in common, we want to estimate σ_{ij}, using an estimator $\hat{\sigma}_{ij}$ which is a real valued function of n_{ij}, n_i, n_j, N_i and N_j. The estimator $\hat{\sigma}_{ij}$ has to be unbiased and should be a minimum variance unbiased estimator. Further, the variance and higher moments of $\hat{\sigma}_{ij}$ have to be derived. The coefficients of skewness and excess have to be evaluated to study their effects on the predicative value of $\hat{\sigma}_{ij}$. Natural extensions to defining estimators of $\sigma_{ij..\ell}$, the number of units which populations $U_i, U_j, \ldots U_\ell$ have in common are investigated.

Analysis: We shall make the observation that σ_{ij}, σ_{ijk} etc. are values or constants.

Definition
$$\hat{\sigma}_{ij} = \frac{N_i N_j}{n_i n_j} \cdot n_{ij} \qquad (1)$$

Theorem 1: $\hat{\sigma}_{ij}$ as defined by $\hat{\sigma}_{ij} = \frac{N_i N_j}{n_i n_j} n_{ij}$ is an unbiased estimator of σ_{ij} and in general $\hat{\sigma}_{ij\ldots\ell} = \frac{N_i}{n_i}\frac{N_j}{n_j}\ldots\frac{N_\ell}{n_\ell} n_{ij\ldots\ell}$ is an unbiased estimator of $\sigma_{ij\ldots\ell}$.

Proof: We have to show that the expected vlaue of $\hat{\sigma}_{ij}$ written as $E(\hat{\sigma}_{ij})$

is equal to σ_{ij}.

We shall define a function $\bar{\delta}_{(i,j)}$ as follows:

$$\bar{\delta}_{\alpha(i,j)} = \begin{cases} 1 & \text{if the unit } \alpha \text{ appears in both } U_i \text{ and } U_j \\ 0 & \text{otherwise} \end{cases}$$

In addition we define a random variable $\delta_{\alpha(i,j)}$ as follows:

$$\delta_{\alpha(i,j)} = \begin{cases} 1 & \text{if the unit } \alpha \text{ appears in both } S_i \text{ and } S_j \\ 0 & \text{otherwise} \end{cases}$$

Obviously, $\sigma_{ij} = \sum_{\alpha} \bar{\delta}_{\alpha(i,j)}$ where the Σ is over all α in both U_i and U_j.

Further $n_{ij} = \sum_{\alpha}' \delta_{\alpha(i,j)}$ where summation Σ' is over all α appearing in both S_i and S_j.

$$E(n_{ij}) = \sum_{\alpha} E(\delta_{\alpha(i,j)})$$

and $$E(\delta_{\alpha(i,j)}) = \bar{\delta}_{\alpha(i,j)} \frac{\binom{N_i - 1}{n_i - 1}\binom{N_j - 1}{n_j - 1}}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

Thus $$E(n_{ij}) = \sum_{\alpha} E(\delta_{\alpha(i,j)}) = \sum_{\alpha} \bar{\delta}_{(i,j)} \frac{\binom{N_i - 1}{n_i - 1}\binom{N_j - 1}{n_j - 1}}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

$$= \frac{n_i \cdot n_j}{N_i \cdot N_j} \sum_{\alpha} \bar{\delta}_{\alpha(i,j)}$$

$$= \frac{n_i \cdot n_j}{N_i \cdot N_j} \cdot \sigma_{ij}$$

i.e.,

$$E\left\{\frac{N_i \cdot N_j}{n_i \quad n_j} \cdot n_{ij}\right\} = \sigma_{ij}$$

Definining the function $\bar{\delta}_{\alpha(i,j,\ldots,\ell)}$ and the random variable $\delta_{\alpha(i,j,\ldots,\ell)}$ for more than two populations, we can show that

$$E(n_{ij\ldots\ell}) = E\left\{\sum_{\alpha}' \delta_{\alpha(i,j,\ldots,\ell)}\right\}$$

$$= \sum_{\alpha} \bar{\delta}_{\alpha(i,j,\ldots)} \cdot \frac{\binom{N_i - 1}{n_i - 1}\binom{N_j - 1}{n_j - 1}\cdots\binom{N_\ell - 1}{n_\ell - 1}}{\binom{N_i}{n_i}\binom{N_j}{n_j}\cdots\binom{N_\ell}{n_\ell}}$$

$$= \frac{n_i \; n_j \cdots n_1}{N_i \; N_j \cdots N_1} \quad \sigma_{ij \ldots 1}$$

Thus, $$E\left\{ \frac{N_i \; N_j \cdots N_1}{n_i \; n_j \cdots n_1} \cdot n_{ij \ldots 1} \right\} = \sigma_{ij \ldots 1}$$

Theorem 2: The variance of $\hat{\sigma}_{ij}$, written as Var $(\hat{\sigma}_{ij})$ is equal to

$$\frac{N_i \; N_j}{n_i \; n_j} \sigma_{ij} - \frac{N_i \; N_j}{(N_i - 1)(N_j - 1)} \cdot \frac{(n_i - 1)(n_j - 1)}{n_i \; n_j} \sigma_{ij}$$

$$+ \sigma_{ij}^2 \left\{ \frac{N_i \; N_j}{(N_i - 1)(N_j - 1)} \frac{(n_i - 1)(n_j - 1)}{n_i \cdot n_j} - 1 \right\}. \qquad (4)$$

and in general

$$\text{Var}\,(\hat{\sigma}_{ij \ldots l}) = \frac{N_i \; N_j \ldots N_l}{n_i \; n_j \ldots n_l} \sigma_{ij \ldots l} - \frac{N_i \; N_j \ldots N_l}{(N_i - 1)(N_j - 1) \ldots (N_l - 1)} \frac{(n_i - 1)}{n_i} \cdot$$

$$\frac{(n_j - 1)}{n_j} \cdots \frac{(n_l - 1)}{n_l} \sigma_{ij} + \sigma_{ij}^2 \left\{ \frac{N_i \; N_j \ldots N_l}{(N_i - 1)(N_j - 1) \ldots (N_l - 1)} \frac{(n_i - 1)(n_j - 1) \ldots (n_l - 1)}{n_i \quad n_j \ldots n_l} - 1 \right\}$$

Proof:

$$\text{Var}\left(\hat{\sigma}_{ij}\right) = E\left(\frac{N_i N_j}{n_i n_j} \cdot n_{ij} - \sigma_{ij}\right)^2$$

$$= \frac{\sum\left(\frac{N_i N_j}{n_i n_j} n_{ij} - \sigma_{ij}\right)^2}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

where the $\sum$ is over all pairs of random samples of sizes n_i and n_j and $\frac{1}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$ is the probability for a pair of samples.

$$\text{Var}\left(\hat{\sigma}_{ij}\right) = \frac{\sum\left(\frac{N_i N_j}{n_i n_j}\right)^2 n_{ij}^2 - 2\sigma_{ij}\sum\left(\frac{N_i N_j}{n_i n_j}\right) n_{ij} + \sum \sigma_{ij}^2}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

Now

$$\frac{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sum n_{ij}^2}{\binom{N_i}{n_i}\binom{N_j}{n_j}} = \frac{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sum_{\alpha} \delta_{\alpha(i,j)} \binom{N_j - 1}{n_i - 1}\binom{N_j - 1}{n_j - 1}}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

$$+ 2\frac{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sum_{\alpha,\beta} \delta_{\alpha\beta(i,j)} \binom{N_i - 2}{n_i - 2}\binom{N_j - 2}{n_j - 2}}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

Since we have to consider the occurrence of single units such as α and the pairs such as α, β, in the samples in order to evaluate n^2_{ij}. Since $\delta_{\beta\alpha(ij)}$ and $\delta_{\alpha\beta(i,j)}$ are counted as one $\sum_{\alpha,\beta} \delta_{\alpha\beta(i,j)} = \frac{\sigma_{ij}(\sigma_{ij} - 1)}{2}$

and $\sum_{\alpha} \delta_{\alpha(ij)} = \sigma_{ij}$ as before. Further the number of samples in which α alone occurs in $\binom{N_i-1}{n_i-1}\binom{N_j-1}{n_j-1}$ and the number of samples in which α and β both occur is $\binom{N_i-2}{n_i-2}\binom{N_j-2}{n_j-2}$

Thus

$$\frac{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sum n_{ij}^2}{\binom{N_i}{n_i}\binom{N_j}{n_j}} = \frac{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sigma_{ij} \binom{N_i-1}{n_i-1}\binom{N_j-1}{n_j-1}}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

$$+ \frac{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sigma_{ij}(\sigma_{ij}-1) \binom{N_i-2}{n_i-2}\binom{N_j-2}{n_j-2}}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

$$= \frac{N_i N_j}{n_i n_j}\sigma_{ij} + \frac{N_i N_j (n_i-1)(n_j-1)(\sigma_{ij}^2 - \sigma_{ij})}{n_i n_j (N_i-1)(N_j-1)},$$

$$\frac{\sigma_{ij} \sum n_{ij} \binom{N_i N_j}{n_i n_j}}{\binom{N_i}{n_i}\binom{N_j}{n_j}} = \sigma_{ij}^2 \quad \text{from theorem 1.}$$

Thus $\mathrm{Var}(\hat{\sigma}_{ij}) = \frac{N_i N_j}{n_i n_j}\sigma_{ij} + \frac{N_i N_j (n_i-1)(n_j-1)}{(N_i-1)(N_j-1) n_i n_j}(\sigma_{ij}^2 - \sigma_{ij}) - 2\sigma_{ij}^2 + \sigma_{ij}^2$

$$= \frac{N_i N_j}{n_i n_j}\sigma_{ij} - \frac{N_i N_j (n_i-1)(n_j-1)}{(N_i-1)(N_j-1) n_i n_j}\sigma_{ij}$$

$$+ \sigma_{ij}^2 \left\{ \frac{N_i \cdot N_j (n_i-1)(n_j-1)}{(N_i-1)(N_j-1) n_i \cdot n_j} - 1 \right\}$$

The extension to evaluation of $\mathrm{Var}(\sigma_{ij\ldots l})$ of the above technique is obvious. This completes the proof.

Corollary 1

If $\frac{n_i}{N_i}$ and $\frac{n_j}{N_j}$ are small, then

$$Var\,(\hat{\sigma}_{ij}) \cong \frac{N_i\,N_j}{n_i\,n_j}\,\sigma_{ij} \qquad (6)$$

Corollary 2

When $n_i = N_i$ and $N_j = n_j$, $Var\,(\hat{\sigma}_{ij}) = 0$ as is to be expected.

Theorem 3: If for all the subsets of the set of r documents, σ's are defined and if M is a real valued function of the σ's, then there is at most one sample function m such that E(m) = M (This theorem is due to Goodman).

Proof: Excluding the empty set and the sets of single populations, there are $2^r - r - 1 = t$, subsets of the set of r populations. Let these t subsets be ordered and ranked so that they recieve the ranks 1, 2, . . . , t. Let s_j and σ_j be the estimator and its true value for the subset which receives the rank j in the ordering. The estimator need not be the particular one we have defined in previous theorems. The sample space consists of the subsets $\{[s_1, s_2, \ldots, s_t]\}$ of the t-dimensional Euclidian space. We order this subset by increasing values of s_t. For equal values of s_t we order the vectors by s_{t-1} and so on, so that for equal values of s_2, we order the vectors of s_1. We may thus describe the sample space as a sequence of

vectors

$$B_1 = \left[s_1(1),\ s_2(1),\ \ldots,\ s_t(1)\right]$$

$$B_2 = \left[s_1(2),\ s_2(2),\ \ldots,\ s_t(2)\right] \quad \text{etc. where } B_1 \text{ is}$$

the smallest ordered vector, B_2 the next smallest etc. To each sample vector, B_j, there is a population vector P_j given by

$$P_j = \left[\sigma_1(j),\ \sigma_2(j),\ \ldots,\ \sigma_t(j)\right] \quad \text{where as defined}$$

before $\sigma_i(j)$ corresponds to $s_i(j)$ for $i = 1, 2, \ldots, t$. Let $Pr(B_i/P_h)$ be the probability of obtaining sample vector B_i when P_h is the true population vector. Evidently $Pr\left\{B_i/P_h\right\} = 0$ if $i > h$, and $Pr(B_i/P_i) > 0$ for all i. Hence an unbiased estimator $m(B_i)$ of the population function $M(P_h)$, must be such that

$$\sum_{i=1}^{h} m(B_i)\ Pr(B_i/P_h) = M(P_h) \qquad (7)$$

for $h = 1, 2, 3, \ldots$ This necessary condition insures the uniqueness of $m(B_i)$ since $m(B_i)$ must satisfy the recursive formula (7). This completes the proof.

