AN ALGORITHM FOR ESTIMATING THE MEDIANS OF A WEIGHTED GRAPH SUBJECT TO SIDE CONSTRAINTS, AND AN APPLICATION TO RURAL HOSPITAL LOCATIONS IN BRITISH COLUMBIA

by

ROY ALEXANDER WHITAKER
B.A., University of Exeter, 1962
MSc., Pennsylvania State University, 1965

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

In the Department

of

Geography

We accept this thesis as conforming to
the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

March 1971

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Geography

The University of British Columbia
Vancouver 8, Canada

Date April 27, 1971

ABSTRACT

Plant location as a centralized planning objective in which some agency has control over most of the system elements can be reduced, in many circumstances, to the problem of finding the medians of a weighted graph. This concept is feasible if it can be assumed that each location sought is constrained to a subset of p nodes on an n node network. This combinatorial programming problem can be formally stated as follows: if G is a weighted graph, $w(v_i v_j)$ the weighted distance of node v_i to node v_j, and X_p is any set of p nodes on G $(x_1, x_2, \cdots, x_p)$, then the required set of p nodes X_p^* on G is the p median of the graph if it satisfies the expression

$$\sum_{i=1}^{n} w(v_i \; X_p) \geq \sum_{i=1}^{n} w(v_i \; X_p^*)$$

Although this objective can be explicitly optimized by branch bound algorithms, those developed to date can become computationally infeasible for some large scale problems. A fast method for estimating the medians of a weighted

graph is given which will provide optimal or near optimal solutions on any type of network. The heuristic procedures adopted in this study can be generalized in terms of three basic steps; 1) partition the graph to obtain an initial feasible solution, 2) re-iterate over step 1 to achieve a local minimum, and 3) perturb this convergence to test for a lower bound. The design of steps 1 and 3 are crucial to the success of the algorithmic method. Two procedures are given for the basic partitioning of the graph, one of which is a modification of a criterion originally developed by Singer (1968). The other method introduces a node elimination recursion which appears, experimentally, to be the more efficient procedure for certain types of weighted networks. Efficient perturbation methods are developed for testing the lower bounds obtained.

The basic model structure is modified by the introduction of heuristics for the constrained plant location problem under a wide variety of restrictions. Numerical procedures are suggested for restricting the search to a subset of m potential plant sites among all n nodes on the network. Heuristics are developed for forcing certain locations into solution, for placing upper bound constraints on plant sizes, and for restricting the maximum link distance over which a particular allocation might be made.

Attention is given to the problem of estimating the joint minimization of plant and transportation cost functions over a network surface. For dynamic location-allocation systems an explicit dynamic programming formulation is developed for the optimal sequencing of plant locations over time subject, if necessary, to periodic variations in all cost functions and node weights.

An application of the basic median algorithm to the problem of rural hospital locations in Southeast British Columbia is demonstrated, and computer codes are listed for all the specified models.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure		Page

ACKNOWLEDGMENTS

I should like to extend my gratitude to the many people whose assistance and encouragement made possible the completion of this study. I am indebted to Dr. D. O. Anderson for his help with the epidemiological aspects of the study, to Dr. G. Gates and K. G. Denike for their comments on the logical structure of the algorithms, and to Dr. M. Church and M. J. Patterson for their valuable suggestions which led to improvements in the computer codes. To Mrs. M. Werntz I express my thanks for reducing the computer tapes of hospital separation data to manageable proportions, and to the British Columbia Hospital Insurance Scheme for making these tapes available. Finally I should like to acknowledge with appreciation the Central Mortgage and Housing Corporation for their generous support of this study over its last two years.

CHAPTER I

INTRODUCTION AND STUDY OUTLINE

Scientific and technological advances in medicine have created a need for refinements in the organizations through which health care is given. As medical technology grows ever more complex it has become increasingly apparent that an inherent tendency exists towards ever more specialization and fragmentation in the provision of medical care. This trend, together with mounting social criticism against the continuation of inequalities in the receipt of health care among some underprivileged segments of the population, has set in motion an inflationary spiral whose results are to be seen in medical costs rising more rapidly than the economy can comfortably sustain (McLachlan, 1967). The implications of such a trend are particularly serious if considered against the reality that the needs of medicine must compete with other sectors of the economy for the allocation of resources all too scarce for the number and variety of demands placed upon them.

The rational use of such resources as are allotted to the health care sector has therefore become a desirable end, one whose need has been increasingly recognized over the last few years. Yet given the complexity of the internal

relationships which exist among the various components of the medical sector, and given the multiplicity of exogeneous environmental variables which bear upon it from without, such a goal requires or has required new planning strategies at almost every level of operation.

Objectives of Study

It is the aim of this study to focus upon one subset of these decision making processes, specifically those which are concerned with some facets of the location of hospitals in rural* areas. The construction and maintenance of hospital units represents the single most costly component of the health care system, and accounts for nearly fourty per cent of total expenditures on medical health (Klaman, 1969). At such a price it seems apparent that the community at large can no longer afford the practice of operating a separate hospital in localities where neither local needs nor the supply of qualified personnel can justify the construction of such costly units (Myers, 1961).

The end result of such policies, all too prevalent in the past, is to be seen in the underutilization of facilities and consequent high operating costs whose burden must eventually be borne by the taxpayer at large. In connection with this point McGibony and Block could state as early as 1949 that while some

> *. . . small hospitals are needed . . . it is also obvious that by themselves they are not able to provide a complete service to the patient and*

*non-metropolitan

> *that unless in some way they are tied in with larger hospitals they may provide, not service, but a disservice to their patients* (McGibony and Block, 1949).

At the other end of the scale it is equally clear that the provision of too few facilities, particularly in rural areas, may result in the denial to people of reasonable access to a level of care necessary for the maintenance of their health. The objectives of this study represent an approach to the resolution of this conflict. In more formal terms they may be stated as follows:

1. To identify and assess some of the major components which control the movement of patients in rural areas, through the medical care system to some suitable hospital treatment.

2. To develop an operational model based upon the most significant criteria developed in 1., which can reproduce the system of flows in the existing network with some success, given the constraint of present locations at fixed sizes (measured in terms of some function of the number of beds now available).

3. To devise a hypothetical hospital system by means of such a model that consists of the number, sizes and locations of facilities which can satisfy the particular needs of patient areas subject to the constraints that,

   i. no patient, or maximum number of patients, be more than a maximum predetermined distance (or time) from hospital care and

   ii. that some upper bound in total cost, i.e. a system budget constraint, is not exceeded.

   Comparison of the current and hypothetical location sets will allow the efficiency of the present operating system to be assessed with respect to the particular spatial needs of its dependent patient areas. Imbalances in the present network, if found, can suggest one basis for policy decisions concerning the future

location or relocation of hospital facilities, and for the determination or continuation of referral centres for more specialized types of care.

4. To make provision for the direct incorporation into the model of the present locational pattern at its current scale of operations. Then, given this, some future optimal or near-optimal system of hospital locations (at a particular point in time) may be devised if based upon some forecast of patient demand at discrete points or areas across that demand surface whose bounds are the bounds of the network as a whole. This objective simply recognizes that hospitals once built, and however wisely located they may or may not be, are liable to be maintained for as long as is possible. Locational inertia of this kind is directly attributable to the high initial investment costs of hospital construction and to