Theorem 4: $\hat{\sigma}_{ij} = \frac{N_i}{n_i}\frac{N_j}{n_j} n_{ij}$ is a minimum variance unbiased estimator of σ_{ij}. [This is Goodman's argument]

Proof: The proofs for Theorems 1, 2, and 3 indicate that if we want to

define an unbiased estimator, our best strategy is to use $\hat{\sigma}_{ij}$. This means that $\hat{\sigma}_{ij}$ is indeed the minimum variance unbiased estimator.

1. Confidence Limits

In computing the confidence limits for estimates, we usually assume that the estimator has a normal distribution. We shall make this assumption about the probability distribution of $\hat{\sigma}_{ij}$ and shall later discuss in detail, the magnitude of approximation involved in such an assumption. If the assumption holds, lower and upper confidence limits for σ_{ij} are given by

$$\hat{\sigma}_{ij} - z\,Var(\hat{\sigma}_{ij}) < \sigma_{ij} < \hat{\sigma}_{ij} + z\,Var(\hat{\sigma}_{ij}) \qquad (8)$$

where z is the value of the normal deviate corresponding to the desired confidence probability. The most common values of z are:

Confidence probability	.8	.9	.95	.99
z	1.28	1.64	1.96	2.58

If sample size is less than 30 and $n_i = n_j = n$, the probability points z may be taken from Student's t-table with $(n - 1)$ degrees of freedom.

Thus in our problem if $\frac{n_i}{N_i}$ are small

$$\hat{\sigma}_{ij} - z\sqrt{\frac{N_i N_j}{n_i n_j}}\,\sigma_{ij} < \sigma_{ij} < \hat{\sigma}_{ij} + z\sqrt{\frac{N_i N_j}{n_i n_j}}\,\sigma_{ij}$$

Thus if $\frac{n_i}{N_i} = \frac{n_j}{N_j} = a$ and if $\left|\sigma_{ij} - \hat{\sigma}_{ij}\right|$ has to be less

than ϵ with some specified confidence probability then

$$a^2 \geq (z \cdot \sigma_{ij})/\epsilon^2 \qquad (9)$$

Formula (9) gives the appropriate value of a , the sampling fraction a.

2. Validity of the Normal Approximation

"In the usual sampling problems sample means are the estimators and the finiteness of the second moment guarantees, by the Central Limit Theorem, that the sample means are normally distributed if the populations are infinite. Madow (1948) proved that for a large class of finite populations the distribution of the sample mean tends to normality even if the sampling fraction $\frac{n}{N}$ is not negligible and sampling is without replacement. From the study of theoretical distributions that are skewed and from the results of sampling experiments on actual skewed populations some statements can be made about what usually happens to confidence probabilities when we sample from positively skew populations. The sample size is assumed large enough so that the distribution of the estimator shows some approach to normality. The statements are as follows:

(1) The frequency with which the assertion estimate - (1.96) s.d. (estimator) < (true value) < estimate - (1.96) s.d. (estimator, is wrong, is usually slightly higher than 5% .

(2) The frequency with which (true value) > estimate + 1.96 s.d. (estimator) is greater than 2.5% .

(3) The frequency with which (true value) < estimate - 1.96 s.d. (estimator) is less than 2.5% .

Here s.d. stands for standard deviation." [Cochran]

If the population is negatively skewed statement (1) will still be true but in (2) the frequency will be less than 2.5% and in (3) the frequency will be greater than 2.5% . Since any standard book on sampling theory gives an excellent discussion on skewness and its effect on confidence statement, we shall not elaborate on this point any further.

We shall show that the distribution $\hat{\sigma}_{ij}$ could be both positively and negatively skewed even though most frequently in practical situations $\hat{\sigma}_{ij}$ is positively skewed. It is worthwhile to emphasize the fact that if we are only interested in the absolute value of the error of an estimate, then a fair amount of skewness in the distribution of the estimator can be tolerated. However, if we want to make a confidence statement, the normal approximation is not trustworthy unless very little skewness remains in the distribution of the estimator.

There is no general rule as to how large the sample size must be for the use of normal approximation. For populations in which the deviation from normality consists mainly of marked positive skewness, a practical rule due to W. G. Cochran is as follows:

$$\text{Sample size} \quad > \cdot \ 25 \ G_1^2 \qquad (10)$$

where

$$G_1 = \frac{\text{(Third Central Moment)}}{\text{(Second Central Moment)}^{3/2}}$$

This rule is designed so that a 95% confidence probability statement will be wrong not more than 6% of the time.

3. Skewness in the distribution of $\hat{\sigma}_{ij}$

Theorem 5: Sampling fractions being small the probability distribution of $\hat{\sigma}_{ij}$ is positively skewed or negatively skewed according to whether

$$\left\{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sigma_{ij} - 3\left(\frac{N_i N_j}{n_i n_j}\right) \sigma_{ij}^2 - 3\sigma_{ij}(\sigma_{ij}-1)^2 + 2\sigma_{ij}^3\right\} \quad \text{is} \quad (11)$$

positive or negative respectively.

Proof:

$$E(\hat{\sigma}_{ij} - \sigma_{ij})^3 = E(\hat{\sigma}_{ij}^3 - 3\hat{\sigma}_{ij}^2 \cdot \sigma_{ij} + 3\hat{\sigma}_{ij} \cdot \sigma_{ij} - \sigma_{ij}^3)$$
$$= E(\hat{\sigma}_{ij}^3) - 3\sigma_{ij} E(\hat{\sigma}_{ij}^2) + 3\sigma_{ij} E(\hat{\sigma}_{ij}) - \sigma_{ij}^3$$

Now,

$$E(\hat{\sigma}_{ij})^3 = \left(\frac{N_i N_j}{n_i n_j}\right)^3 E(n_{ij}^3)$$
$$= \left(\frac{N_i N_j}{n_i n_j}\right)^3 \left\{ \frac{\sum_{\alpha} \delta_{\alpha(i,j)} \binom{N_i - 1}{n_i - 1}\binom{N_j - 1}{n_j - 1}}{\binom{N_i}{n_i}\binom{N_j}{n_j}} \right.$$
$$+ \frac{3.2 \sum_{\alpha,\beta} \delta_{\alpha,\beta(i,j)} \binom{N_i - 2}{n_i - 2}\binom{N_j - 2}{n_j - 2}}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$
$$\left. + \frac{6 \sum_{\alpha,\beta,\gamma} \delta_{\alpha,\beta,\gamma(i,j)} \binom{N_i - 3}{n_i - 3}\binom{N_j - 3}{n_j - 3}}{\binom{N_i}{n_i}\binom{N_j}{n_j}} \right\} \quad (12)$$

where $\delta_{\alpha(i,j)}$ and $\delta_{\alpha,\beta(i,j)}$ are defined as in the proof of Theorem 2. $\delta_{\alpha,\beta,r(i,j)}$ is defined as follows:

$$\delta_{\alpha,\beta,r(i,j)} = \begin{cases} 1 & \text{if units } \alpha, \beta \text{ and } r \text{ appear in both } S_i \text{ and } S_j \\ 0 & \text{otherwise.} \end{cases}$$

$\sum \delta_{\alpha,\beta(i,j)}$ has to be taken twice since the order in which α and β are taken gives rise to different terms. Equation (12) on simplification gives

$$E(\hat{\sigma}_{ij}^{3}) = \left\{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sigma_{ij} + 3\left(\frac{N_i N_j}{n_i n_j}\right)^2 \frac{(n_i-1)(n_j-1)}{(N_i-1)(N_j-1)} \sigma_{ij}(\sigma_{ij}-1) + \left(\frac{N_i N_j}{n_i n_j}\right)^2 \frac{(n_i-1)(n_i-2)(n_j-1)(n_j-2)}{(N_i-1)(N_i-2)(N_j-1)(N_j-2)} \sigma_{ij}(\sigma_{ij}-1)(\sigma_{ij}-2)\right\} \tag{13}$$

We have already evaluated $E(\hat{\sigma}_{ij}^{2})$ and $E(\sigma_{ij})$. Thus

$$E(\hat{\sigma}_{ij}-\sigma_{ij})^3 = \left[\left\{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sigma_{ij} + 3\left(\frac{N_i N_j}{n_i n_j}\right)\frac{(n_i-1)(n_j-1)}{(N_i-1)(N_j-1)} \sigma_{ij}(\sigma_{ij}-1) + \left(\frac{N_i N_j}{n_i n_j}\right)^2 \frac{(n_i-1)(n_i-2)(n_j-1)(n_j-2)}{(N_i-1)(N_i-2)(N_j-1)(N_j-2)} \sigma_{ij}(\sigma_{ij}-1)(\sigma_{ij}-2)\right\} - 3\cdot\sigma_{ij}\left(\frac{N_i N_j}{n_i n_j}\right)^2\left\{\frac{n_i n_j}{N_i N_j}\sigma_{ij} + \frac{n_i(n_i-1)n_j(n_j-1)}{N_i(N_i-1)N_j(N_j-1)}\sigma_{ij}(\sigma_{ij}-1)\right\} + 3\sigma_{ij}^3 - \sigma_{ij}^3\right] \tag{14}$$

when $\frac{n_i}{N_i}$ and $\frac{n_j}{N_j}$ are small, Equation (14) gives

$$E(\hat{\sigma}_{ij} - \sigma_{ij})^3 \cong \left\{ \left(\frac{N_i N_j}{n_i n_j} \right)^2 \sigma_{ij} - 3\left(\frac{N_i N_j}{n_i n_j} \right) \sigma_{ij}^2 - 3\sigma_{ij}(\sigma_{ij} - 1)^2 + 2\sigma_{ij}^3 \right\} \quad (15)$$

From equation (15) when $E(\hat{\sigma}_{ij} - \sigma_{ij})^3 > 0$ the distribution of $\hat{\sigma}_{ij}$ has positive skewness and if $E(\hat{\sigma}_{ij} - \sigma_{ij})^3 < 0$ then distribution of $\hat{\sigma}_{ij}$ has negative skewness. This is equivalent to the assertion in the Theorem.

Note: When $\frac{n_i}{N_i}$ and $\frac{n_j}{N_j}$ are not small equation (14) must be used to determine skewness. In the extreme case when $n_i = N_i$ and $n_j = N_j$, it is easy to verify that Eq. (14) reduces to zero as is to be expected.

Corollary 1: When $\frac{n_i}{N_i}$ and $\frac{n_j}{N_j}$ are small the coefficient of skewness, G_1, of the distribution of $\hat{\sigma}_{ij}$ is given by:

$$G_1 = \frac{\left\{ \left(\frac{N_i N_j}{n_i n_j} \right)^2 \sigma_{ij} - 3\left(\frac{N_i N_j}{n_i n_j} \right) \sigma_{ij}^2 - 3\sigma_{ij}(\sigma_{ij} - 1)^2 + 2\sigma_{ij}^3 \right\}}{\left(\frac{N_i N_j}{n_i n_j} \sigma_{ij} \right)^{3/2}} \quad (16)$$

Corollary 2: When $\frac{n_i}{N_i}$, i = 1, 2, ,r . are small, the coefficient of skewness of the distribution of $\hat{\sigma}_{ij\ldots\ell}$ is given by G_1 -

$$G_1 = \frac{\left(\frac{N_i N_j \cdots N_\ell}{n_i n_j \cdots n_\ell} \right)^2 \sigma_{ij\ldots\ell} - 3\left(\frac{N_i N_j \cdots N_\ell}{n_i n_j \cdots n_\ell} \right) \sigma_{ij\ldots\ell}^2 - 3\sigma_{ij\ldots\ell}(\sigma_{ij\ldots\ell} - 1)^2 + 2\sigma_{ij\ldots\ell}^3 \}}{\left(\frac{N_i N_j \cdots N_\ell}{n_i n_j \cdots n_\ell} \cdot \sigma_{ij\ldots\ell} \right)^{3/2}}$$

4. Effect of Non-Normality on Estimated Variance

We have derived earlier the variance of the estimator $\hat{\sigma}_{ij}$. The expression for the variance is in terms of σ_{ij}, the population value. In

practice, when the variance has to be computed using sample values, it is important to investigate the effect of non-normality of the distribution of $\hat{\sigma}_{ij}$ on the computed variance, say S^2. One such effect is that the estimated variance S^2 may be more highly variable from sample (pair) to sample (pair) than we expect, if we assume that we are sampling from a normal distribution. For any infinite population the variance of estimated variance S^2 in m repeated samples (pairs), is

$$\text{Variance (estimated variance)} = \frac{2\{E(\hat{\sigma}_{ij} - \sigma_{ij})^2\}^2}{m-1} + \frac{\kappa_4}{m} \qquad (17)$$

The first term on the right hand side of Eq. (17) is the value which the variance of S^2 has when the parent distribution is normal. The second term represents the effect of non-normality. The quantity κ_4 is Fisher's fourth cumulant (Fisher, 1932) and is given by

$$\kappa_4 = E'(\hat{\sigma}_{ij} - \sigma_{ij})^4 - 3E(\hat{\sigma}_{ij} - \sigma_{ij})^2 \qquad (18)$$

The skewness in the distribution as measured by G_1 does not affect the stability of S^2. The important factor is the fourth moment in the population. The cumulant κ_4 is zero for a normal distribution. It may take positive or negative values in other distributions but in those encountered in sampling practice κ_4 appears to be positive much more often than negative and may have a high value for some parent distributions.

$$\text{Variance } (S^2) = 2\,\frac{E(\hat{\sigma}_{ij} - \sigma_{ij})^2}{m-1}^2 \left\{1 + \frac{m-1}{2m} G_2\right\} \qquad (19)$$

where $G_2 = \kappa_4 / \{\mathrm{Var}(\hat{\sigma}_{ij})\}^2$ is Fisher's measure of kurtosis $(1 + \frac{m-1}{2m} G_2)$ is the factor by which the variance of S^2 is inflated due to non-normality. Since this factor is almost independent of m, the inflation remains even with large m.

We shall determine G_2 for $\hat{\sigma}_{ij}$. For this we have to determine $E(\hat{\sigma}_{ij} - \sigma_{ij})^4$.

$$E(\hat{\sigma}_{ij} - \sigma_{ij})^4 = E(\hat{\sigma}_{ij}^4) - 4\sigma_{ij} E(\hat{\sigma}_{ij}^3) + 6\sigma_{ij}^2 E(\hat{\sigma}_{ij}^2) - 4\sigma_{ij}^3 E(\hat{\sigma}_{ij}) + \sigma_{ij}^4$$

$E(\hat{\sigma}_{ij}^4)$ can be computed using the δ-functions in the same way as the first, second and third moments. We have to define an additional δ-function as follows:

$$\delta_{\alpha,\beta,r,\epsilon(i,j)} = \begin{cases} 1 & \text{if elements } \alpha, \beta, r, \text{ and } \epsilon \text{ appear in both } S_i \text{ and } S_j. \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$E(\hat{\sigma}_{ij}^4) = \left(\frac{N_i N_j}{n_i n_j}\right)^4 E\{(n_{ij})^4\}$$

$$= \left(\frac{N_i N_j}{n_i n_j}\right)^4 \left\{ \frac{\sum_{\alpha} \delta_{\alpha(i,j)} \binom{N_i - 1}{n_i - 1}\binom{N_j - 1}{n_j - 1}}{\binom{N_i}{n_i}\binom{N_j}{n_j}} \right.$$

$$\left. + \frac{14 \sum_{\alpha,\beta} \delta_{\alpha,\beta(i,j)} \binom{N_i - 2}{n_i - 2}\binom{N_j - 2}{n_j - 2}}{\binom{N_i}{n_i}\binom{N_j}{n_j}} \right.$$

$$+ \frac{36 \sum_{\alpha,\beta,r} \delta_{\alpha,\beta,r(i,j)} \binom{N_i - 3}{n_i - 3}\binom{N_j - 3}{n_j - 3}}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

$$+ \frac{24 \sum_{\alpha,\beta,r,\varepsilon} \delta_{\alpha,\beta,r,\varepsilon(i,j)} \binom{N_i - 4}{n_i - 4}\binom{N_j - 4}{n_j - 4}}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$$

As before $\frac{1}{\binom{N_i}{n_i}\binom{N_j}{n_j}}$ is the probability of choosing two samples of sizes n_i and n_j from U_i and U_j

$\binom{N_i - 1}{n_i - 1}\binom{N_j - 1}{n_j - 1}$ = number of pairs of samples in which a particular unit, say α, appears.

$\binom{N_i - 2}{n_i - 2}\binom{N_j - 2}{n_j - 2}$ = number of pairs of samples in which two particular units, say α and β, appear.

$\binom{N_i - 3}{n_i - 3}\binom{N_j - 3}{n_j - 3}$ = number of pairs of samples in which three particular units, say α, β, and r, appear.

$\binom{N_i - 4}{n_i - 4}\binom{N_j - 4}{n_j - 4}$ = number of pairs of samples in which four particular units, say α, β, r, and δ, appear.

The coefficients 14, 36 and 24 are obtained as follows: Since we are taking n_{ij}^4, we have to obtain the multinomial coefficients for 2 units, 3 units, and 4 units. For 2 units, we take $\frac{4!}{2!\,2!}$ for terms like $\alpha^2\beta^2$. $2\,\frac{4!}{3!\,1!}$ for terms like $\alpha^3\beta$ and $\alpha\beta^3$ giving a total of 14. For 3 units, we have to take $3\,\frac{4!}{2!\,1!\,1!}$ = 36 for terms like $\alpha^2\beta r$, $\beta^2 r\alpha$, $r^2\alpha\beta$

For 4 units we have $\frac{4!}{1!\,1!\,1!\,1!} = 24$.

Thus

$$E(\sigma_{ij}^{\cdot 4}) = \left(\frac{N_i N_j}{n_i n_i}\right)^4 \left\{ \frac{n_i n_j}{N_i N_j}\,\sigma_{ij} + \frac{14 \cdot n_i(n_i-1)n_j(n_j-1)}{N_i(N_i-1)N_j(N_j-1)} \cdot \frac{\sigma_{ij}(\sigma_{ij}-1)}{2!} \right.$$

$$+ \frac{36 \cdot n_i(n_i-1)(n_i-2)n_j(n_j-1)(n_j-2)}{N_i(N_i-1)(N_i-2)N_j(N_j-1)(N_j-2)} \cdot \frac{\sigma_{ij}(\sigma_{ij}-1)(\sigma_{ij}-2)}{3!}$$

$$\left. + \frac{24 \cdot n_i(n_i-1)(n_i-2)(n_i-3)n_j(n_j-1)(n_j-2)(n_j-3)}{N_i(N_i-1)(N_i-2)(N_i-3)N_j(N_j-1)(N_j-2)(N_j-3)} \cdot \frac{\sigma_{ij}(\sigma_{ij}-1)(\sigma_{ij}-2)(\sigma_{ij}-3)}{4!} \right\}$$

$$= \left(\frac{N_i N_j}{n_i n_j}\right)^3 \sigma_{ij} + 7\left(\frac{N_i N_j}{n_i n_j}\right)^3 \frac{(n_i-1)(n_j-1)}{(N_i-1)(N_j-1)}\,\sigma_{ij}(\sigma_{ij}-1)$$

$$+ 6\left(\frac{N_i N_j}{n_i n_j}\right)^3 \frac{(n_i-1)(n_i-2)(n_j-1)(n_j-2)}{(N_i-1)(N_i-2)(N_j-1)(N_j-2)}\,\sigma_{ij}(\sigma_{ij}-1)(\sigma_{ij}-2)$$

$$+ \left(\frac{N_i N_j}{n_i n_j}\right)^3 \frac{(n_i-1)(n_i-2)(n_i-3)(n_j-1)(n_j-2)(n_j-3)}{(N_i-1)(N_i-2)(N_i-3)(N_j-1)(N_j-2)(N_j-3)}\,\sigma_{ij}(\sigma_{ij}-1)(\sigma_{ij}-2)(\sigma_{ij}-3)$$

It is easy to verify that when $n_i = N_i$ and $n_j = N_j$, $E(\hat{\sigma}_{ij}^{\cdot 4}) = \sigma_{ij}^{\cdot 4}$. Now using the values of $E(\hat{\sigma}_{ij}^{\cdot 3})$, $E(\hat{\sigma}_{ij}^{\cdot 2})$ and $E(\hat{\sigma}_{ij}^{\cdot})$ computed earlier, we have

$$E(\hat{\sigma}_{ij}^{\cdot} - \sigma_{ij})^4 = \left(\frac{N_i N_j}{n_i n_j}\right)^3 \sigma_{ij} + 7\left(\frac{N_i N_j}{n_i n_j}\right)^3 \frac{(n_i-1)(n_j-1)}{(N_i-1)(N_j-1)}\,\sigma_{ij}(\sigma_{ij}-1)$$

$$+6\left(\frac{N_i N_j}{n_i n_j}\right)^3 \frac{(n_i-1)(n_i-2)(n_j-1)(n_j-2)}{(N_i-1)(N_j-2)(N_j-1)(N_j-2)} \quad \sigma_{ij}(\sigma_{ij}-1)(\sigma_{ij}-2)$$

$$+\left(\frac{N_i N_j}{n_i n_j}\right)^3 \frac{(n_i-1)(n_i-2)(n_i-3)(n_j-1)(n_j-2)(n_j-3)}{(N_i-1)(N_i-2)(N_i-3)(N_j-1)(N_j-2)(N_j-3)} \quad \sigma_{ij}(\sigma_{ij}-1)(\sigma_{ij}-2)(\sigma_{ij}-3)$$

$$-4\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sigma_{ij}^2 - 12\left(\frac{N_i N_j}{n_i n_j}\right)^2 \frac{(n_i-1)(n_j-1)}{(N_i-1)(N_j-1)} \quad \sigma_{ij}^2(\sigma_{ij}-1)$$

$$-4\left(\frac{N_i N_j}{n_i n_j}\right)^2 \frac{(n_i-1)(n_i-2)(n_j-1)(n_j-2)}{(N_i-1)(N_i-2)(N_j-1)(N_j-2)} \quad \sigma_{ij}^2(\sigma_{ij}-1)(\sigma_{ij}-2)$$

$$+6\left(\frac{N_i N_j}{n_i n_j}\right)\sigma_{ij}^3 + 6\frac{(n_i-1)(n_j-1)}{(N_i-1)(N_j-1)} \quad \sigma_{ij}^3(\sigma_{ij}-1) - 3\sigma_{ij}^4$$

Notice that when $n_i = N_i$ and $n_j = N_j$, $E(\hat{\sigma}_{ij} - \sigma_{ij})^4 = 0$ as is to be expected.

The extension of the above proof to $E(\hat{\sigma}_{ij\ldots l} - \sigma_{ij\ldots l})^4$ is obvious.

Theorem 6: When $\frac{n_i}{N_i}$ and $\frac{n_j}{N_j}$ are small $E(\hat{\sigma}_{ij} - \sigma_{ij})^4 \simeq \left(\frac{N_i N_j}{n_i n_j}\right)^3 \sigma_{ij}$.
In general when $\frac{n_i}{N_i}, \frac{n_j}{N_j}, \ldots, \frac{n}{N_l}$ are small $E(\hat{\sigma}_{ij\ldots l} - \sigma_{ij\ldots l})^4 \simeq \left(\frac{N_i N_j \ldots N_l}{n_i n_j \ldots n_l}\right)^3 \sigma_{ij\ldots l}$
The proof of this theorem follows from (20).

Theorem 7: When $\frac{n_i}{N_i}$ and $\frac{n_j}{N_j}$ are small, the coefficient of kurtosis G_2 of the distribution of $\hat{\sigma}_{ij} \simeq \left(\frac{N_i N_j}{n_i n_j}\right)\frac{1}{\sigma_{ij}} - 3$ and in general the coefficient of kurtosis G_2 of the distribution of $\hat{\sigma}_{ij\ldots l} \simeq \frac{N_i N_j \ldots N_l}{n_i n_j \ldots n_l}\frac{1}{\sigma_{ij\ldots l}} - 3$.

Proof:

$$G_2 = \frac{E(\hat{\sigma}_{ij} - \sigma_{ij})^4 - 3\left\{E(\hat{\sigma}_{ij} - \sigma_{ij})^2\right\}^2}{\left\{E(\hat{\sigma}_{ij} - \sigma_{ij})^2\right\}^2}$$ by definition.

Thus

$$G_2 \simeq \frac{\left(\frac{N_i N_j}{n_i n_j}\right)^3 \sigma_{ij}}{\left(\frac{N_i N_j}{n_i n_j}\right)^2 \sigma_{ij}^2} - 3 = \frac{N_i N_j}{n_i n_j} \frac{1}{\sigma_{ij}} - 3$$

$$G_2 = \frac{E(\hat{\sigma}_{ij\ldots\ell} - \sigma_{ij\ldots\ell})^4}{\left\{E(\hat{\sigma}_{ij\ldots\ell} - \sigma_{ij\ldots\ell})^2\right\}^2} - 3 \simeq \frac{N_i N_j \ldots N_\ell}{n_i n_j \ldots n_\ell} \frac{1}{\sigma_{ij\ldots\ell}} - 3$$

5. Observations about the skewness and kurtosis of the distribution of $\hat{\sigma}_{ij}$

When sampling fractions are small the coefficient of skewness of G_1 of the distribution of $\hat{\sigma}_{ij}$ is approximately $\sqrt{\left(\frac{N_i N_j}{n_i n_j} \frac{1}{\sigma_{ij}}\right)}$. This indicates that for small values of σ_{ij}, the skewness is much pronounced unless sampling fractions are made large, in which case the exact formula for G_1 has to be used to compute the skewness. Except for documents that have nothing in common, $\frac{N_i N_j}{n_i n_j}$ is an upperbound for skewness when sampling fractions are small. The coefficient of excess or kurtosis G_2 for small sampling fractions varies as $\frac{N_i N_j}{n_i n_j} \frac{1}{\sigma_{ij}} - 3$. Here again when σ_{ij} is very small G_2 is large and for all pairs of documents which have at least one unit in common; $\frac{N_i N_j}{n_i n_j} - 3$ is an upper bound for G_2.

The probability distribution of n_{ij}

In order to show by computation that the unbiased estimator $\frac{N_i N_j}{n_i n_j} n_{ij}$ is indeed the maximum likelihood estimator, we have to evaluate the probability for n_{ij}. In order that the two samples of sizes n_i and n_j should have n_{ij} = a co-occurrences, these items should be included in the samples. Let N be the number of items that co-occur in the populations U_i and U_j, then any $\binom{N}{a}$ could be included in the two samples and the remaining n_i -a and n_j -a items of the two populations, respectively, so that none of the common items appear among both $(n_i - a)$ and $(n_j - a)$ items of the two samples.

Thus, the probability that n_{ij} = a can be easily seen to be

$$P(n_{ij} = a) = \frac{\sum_{y=0}^{N-a} \binom{N}{a} \binom{N-a}{y} \binom{N_i - N}{n_i - a - y} \binom{N_j - a - y}{n_j - a}}{\binom{N_i}{n_i} \binom{N_j}{n_j}}$$

An alternative method of derivation of $P(n_{ij} = a)$ will be briefly discussed below. The derivation is based on a theorem concerning the realization of m among N events. For a more detailed reading Chapter IV of W. Feller's "An Introduction to Probability Theory and its Applications" forms an excellent reference. If $A_1, A_2, \ldots, A_N$ are any N events, not necessarily mutually exclusive, then $P_{(a)}$ the

probability that exactly a among the N events occur can be expressed in terms of S_a, S_{a+1} S_N where $S_a =$ $P(A_1, \ldots, A_a)$ where $P(A_1, \ldots, A_a)$ is the probability of joint occurence of $A_1, \ldots, A_a$. In essence, the theorem asserts that

$$P_{(a)} = S_a - \binom{a+1}{a} S_{a+1} + \binom{a+2}{a} S_{a+2} + \cdots \pm \binom{N}{a} S_N$$

In the context of our problem, S_a is the probability that any a of the common items will appear in both the samples. This is obviously $\dfrac{\binom{N}{a}\binom{n_i}{a}\binom{n_j}{a}}{\binom{N_i}{a}\binom{N_j}{a}}$.

Thus:

$$P\left\{N_{ij} = a\right\} = P_{(a)} = \frac{\binom{N}{a}\binom{n_i}{a}\binom{n_j}{a}}{\binom{N_i}{a}\binom{N_j}{a}} - \frac{\binom{a+1}{a}\binom{N}{a+1}\binom{n_i}{a+1}\binom{n_j}{a+1}}{\binom{N_i}{a+1}\binom{N_j}{a+1}}$$

$$+ \frac{\binom{a+2}{a}\binom{N}{a+2}\binom{n_i}{a+2}\binom{n_j}{a+2}}{\binom{N_i}{a+2}\binom{N_j}{a+2}} + \pm \frac{\binom{N}{a}\binom{N}{N}\binom{n_i}{N}\binom{n_j}{N}}{\binom{N_i}{N}\binom{N_j}{N}}$$

$$= \sum_{x=a}^{N} \left\{ (-1)^{x-a} \binom{N}{x}\binom{x}{a}\binom{n_i}{x}\binom{n_j}{x} \Big/ \binom{N_i}{x}\binom{N_j}{x} \right\}$$

The r^{th} moment about the origin

Theorem

$$E\left\{\hat{\sigma}_{ij}^{\,r}\right\} = \left(\frac{N_i N_j}{n_i\, n_j}\right)^r \left\{ D(r;a_1^{(1)}) \frac{\sum_{\alpha} \bar{\delta}_{\alpha(i,j)} \binom{N_i - 1}{n_i - 1}\binom{N_j - }{n_j - }}{\binom{N_i}{n_i}\binom{N_j}{n_j}} \right.$$

$$+ D(r;a_1^{(2)}, a_2^{(2)}) \sum_{\alpha} \frac{\bar{\delta}_{\alpha_1,\alpha_2(i,j)} \binom{N_i - 2}{n_i - 2}\binom{N_j - 2}{n_j - 2}}{\binom{N_i}{n_i}\binom{N_j}{n_j}} + \ldots\ldots$$

$$\left. + D(r;a_1^{(r)}, a_2^{(r)}, a_3^{(r)}, \ldots\ldots, a_r^{(r)}) \sum_{\alpha_1,\ldots\alpha_{r-1},\alpha_r} \bar{\delta}_{\alpha_1,\ldots\ldots,\alpha_{r-1},\alpha_r(i,j)} \frac{\binom{N_i - r}{n_i - r}\binom{N_j - r}{n_j - r}}{\binom{N_i}{n_i}\binom{N_j}{n_j}} \right.$$

Where $\alpha_1, \alpha_2, \ldots, \alpha_r$ are items in the populations.

and where $D(r;a_1^{(s)}, a_2^{(s)} \ldots, a_s^{(s)})$ is the denumerant introducted by Sylvester for partitions. $\left(1 \leq s \leq r\right)$

Thus, $D(r;a_1^{(s)}, a_2^{(s)}, \ldots, a_s^{(s)})$ denotes the number of partitions of r into parts $a_1^{(s)}, a_2^{(s)}, \ldots, a_s^{(s)}$ which is the same as the number of solutions in integers of $a_1^{(s)} x_1 + a_2^{(s)} x_2 + \ldots + a_s^{(s)} x_s = r$.

The generating function for this is $D(t;a_1^{(s)}, \ldots a_s^{(s)})$

$$= \sum_{r} D(r;a_1^{(s)}, \ldots a_s^{(s)})\, t^r$$

$$= \frac{1}{(1 - t^{a_1^{(s)}})\,(1 - t^{a_2^{(s)}}) \ldots (1 - t^{a_s^{(s)}})}$$

Computational verification of the fact that $\frac{N_i}{n_i}\frac{N_j}{n_j} \cdot n_{ij}$ is the maximum likelihood estimate of σ_{ij}.

We are interested in finding that value of σ_{ij} which maximizes the probability that the random variable x has an observed value n_{ij}. In one sample problem such as the one discussed by Feller (4), the calculations on the basis of sampling with replacement in two stages reduce the determination of the maximum likely number of items to some fairly simple inequalities in terms of certain observed values. But in the two sample cases, any similar calculations are very difficult and as such we shall use a computational scheme. Here we shall only include the tabulation of $\phi(x)$ for $N_i = N_j = 100$, $n_i = n_j = 50$, $\sigma_{ij} = 4\ (4)\ 48$ and for $N_i = N_j = 200$ $n_i = n_j = 100$ $\sigma_{ij} = 4\ (4)\ 40$. These tabulations are sufficient to indicate that $\frac{N_i N_j}{n_i n_j} \cdot n_{ij}$ is the maximum likelihood estimate of σ_{ij}. When $\frac{N_i N_j}{n_i n_j}$ is not an integer $\left[\frac{N_i N_j}{n_i n_j} n_{ij}\right]$ should be used as the maximum likelihood estimate of σ_{ij}. The tabulation is self-explanatory.

Tabulation of the probability distribution of co-occurrences when the sampling fractions are each 1/2.

N_1 = # of items in the 1st population
N_2 = # of items in the 2nd population
N = # of items common to or co-occurring in both populations
$\frac{N_1}{2}$, $\frac{N_2}{2}$ are the sizes of samples

X = observed # of co-occurring items in samples
$P_x(N)$ = Probability of obtaining x co-occurrences when there are N common items in the two populations.

For various values of N, the maximum likelihood value x and the corresponding probability are underlined.

N1	N2	N	X	PX(N)
100	100	4	0	0.31214111
100	100	4	1	0.42758369
100	100	4	2	0.21184912
100	100	4	3	0.04497169
100	100	4	4	0.00344940
100	100	8	0	0.09383162
100	100	8	1	0.26685131
100	100	8	2	0.32007057
100	100	8	3	0.21137340
100	100	8	4	0.08401838
100	100	8	5	0.02057208
100	100	8	6	0.00302837
100	100	8	7	0.00024489
100	100	8	8	0.00000832
100	100	12	0	0.02708941
100	100	12	1	0.12015590
100	100	12	2	0.23534560
100	100	12	3	0.26903435
100	100	12	4	0.19980867
100	100	12	5	0.10151327
100	100	12	6	0.03615530
100	100	12	7	0.00909021
100	100	12	8	0.00160017
100	100	12	9	0.00019220
100	100	12	10	0.00001494
100	100	12	11	0.00000067
100	100	12	12	0.00000001
100	100	16	0	0.00748815
100	100	16	1	0.04612894
100	100	16	2	0.12825721
100	100	16	3	0.21354941
100	100	16	4	0.23819625

N_1	N_2	N	x	$P_N(x)$
100	100	16	5	0.18862925
100	100	16	6	0.10964145
100	100	16	7	0.04768691
100	100	16	8	0.01567438
100	100	16	9	0.00390386
100	100	16	10	0.00073375
100	100	16	11	0.00010290
100	100	16	12	0.00001054
100	100	16	13	0.00000076
100	100	16	14	0.00000003
100	100	16	15	0.00000000
100	100	16	16	0.00000000
100	100	20	0	0.00197504
100	100	20	1	0.01587299
100	100	20	2	0.05830571
100	100	20	3	0.13009740
100	100	20	4	0.19766334
100	100	20	5	0.21725863
100	100	20	6	0.17914570
100	100	20	7	0.11341063
100	100	20	8	0.05594671
100	100	20	9	0.02170372
100	100	20	10	0.00665252
100	100	20	11	0.00161269
100	100	20	12	0.00030839
100	100	20	13	0.00004622
100	100	20	14	0.00000537
100	100	20	15	0.00000047
100	100	20	16	0.00000003
100	100	20	17	0.00000000
100	100	20	18	0.00000000
100	100	20	19	0.00000000
100	100	20	20	0.00000000
100	100	24	0	0.00049513
100	100	24	1	0.00499459
100	100	24	2	0.02321275
100	100	24	3	0.06611821
100	100	24	4	0.12951750
100	100	24	5	0.18559521
100	100	24	6	0.20204028
100	100	24	7	0.17128345
100	100	24	8	0.11501052
100	100	24	9	0.06187926
100	100	24	10	0.02688570
100	100	24	11	0.00947900
100	100	24	12	0.00271828
100	100	24	13	0.00063414
100	100	24	14	0.00012010
100	100	24	15	0.00001839
100	100	24	16	0.00000226
100	100	24	17	0.00000022
100	100	24	18	0.00000001
100	100	24	19	0.00000000
100	100	24	20	0.00000000
100	100	24	21	0.00000000
100	100	24	22	0.00000000
100	100	24	23	0.00000000
100	100	24	24	0.00000000
100	100	28	0	0.00011746
100	100	28	1	0.00144937
100	100	28	2	0.00828432

N_1	N_2	N	x	$P_N(x)$
100	100	28	3	0.02919267
100	100	28	4	0.07120921
100	100	28	5	0.12798472
100	100	28	6	0.17614900
100	100	28	7	0.19048589
100	100	28	8	0.16477585
100	100	28	9	0.11548814
100	100	28	10	0.06619021
100	100	28	11	0.03122428
100	100	28	12	0.01217704
100	100	28	13	0.00393627
100	100	28	14	0.00105583
100	100	28	15	0.00023492
100	100	28	16	0.00004328
100	100	28	17	0.00000658
100	100	28	18	0.00000082
100	100	28	19	0.00000008
100	100	28	20	0.00000000
100	100	28	21	0.00000000
100	100	28	22	0.00000000
100	100	28	23	0.00000000
100	100	28	24	0.00000000
100	100	28	25	0.00000000
100	100	28	26	0.00000000
100	100	28	27	0.00000000
100	100	28	28	0.00000000
100	100	32	0	0.00002623
100	100	32	1	0.00038896
100	100	32	2	0.00268119
100	100	32	3	0.01144083
100	100	32	4	0.03394464
100	100	32	5	0.07457060
100	100	32	6	0.12612309
100	100	32	7	0.16859476
100	100	32	8	0.18145308
100	100	32	9	0.15937337
100	100	32	10	0.11538137
100	100	32	11	0.06936508
100	100	32	12	0.03481675
100	100	32	13	0.01464684
100	100	32	14	0.00517721
100	100	32	15	0.00153968
100	100	32	16	0.00038537
100	100	32	17	0.00008111
100	100	32	18	0.00001433
100	100	32	19	0.00000212
100	100	32	20	0.00000026
100	100	32	21	0.00000002
100	100	32	22	0.00000000
100	100	32	23	0.00000000
100	100	32	24	0.00000000
100	100	32	25	0.00000000
100	100	32	26	0.00000000
100	100	32	27	0.00000000
100	100	32	28	0.00000000
100	100	32	29	0.00000000
100	100	32	30	0.00000000
100	100	32	31	0.00000000
100	100	32	32	0.00000000
100	100	36	0	0.00000548
100	100	36	1	0.00009647

N_1	N_2	N	X	$P_N(X)$
100	100	36	2	0.00079096
100	100	36	3	0.00402561
100	100	36	4	0.01429017
100	100	36	5	0.03768742
100	100	36	6	0.07680590
100	100	36	7	0.12421554
100	100	36	8	0.16246317
100	100	36	9	0.17425179
100	100	36	10	0.15487679
100	100	36	11	0.11498225
100	100	36	12	0.07173430
100	100	36	13	0.03777774
100	100	36	14	0.01684935
100	100	36	15	0.00637883
100	100	36	16	0.00205251
100	100	36	17	0.00056161
100	100	36	18	0.00013064
100	100	36	19	0.00002581
100	100	36	20	0.00000432
100	100	36	21	0.00000061
100	100	36	22	0.00000007
100	100	36	23	0.00000000
100	100	36	24	0.00000000
100	100	36	25	0.00000000
100	100	36	26	0.00000000
100	100	36	27	0.00000000
100	100	36	28	0.00000000
100	100	36	29	0.00000000
100	100	36	30	0.00000000
100	100	36	31	0.00000000
100	100	36	32	0.00000000
100	100	36	33	0.00000000
100	100	36	34	0.00000000
100	100	36	35	0.00000000
100	100	36	36	0.00000000
100	100	40	0	0.00000107
100	100	40	1	0.00002205
100	100	40	2	0.00021289
100	100	40	3	0.00127856
100	100	40	4	0.00536684
100	100	40	5	0.01677578
100	100	40	6	0.04062538
100	100	40	7	0.07829129
100	100	40	8	0.12239429
100	100	40	9	0.15743871
100	100	40	10	0.16843974
100	100	40	11	0.15113331
100	100	40	12	0.11445647
100	100	40	13	0.07352436
100	100	40	14	0.04021275
100	100	40	15	0.01877760
100	100	40	16	0.00750062
100	100	40	17	0.00256594
100	100	40	18	0.00075217
100	100	40	19	0.00018890
100	100	40	20	0.00004061
100	100	40	21	0.00000747
100	100	40	22	0.00000117
100	100	40	23	0.00000016
100	100	40	24	0.00000001
100	100	40	25	0.00000000

N_1	N_2	N	X	$P_N(X)$
100	100	40	26	0.00000000
100	100	40	27	0.00000000
100	100	40	28	0.00000000
100	100	40	29	0.00000000
100	100	40	30	0.00000000
100	100	40	31	0.00000000
100	100	40	32	0.00000000
100	100	40	33	0.00000000
100	100	40	34	0.00000000
100	100	40	35	0.00000000
100	100	40	36	0.00000000
100	100	40	37	0.00000000
100	100	40	38	0.00000000
100	100	40	39	0.00000000
100	100	40	40	0.00000000
100	100	44	0	0.00000019
100	100	44	1	0.00000462
100	100	44	2	0.00005218
100	100	44	3	0.00036699
100	100	44	4	0.00180650
100	100	44	5	0.00663222
100	100	44	6	0.01889621
100	100	44	7	0.04292551
100	100	44	8	0.07926784
100	100	44	9	0.12072065
100	100	44	10	0.15330181
100	100	44	11	0.16371922
100	100	44	12	0.14802530
100	100	44	13	0.11390086
100	100	44	14	0.07489260
100	100	44	15	0.04221094
100	100	44	16	0.02044027
100	100	44	17	0.00851767
100	100	44	18	0.00305741
100	100	44	19	0.00094572
100	100	44	20	0.00026205
100	100	44	21	0.00005785
100	100	44	22	0.00001141
100	100	44	23	0.00000193
100	100	44	24	0.00000028
100	100	44	25	0.00000003
100	100	44	26	0.00000000
100	100	44	27	0.00000000
100	100	44	28	0.00000000
100	100	44	29	0.00000000
100	100	44	30	0.00000000
100	100	44	31	0.00000000
100	100	44	32	0.00000000
100	100	44	33	0.00000000
100	100	44	34	0.00000000
100	100	44	35	0.00000000
100	100	44	36	0.00000000
100	100	44	37	0.00000000
100	100	44	38	0.00000000
100	100	44	39	0.00000000
100	100	44	40	0.00000000
100	100	44	41	0.00000000
100	100	44	42	0.00000000
100	100	44	43	0.
100	100	44	44	0.
100	100	48	0	0.00000003

N_1	N_2	N	X	$P_N(X)$
100	100	48	1	0.00000088
100	100	48	2	0.00001159
100	100	48	3	0.00009500
100	100	48	4	0.00054537
100	100	48	5	0.00233697
100	100	48	6	0.00777967
100	100	48	7	0.02067328
100	100	48	8	0.04471797
100	100	48	9	0.07989366
100	100	48	10	0.11922121
100	100	48	11	0.14989592
100	100	48	12	0.15988345
100	100	48	13	0.14546409
100	100	48	14	0.11337315
100	100	48	15	0.07594950
100	100	48	16	0.04384447
100	100	48	17	0.02185249
100	100	48	18	0.00941551
100	100	48	19	0.00350972
100	100	48	20	0.00113213
100	100	48	21	0.00031595
100	100	48	22	0.00007623
100	100	48	23	0.00001588
100	100	48	24	0.00000285
100	100	48	25	0.00000044
100	100	48	26	0.00000005
100	100	48	27	0.00000000
100	100	48	28	0.00000000
100	100	48	29	0.00000000
100	100	48	30	0.00000000
100	100	48	31	0.00000000
100	100	48	32	0.00000000
100	100	48	33	0.00000000
100	100	48	34	0.00000000
100	100	48	35	0.00000000
100	100	48	36	0.00000000
100	100	48	37	0.00000000
100	100	48	38	0.00000000
100	100	48	39	0.00000000
100	100	48	40	0.00000000
100	100	48	41	0.00000000
100	100	48	42	0.00000000
100	100	48	43	0.00000000
100	100	48	44	0.00000000
100	100	48	45	0.
100	100	48	46	0.
100	100	48	47	0.
100	100	48	48	0.
200	200	4	0	0.31428289
200	200	4	1	0.42470451
200	200	4	2	0.21139818
200	200	4	3	0.04592994
200	200	4	4	0.00367481
200	200	8	0	0.09698421
200	200	8	1	0.26693611
200	200	8	2	0.31568574
200	200	8	3	0.20949508
200	200	8	4	0.08531594
200	200	8	5	0.02183061
200	200	8	6	0.00342709
200	200	8	7	0.00030173
200	200	8	8	0.00001140
200	200	12	0	0.02936734
200	200	12	1	0.12352014
200	200	12	2	0.23383006
200	200	12	3	0.26341031
200	200	12	4	0.19663910
200	200	12	5	0.10246848
200	200	12	6	0.03821433
200	200	12	7	0.01027544
200	200	12	8	0.00197683
200	200	12	9	0.00026533
200	200	12	10	0.00002357
200	200	12	11	0.00000124
200	200	12	12	0.00000002
200	200	16	0	0.00871996
200	200	16	1	0.04983979
200	200	16	2	0.13110764
200	200	16	3	0.21067868
200	200	16	4	0.23143839
200	200	16	5	0.18427483
200	200	16	6	0.10999131
200	200	16	7	0.05019797
200	200	16	8	0.01770021
200	200	16	9	0.00483747
200	200	16	10	0.00102117
200	200	16	11	0.00016472
200	200	16	12	0.00001990
200	200	16	13	0.00000174
200	200	16	14	0.00000010
200	200	16	15	0.00000000
200	200	16	16	0.00000000
200	200	20	0	0.00253714
200	200	20	1	0.01848187
200	200	20	2	0.06278110
200	200	20	3	0.13221341
200	200	20	4	0.19357269
200	200	20	5	0.20941277
200	200	20	6	0.17367016
200	200	20	7	0.11304544
200	200	20	8	0.05864894
200	200	20	9	0.02448764
200	200	20	10	0.00827218
200	200	20	11	0.00226451
200	200	20	12	0.00050139
200	200	20	13	0.00008929
200	200	20	14	0.00001266
200	200	20	15	0.00000140
200	200	20	16	0.00000012
200	200	20	17	0.00000001
200	200	20	18	0.00000000
200	200	20	19	0.00000000
200	200	20	20	0.00000000
200	200	24	0	0.00072282
200	200	24	1	0.00644512
200	200	24	2	0.02702987
200	200	24	3	0.07094684
200	200	24	4	0.13082855
200	200	24	5	0.18035992
200	200	24	6	0.19312188
200	200	24	7	0.16472116
200	200	24	8	0.11387993

N_1	N_2	N	X	$P_N(X)$
200	200	24	9	0.06459489
200	200	24	10	0.03031338
200	200	24	11	0.01183376
200	200	24	12	0.00385459
200	200	24	13	0.00104852
200	200	24	14	0.00023790
200	200	24	15	0.00004487
200	200	24	16	0.00000700
200	200	24	17	0.00000089
200	200	24	18	0.00000009
200	200	24	19	0.00000001
200	200	24	20	0.00000000
200	200	24	21	0.00000000
200	200	24	22	0.00000000
200	200	24	23	0.00000000
200	200	24	24	0.00000000
200	200	28	0	0.00020147
200	200	28	1	0.00213890
200	200	28	2	0.01074458
200	200	28	3	0.03400273
200	200	28	4	0.07614194
200	200	28	5	0.12846885
200	200	28	6	0.16980676
200	200	28	7	0.18048932
200	200	28	8	0.15713903
200	200	28	9	0.11356562
200	200	28	10	0.06879941
200	200	28	11	0.03518806
200	200	28	12	0.01527118
200	200	28	13	0.00564244
200	200	28	14	0.00177823
200	200	28	15	0.00047825
200	200	28	16	0.00010967
200	200	28	17	0.00002139
200	200	28	18	0.00000353
200	200	28	19	0.00000049
200	200	28	20	0.00000005
200	200	28	21	0.00000000
200	200	28	22	0.00000000
200	200	28	23	0.00000000
200	200	28	24	0.00000000
200	200	28	25	0.00000000
200	200	28	26	0.00000000
200	200	28	27	0.00000000
200	200	28	28	0.00000000
200	200	32	0	0.00005490
200	200	32	1	0.00068006
200	200	32	2	0.00400404
200	200	32	3	0.01492303
200	200	32	4	0.03955879
200	200	32	5	0.07945380
200	200	32	6	0.12577412
200	200	32	7	0.16115972
200	200	32	8	0.17035773
200	200	32	9	0.15065817
200	200	32	10	0.11264884
200	200	32	11	0.07178682
200	200	32	12	0.03922705
200	200	32	13	0.01846374
200	200	32	14	0.00751053
200	200	32	15	0.00264594
200	200	32	16	0.00080828
200	200	32	17	0.00021416
200	200	32	18	0.00004918
200	200	32	19	0.00000978
200	200	32	20	0.00000168
200	200	32	21	0.00000025
200	200	32	22	0.00000003
200	200	32	23	0.00000000
200	200	32	24	0.00000000
200	200	32	25	0.00000000
200	200	32	26	0.00000000
200	200	32	27	0.00000000
200	200	32	28	0.00000000
200	200	32	29	0.00000000
200	200	32	30	0.00000000
200	200	32	31	0.00000000
200	200	32	32	0.00000000
200	200	36	0	0.00001461
200	200	36	1	0.00020798
200	200	36	2	0.00141210
200	200	36	3	0.00609086
200	200	36	4	0.01875889
200	200	36	5	0.04395681
200	200	36	6	0.08154303
200	200	36	7	0.12303170
200	200	36	8	0.15393197
200	200	36	9	0.16202658
200	200	36	10	0.14506702
200	200	36	11	0.11142509
200	200	36	12	0.07391334
200	200	36	13	0.04256478
200	200	36	14	0.02136594
200	200	36	15	0.00937696
200	200	36	16	0.00360607
200	200	36	17	0.00121696
200	200	36	18	0.00036068
200	200	36	19	0.00009389
200	200	36	20	0.00002146
200	200	36	21	0.00000430
200	200	36	22	0.00000075
200	200	36	23	0.00000011
200	200	36	24	0.00000001
200	200	36	25	0.00000000
200	200	36	26	0.00000000
200	200	36	27	0.00000000
200	200	36	28	0.00000000
200	200	36	29	0.00000000
200	200	36	30	0.00000000
200	200	36	31	0.00000000
200	200	36	32	0.00000000
200	200	36	33	0.00000000
200	200	36	34	0.00000000
200	200	36	35	0.00000000
200	200	36	36	0.00000000
200	200	40	0	0.00000379
200	200	40	1	0.00006133
200	200	40	2	0.00047415
200	200	40	3	0.00233538
200	200	40	4	0.00823795
200	200	40	5	0.02218003
200	200	40	6	0.04743809

N_1	N_2	N	X	$P_N(X)$
200	200	40	7	0.08282109
200	200	40	8	0.12037122
200	200	40	9	0.14779348
200	200	40	10	0.15504146
200	200	40	11	0.14020139
200	200	40	12	0.11005825
200	200	40	13	0.07542231
200	200	40	14	0.04532545
200	200	40	15	0.02397171
200	200	40	16	0.01118898
200	200	40	17	0.00461896
200	200	40	18	0.00168900
200	200	40	19	0.00054763
200	200	40	20	0.00015752
200	200	40	21	0.00004020
200	200	40	22	0.00000909
200	200	40	23	0.00000182
200	200	40	24	0.00000032
200	200	40	25	0.00000004
200	200	40	26	0.00000000
200	200	40	27	0.00000000
200	200	40	28	0.00000000
200	200	40	29	0.00000000
200	200	40	30	0.00000000
200	200	40	31	0.00000000
200	200	40	32	0.00000000
200	200	40	33	0.00000000
200	200	40	34	0.00000000
200	200	40	35	0.00000000
200	200	40	36	0.00000000
200	200	40	37	0.00000000
200	200	40	38	0.00000000
200	200	40	39	0.00000000
200	200	40	40	0.00000000

Limiting Probability Distribution

Case I. Let $\frac{n_i}{N_i} \longrightarrow c_i$ and $\frac{n_j}{N_j} \longrightarrow c_j$ as $n_i,\ n_j,\ N_i,\ N_j \longrightarrow \infty$

Let N be the number of common items in the two populations and let $n_{ij} = a$.

$$\text{Then} \quad P\{n_{ij} = a\} = \sum_{x=a}^{N} (-1)^{x-a} \frac{\binom{N}{x}\binom{x}{a}\binom{n_i}{x}\binom{n_j}{x}}{\binom{N_i}{x}\binom{N_j}{x}}$$

$$= \sum_{x=a}^{N} (-1)^{x-a} \frac{\binom{N}{x}\binom{x}{a}\, n_i^x \left(1-\frac{1}{n_i}\right)\cdots\left(1-\frac{x-1}{n_i}\right) n_j^x \left(1-\frac{1}{n_j}\right)\cdots\left(1-\frac{x-1}{n_j}\right)}{N_i^x \left(1-\frac{1}{N_i}\right)\cdots\left(1-\frac{x-1}{N_i}\right) N_j^x \left(1-\frac{1}{N_j}\right)\cdots\left(1-\frac{x-1}{N_j}\right)}$$

$$= \sum_{x=a}^{N} (-1)^{x-a} \frac{\binom{N}{x}\binom{x}{a}\left(\frac{n_i n_j}{N_i N_j}\right)\left(1-\frac{1}{n_i}\right)\cdots\left(1-\frac{x_i-1}{n_i}\right)\left(1-\frac{1}{n_j}\right)\cdots\left(1-\frac{x-1}{n_j}\right)}{\left(1-\frac{1}{N_i}\right)\cdots\left(1-\frac{x-1}{N_i}\right)\left(1-\frac{1}{N_j}\right)\cdots\left(1-\frac{x-1}{N_j}\right)}$$

$$\therefore \quad \lim_{\ni} n_i, n_j, N_i, N_j \to \infty \quad \ni \frac{n_i}{N_i} \to c_i,\ \frac{n_j}{N_j} \to c_j \qquad P(n_{ij} = a) = \sum_{x=a}^{N} (-1)^{x-a} \binom{N}{x}\binom{x}{a}(c_i c_j)^x$$

Let $c_i \cdot c_j = p$ (evidently both c_i and $c_j < 1$)

$$\text{Thus} \quad \lim P(n_{ij} = a) = \sum_{x=a}^{N} (-1)^{x-a} \binom{N}{x}\binom{x}{a} p^x$$

$$\binom{N}{a+r}\binom{a+r}{a} = \binom{N}{a}\binom{N-a}{r}$$

Since $\binom{N}{a+r}\binom{a+r}{a} = \dfrac{N!\ (a+r)!}{(a+r)!\,(N-a-r)!\,a!\,r!}$

$$= \frac{N!\ a!\ (N-a)!\ (a+r)!}{a!\,(N-a)!\,(N-a-r)!\,(a+r)!\,a!\,r!}$$

$$= \binom{N}{a}\binom{N-a}{r}$$

Thus $\lim P(n_{ij}=a) = \sum_{x=a}^{N} (-1)^{x-a}\binom{N}{x}\binom{x}{a}p^{x}$

$$= \binom{N}{a}p^{a}(1-p)^{N-a}$$

Case 2 Since case 1 leads to a binomial distribution, we can obtain a Poisson distribution with the added conditions that

$Np = \dfrac{n_i\ n_j}{N_i N_j}$, N tend to λ as $N \to \infty$ and $p = \dfrac{n_i\ n_j}{N_i N_j} \to 0$

$$\text{Thus} \lim_{\substack{N\to\infty \\ Np\to\lambda}} P\{n_{ij}=a\} = \lim_{\substack{N\to\infty \\ Np\to\lambda}} \binom{N}{a} p^a (1-p)^{N-a}$$

$$= \lim_{\substack{N\to\infty \\ Np\to\lambda}} \left\{ \frac{(Np)^a}{a!} \left(1-\tfrac{1}{N}\right)\left(1-\tfrac{2}{N}\right)\cdots\left(1-\tfrac{a-1}{N}\right)\left(1-\tfrac{Np}{N}\right)^{N-a} \right\}$$

$$= \frac{\lambda^a}{a!} e^{-\lambda}$$

Note: The maximum likelihood property of $\frac{N_i N_j}{n_j\, n_j}\, n_{ij}$, the estimate for N is obvious in the two limiting distributions.

STRATIFIED SAMPLING

Assumptions:

(1) As before.

(2) As before.

(3) Each population is stratified into L non-overlapping strata so that $N_i = N_i^{(1)} + N_i^{(2)} + \ldots + N_i^{(L)}$ and $N_i^{(h)}$ represents the number of units in the h^{th} stratum in U_i.

(4) Sample of size n_i drawn from population U_i comes from the L strata so that $n_i = n_i^{(1)} + n_i^{(2)} + \ldots + n_i^{(L)}$ and $n_i^{(h)}$ stands for the portion of the sample from h^{th} stratum in U_i.

A. Main reasons for stratification

"(a) If data of known precision are wanted for certain subdivisions of the population, it is advisable to treat each subdivision as a "population" in its own right.

(b) Administrative convenience may dictate the use of stratification e.g. an agency conducting a survey may have field offices, each of which can supervise the survey for a part of the population.

(c) There may be a marked difference in the sampling problems in different parts of the population.

(d) Stratification may bring about a gain in precision in the estimates of characteristics of the whole population. The basic idea is that it may be possible to divide a heterogeneous population into subpopulations, each of which is internally homogeneous. If each stratum is homogeneous, in that units vary very little from one another, a precise estimate of any stratum value can be obtained from a small sample in that stratum." [Cochran]

B. Sampling scheme: In stratified sampling the population of N_i units $(1 \leq i \leq r)$ is divided into subpopulations $N_i^{(1)}, N_i^{(2)}, \ldots N_i^{(L)}$ so that $N_i = N_i^{(1)} + N_i^{(2)} + \ldots + N_i^{(L)}$. The subpopulations are called strata. To obtain the full benefit from stratification $N_i^{(h)}$ must be known. When the strata have been determined, a sample is drawn from each different strata. The sample sizes within the strata are denoted by $n_i^{(1)}, n_i^{(2)}, \ldots,$ $n_i^{(L)}$ respectively. If a simple random sample is taken in each stratum, the whole procedure is described as stratified random sampling.

Notations: The following symbols all refer to stratum h:

$N_i^{(h)}$	Total number of units of U_i
$n_i^{(h)}$	Total number of units in sample
$n_{ij}^{(h)}$	Total number of units common to samples from U_i and U_j.
$\sigma_{ij}^{(h)}$	Total number of units common to U_i and U_j.
$\hat{\sigma}_{ij}^{(h)}$	Estimator of number of units common to U_i and U_h

---

σ_{ij} = Total number of units common to U_i and U_j.

We shall denote the estimator of σ_{ij} for the populations U_i and U_j by $(\hat{\sigma}_{ij})_{st}$.

Problems

(1) To define estimators of σ_{ij}.

(2) To investigate properties of such estimators.

(3) To determine the best choice of $n_i^{(h)}$ and $n_j^{(h)}$ so as to obtain maximum precision.

(4) To extend, whenever appropriate, the estimation problems to more than two populations.

Analysis

Theorem 8: In any stratum h, $\hat{\sigma}_{ij}^{(h)} = \frac{N_i^{(h)} N_j^{(h)}}{n_i^{(h)} n_j^{(h)}} n_{ij}^{(h)}$ is an unbiased estimator of $\sigma_{ij}^{(h)}$ and in general $\hat{\sigma}_{ij\ldots\ell}^{(h)} = \frac{N_i^{(h)} N_j^{(h)} \ldots N_\ell^{(h)}}{n_i^{(h)} n_j^{(h)} \ldots n_\ell^{(h)}} n_{ij\ldots\ell}^{(h)}$ is an unbiased estimator of $\sigma_{ij\ldots\ell}$

Proof: Since each stratum can be regarded as a subpopulation we need consider $E(\hat{\sigma}_{ij}^{(h)})$ and $E(\hat{\sigma}_{ij\ldots\ell}^{(h)})$ only for that stratum. Then the proof is the same as for Theorem 1.

Theorem 9: If in every stratum h, $\sigma_{ij}'^{(h)}$ is an unbiased estimator of $\sigma_{ij}^{(h)}$ then $(\sigma_{ij}')_{ST}$ defined by

$$(\sigma_{ij}')_{ST} = \sum_{h=1}^{L} \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \sigma_{ij}'^{(h)}$$

is an unbiased estimator of σ_{ij}.

More generally, if $\sigma_{ij\ldots\ell}'^{(h)}$ is an unbiased estimator of $\sigma_{ij\ldots\ell}^{(h)}$, then $(\sigma_{ij}')_{ST}$ defined by

$$(\sigma'_{ij\ldots\ell})_{st} = \sum_{h=1}^{L} \frac{N_i^{(h)} N_j^{(h)} \ldots N_\ell^{(h)}}{N_i N_j \ldots N_\ell} \sigma'^{(h)}_{ij\ldots\ell} \qquad (22)$$

is an unbiased estimator of $\sigma_{ij\ldots\ell}$.

Proof: We want to show that $E\{(\sigma'_{ij})_{st}\} = \sigma_{ij}$

In every stratum h, $\sigma'^{(h)}_{ij}$ is an unbiased estimator of $\sigma_{ij}^{(h)}$ so that

$$E\{(\sigma'_{ij})_{st}\} = E\left\{\sum_{h=1}^{L} \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \sigma'^{(h)}_{ij}\right\} = \sum_{h=1}^{L} \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \sigma_{ij}^{(h)}$$

We have already stated that the strata could be regarded as independent "populations". Thus $\frac{N_i^{(h)} N_j^{(h)}}{N_i N_j}$ is the probability for the stratum h. Thus

$$\sum \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \sigma_{ij}^{(h)} = E_h(\sigma_{ij}^{(h)})$$

where E_h means averaging for strata. This by definition is indeed σ_{ij}. The extension to $(\sigma_{ij\ldots\ell})_{st}$ is obvious.

Note: The proof for Theorem 8 does not use any particular unbiased estimator of each stratum.

Theorem 10: For $\hat{\sigma}_{ij}^{(h)} = \frac{N_i^{(h)} N_j^{(h)}}{n_i^{(h)} n_j^{(h)}} n_{ij}^{(h)}$ and

$$\hat{\sigma}_{ij\ldots\ell}^{(h)} = \frac{N_i^{(h)} N_j^{(h)} \ldots N_\ell^{(h)}}{n_i^{(h)} n_j^{(h)} \ldots n_\ell^{(h)}} n_{ij\ldots\ell}^{(h)} \text{ , } (\hat{\sigma}_{ij})_{st} \text{ and } (\hat{\sigma}_{ij\ldots\ell})_{st}$$

defined by

$$(\hat{\sigma}_{ij})_{st} = \sum_{h=1}^{L} \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \hat{\sigma}_{ij}^{(h)} \quad \text{and} \quad (\hat{\sigma}_{ij\ldots\ell})_{st} = \sum_{h=1}^{L} \frac{N_i^{(h)} N_j^{(h)} \ldots N_\ell^{(h)}}{N_i N_j \ldots N_\ell} \hat{\sigma}_{ij\ldots\ell}^{(h)} \tag{23}$$

are unbiased estimators of σ_{ij}.

Proof:

$$E(\sigma_{ij})_{st} = E\left\{ \sum_{h=1}^{L} \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \cdot \frac{N_i^{(h)} N_j^{(h)}}{n_i^{(h)} n_i^{(h)}} n_{ij}^{(h)} \right\}$$

$$= \sum_{h=1}^{L} \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \left\{ \sum{}' \frac{N_i^{(h)} N_j^{(h)}}{n_i^{(h)} n_j^{(h)}} n_{ij}^{(h)} \right/ \left(\frac{N_i^{(h)}}{n_i^{(h)}} \right) \left(\frac{N_j^{(h)}}{n_j^{(h)}} \right)$$

where $\sum'$ indicates summation in the stratum h. $\frac{N_i^{(h)} N_j^{(h)}}{N_i N_j}$ is the probability that a unit chosen for each of the pair of samples belongs to stratum h. To make sure that the same unit α belongs to both samples in the stratum h, we assign this unit to both samples and choose the remaining $(n_i^{(h)} - 1)$ and $(n_j^{(h)} - 1)$ units from stratum h for the two samples. Thus if we define

$$\delta_{\alpha(i,j)}^{(h)} = \begin{cases} 1 & \text{if unit } \alpha \text{ from stratum h belongs to Samples } S_i \text{ and } S_j. \\ 0 & \text{otherwise.} \end{cases}$$

then

$$E(\hat{\sigma}_{ij})_{st} = \sum_{h=1}^{L} \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \sum_{\alpha}^{(h)} \delta_{\alpha(i,j)} \frac{N_i^{(h)}}{n_i^{(h)}} \frac{N_j^{(h)}}{n_j^{(h)}} \frac{\binom{N_i^{(h)}-1}{n_i^{(h)}-1}\binom{N_j^{(h)}-1}{n_j^{(h)}-1}}{\binom{N_i^{(h)}}{n_i^{(h)}}\binom{N_j^{(h)}}{n_j^{(h)}}}$$

where $\sum_{\alpha}$ is summation over all the units in U_i and U_j

Thus $E(\hat{\sigma}_{ij})_{st} = \sum_{\alpha} \delta_{\alpha(i,j)} = \sigma_{ij}$

where

$$\delta_{\alpha(i,j)} = \begin{cases} 1 & \text{if unit } \alpha \text{ belongs to } U_i \text{ and } U_j \\ 0 & \text{otherwise.} \end{cases}$$

The extension to $(\hat{\sigma}_{ij\ldots l})_{st}$ is obvious.

Note: $(\hat{\sigma}_{ij})_{sr}$ is not the same as $(\hat{\sigma}_{ij})$ where $(\hat{\sigma}_{ij}) = \sum_{h=1}^{L} \hat{\sigma}_{ij}^{(h)}$.
The difference is that in $(\hat{\sigma}_{ij})$ the individual strata receive their correct weight $\frac{N_i^{(h)} N_j^{(h)}}{N_i N_j}$ It is evident that $(\hat{\sigma}_{ij})$ coincides with $(\hat{\sigma}_{ij})_{st}$ provided in each stratum

$$\frac{n_i^{(h)} \cdot n_j^{(h)}}{n_i n_j} = \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} = \text{constant}$$

This means that the sampling fraction is the same in all strata. This stratification is called <u>stratification with proportional allocation of $n_i^{(h)}$ and $n_j^{(h)}$.</u>
It gives a <u>self-weighting sample.</u>

Theorem 11: For stratified sampling, the variance of $\left\{(\hat{\sigma}_{ij})_{st.}\right\}$ as an estimate of σ_{ij} is

$$\text{Variance}\left\{(\hat{\sigma}_{ij})_{st.}\right\} = \sum_{h=1}^{L} \frac{(N_i^{(h)})^2 (N_j^{(h)})^2}{N_i^2 N_j^2} \text{ Variance } (\hat{\sigma}_{ij}^{(h)}) \tag{24}$$

In general,

$$\text{Variance}\left\{(\hat{\sigma}_{ij\ldots\ell})_{st.}\right\} = \sum_{h=1}^{L} \frac{(N_i^{(h)})^2 (N_j^{(h)})^2 \ldots (N_\ell^{(h)})^2}{N_i^2 . N_j^2 \ldots N_\ell^2} \cdot \text{Variance } (\hat{\sigma}_{ij\ldots\ell}^{(h)}) \tag{25}$$

Proof:

$$(\hat{\sigma}_{ij})_{st.} - \sigma_{ij} = \sum_h \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \hat{\sigma}_{ij}^{(h)} - \sum_h \frac{N_i^{(h)} N_j^{(h)}}{N_i N_j} \sigma_{ij}^{(h)} \tag{26}$$

$$= \sum_h \frac{N_i^{(h)} N_j^{(h)} (\hat{\sigma}_{ij}^{(h)} - \sigma_{ij}^{(h)})}{N_i N_j} \ldots$$

$$\left\{(\hat{\sigma}_{ij})_{st.} - \sigma_{ij}\right\}^2 = \sum_h \frac{(N_i^{(h)})^2 (N_j^{(h)})^2 (\hat{\sigma}_{ij}^{(h)} - \sigma_{ij}^{(h)})^2}{N_i^2 N_j^2}$$

$$+ 2 \sum_{h \neq k} \frac{N_i^{(h)} N_j^{(h)} N_i^{(k)} N_j^{(k)} (\hat{\sigma}_{ij}^{(h)} - \sigma_{ij}^{(h)})(\hat{\sigma}_{ij}^{(k)} - \sigma_{ij}^{(k)})}{N_i^2 N_j^2} \tag{27}$$

Since the error in the estimate is now expressed as a weighted mean of the errors of estimation that have been made in each individual strata. The right hand side of Eq. (23) extends over all pairs of strata. We now have to average over all possible samples. For any cross product term, we begin by keeping the sample in stratum h fixed and average over all samples from stratum k. Since sampling is independent in the two strata, the possible samples in stratum k, will be the same and will have the same probabilities, whatever sample has been drawn in stratum h. But since $\hat{\sigma}_{ij}^{(k)}$ is assumed unbiased, the average $\left(\hat{\sigma}_{ij}^{(k)} - \sigma_{ij}^{(k)}\right)$ is zero. Hence all cross product terms vanish. Thus

$$\text{Variance}\left\{\left(\hat{\sigma}_{ij}\right)_{st.}\right\} = \sum_{k=1}^{L} \frac{\left(N_i^{(k)} N_j^{(k)}\right)^2}{N_i^2 N_j^2} E\left(\hat{\sigma}_{ij}^{(k)} - \sigma_{ij}^{(k)}\right)^2$$

$$= \sum_{k=1}^{L} \left(\frac{N_i^{(k)} N_j^{(k)}}{N_i^2 N_j^2}\right)^2 \cdot \text{Variance}\left(\hat{\sigma}_{ij}^{(k)}\right) \qquad (28)$$

The extension to $\left(\hat{\sigma}_{ij\ldots\ell}\right)_{st.}$ is obvious. Note: we have not used any particular unbiased estimator $\hat{\sigma}_{ij}^{(k)}$ in this proof.

Theorem 12: When $\hat{\sigma}_{ij}^{(k)} = \frac{N_i^{(k)} N_j^{(k)}}{N_i N_j} \cdot n_{ij}^{(k)}$ and

$$\left(\hat{\sigma}_{ij}\right)_{st.} = \sum_{k=1}^{L} \frac{N_i^{(k)} N_j^{(k)}}{N_i N_j} \cdot \frac{N_i^{(k)} N_j^{(k)}}{\lambda_i^{(k)} n_j^{(k)}} n_{ij}^{(k)}$$

, the variance $\left(\hat{\sigma}_{ij}\right)_{st.}$ is

$$\text{Variance}\left\{\left(\hat{\sigma}_{ij}\right)_{st.}\right\} = \sum_{k=1}^{L} \left(\frac{N_i^{(k)} N_j^{(k)}}{N_i N_j}\right)^2 \left[\frac{N_i^{(k)} N_j^{(k)}}{n_i^{(k)} n_j^{(k)}} \sigma_{ij}^{(k)} - \frac{N_i^{(k)} N_j^{(k)} (n_i^{(k)} - 1)(n_j^{(k)} - 1)}{(N_i^{(k)} - 1)(N_j^{(k)} - 1) n_i^{(k)} n_j^{(k)}} \sigma_{ij}^{(k)}\right.$$

$$+\left\{\frac{N_i^{(k)} N_j^{(k)} (n_i^{(k)}-1)(n_j^{(k)}-1)}{(N_i^{(k)}-1)(N_j^{(k)}-1)\, n_i^{(k)} n_j^{(k)}} - 1\right\} \sigma_{ij}^{(k)2} \Bigg],$$

(29)

Proof follows from Theorems 2 and 11.

References

1. W. G. Cochran, "Sampling Techniques", John Wiley and Sons, Inc., 1953

2. Leo A. Goodman, On the Analysis of Samples from k Lists, Annals of Mathematical Statistics, 1952, pp. 632-634

3. John Riordan, "An Introduction to Combinatorial Analysis", John Wiley and Sons, Inc., 1958

4. W. Feller, "An Introduction to Probability Theory and its Applications", John Wiley and Sons, Inc., 1957, pp. 43-45 and pp. 88-103

Appendix D

An Experimental Investigation of "Clustering"

INTRODUCTION

In a previous report under this contract (Co-Author Clusters: AF Contract 19(626)-10 Quarterly Report III) a technique for "clustering" on the basis of co-authorship relationships was explored in which networks, or clusters, of authors were found and authors ranked according to a measure of centrality within each net. This report describes:

A. An experimental investigation of citation indexing which was recently undertaken as a continuation and extension of the above work on clustering techniques.

B. An algorithm for automating the path-tracing technique used to discover central authors within co-author networks.

A. CITATION INDEX CLUSTERING

A citation index is an information retrieval tool which lists, for each document in the index, those documents that have cited it--its "descendants". Several large scale citation indices are now being made (1) (2) (3). In this study, a variation of the citation index concept, an author citation index is being explored, which lists, for each author, those authors who cite him.

Data Collected

A small citation index based on a bibliography collected by Charles Bourne of Stanford Research Institute, plus all documents referred to in these articles (in the field of Information Retrieval)

has been made. The index consists of approximately 500 original titles, 1300 cited titles, 213 original authors and 450 cited authors.

Data Analysis

1. Preliminary data describing the document and author samples have been obtained (total number of different authors and titles in the source and citation decks, frequency distributions, growth and interconnectedness of the samples.) It was found that 93% of the authors are cited 5 times or less, and 69% are cited only once. Eight authors were cited 15 times or more (M. Bailey, C. Bernier, R. Casey, H. Luhn, C. Mooers, J. Perry, C. Shannon, M. Taube). However, in one sense the matrix is highly interconnected. Of the 213 authors in the source deck, 62% (132) are also found in the citation deck, which gives an indication of the growth of the author index.

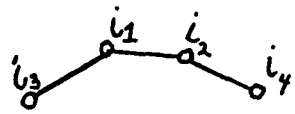
2. An attempt to partition the author citation index in a way analogous to that used in the co-author study (find all authors, b, c, ... who cite a, then all those who cite b, c, ... etc.) is, so far, yielding much larger and more highly interconnected networks than were obtained with co-authors. One network found contains approximately 48 authors. Although the matrix of relations is of course not symmetric, some symmetry was found--7 of the 48 authors cited each other.

3. Examination of a sample citation tracing (on titles) made by John Tukey indicated a "chaining" effect in the citation habits of the authors. (a cites b, then c cites a and b, d cites a, b, and c etc.). This chaining effect will have to be considered in any attempt to obtain "key" authors by some frequency-based criteria.

B. PATH-TRACING ALGORITHM FOR RANKING CO-AUTHORS

In the paper on co-author clusters, a manual path-tracing technique for finding "central" co-authors in a communication network of co-authors was proposed. Here, an algorithm, suitable for computer use is described to replace the manual technique, and a rough estimate of computer time calculated.

In the original paper a network of co-authors was partitioned into sub-networks. Then a sub-network, e.g.,



was drawn in which the nodes represent authors, and the links, the symmetric relation "co-authored with". From this graph, a matrix was obtained and "central" authors derived as follows:

Consider author i

Let p_{ij} = the minimum path (in terms of number of links) between i and j

Let d_i = $\sum_j p_{ij}$ = cumulative "distance" between i and all other authors in the network.

Then

Let $d_i(\min)$ be a measure of "centrality" for i (The "central" author is the one for whom d_i is minimum)

The following algorithm assumes that the network has been partitioned into subnetworks and describes the path-tracing and summation procedure for finding the "central" authors within each sub-network. To automate, we make use of the following known theorem:

In A^k the entry $a_{ij}^{(k)} = c$ if and only if there are c distinct k-chains from i to j. (where A is a matrix with elements a_{ij}, k is the $k^{\underline{th}}$ power of A, and a k-chain is a chain with k links).

Therefore, we proceed as follows:

Consider the author i:

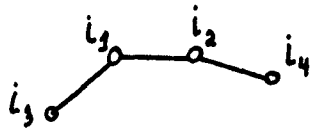
Let a_{ij} = { 1 if i and j are co-authors
0 if not

Let k_{ij} be the smallest value of k such that $a_{ij}^{(k)} > o$.

Then

$$d_i = \sum_j k_{ij}$$

e. g. Considering the first vector $a_{12}\ldots$ look for all non-zero entries. Count the number of non-zero entries. If all entries are non-zero, proceed to the next vector. If some zero entries are found, take the square of the matrix A . Looking only for those entries a_{1j} which were zero in the preceding matrix, count all entries a_{1j} which are now non-zero. Multiply the number of such entries by 2 (the power of the matrix). Add the number obtained to the previous sum. Repeat the procedure by taking successive powers of the matrix until all entries a_{1j} have been accounted for. Then rank the d_i as in the manual technique). For example, given the network:



The manual technique yields the cumulative distances d_1:

$$d_1 = 4$$
$$d_2 = 4$$
$$d_3 = 6$$
$$d_4 = 6$$

To automate, we obtain the symmetric matrix, A

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = A$$

a_{12} and a_{13} are non-zero, therefore, we get the number of such entries times (1) times the power of the matrix, yielding the partial sum $P_1 = (2) \times (1) = 2$

The square of the above matrix, A^2

$$\begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} = A^2$$

Since a_{14} is the only entry in the vector a_{11}.. not previously encountered, we look for it. Here it is non-zero, so, counting the number of such entries, (1), and multiplying by the power of the matrix (2), we get (1) x (2) = 2, added to the partial sum P_1 = the new partial sum $P_2 = 4$.

Since all entries in this vector have been accounted for (a_{ii} is not considered) we need not take further powers of the matrix, and $P_2 = 4 = d_1$,

the cumulative distance obtained with the manual technique. The procedure is then repeated for all other vectors.

For a rough order-of-magnitude estimate of computing time, assume the data is stored with one cell to a computer word. ("Packing" 36 a_{ij} per computer word may be more feasible for larger matrices).

Then if

n = order of the matrix

k = highest power of the matrix needed to find all distances d_i

The number of multiplications, m, required is at most

$$m = n^3 (k-1)$$

(to obtain one element of the new matrix requires n multiplications; there are n x n elements; the process is repeated (k-1) times)

For the largest network found in the co-author experiment

n = 12

k = 4

$m = 3 \cdot 12^3 = 5 \times 10^3$ multiplications

Assuming an average of approximately 10 cycles/multiplication $= 5 \times 10^4$ cycles, at 2μs/cycle =.1 sec. (IBM 7090)

Letting the additions increase the multiplication time by 1/5 (addition takes 2 cycles) gives a total of approximately .12 seconds. And doubling the above figure to allow for other programming instructions gives .24 seconds per network.

Since 23 networks were found in the computer experiment (none as large as above) computing should not be more than 6 seconds.

As a result of work on both co-author and citation patterns, the utility of a general "path tracing" computer program became clear. Such a program would be applicable to citation index (title citation index and/or author c.i.) data, as well as to co-author data, and would have the following functions*:

a. Give the distance between any pair of nodes.

   Distance to be defined either as minimum path as in the co-author study, or as a defined distance--such as, for the citation authors,

   f (m, n, o) where
   - m = number of times a cites b
   - n = number of authors by whom b is cited
   - o = number of authors a cites.

b. Given a distance, find, for any chosen node (person, title) all nodes within that distance.

c. Given a node (e.g., title in a citation index) limit the path tracing by some combination of words within the node (e.g., given a title, select only those titles which contain words a and b but not c, trace only those paths emanating from the selected nodes...)

* In the following description, the "nodes" and "links" would be:

	Author Citation Index	Title Citation Index	Co-Authors
Nodes	authors	titles	authors
Links	"is cited by"	"is cited by"	"co-authored with"

REFERENCES

1. Eugene Garfield, "Citation Indexes for Science" Science, July 15, 1955 Volume 122 pp. 108-111

2. M. M. Kessler, "Technical Information Flow Patterns" Western Joint Computer Conference 1961

3. John Tukey, "A Citation Index for Statistics" (unpublished)

4. Ithiel de Sola Pool and Manfred Kochen, "The Network of Human Contacts" (unpublished)

5. R. Duncan Luce and Albert Perry, "A Method of Matrix Analysis of Group Structure", Psychometrika, Volume 14, No. 1, March, 1949.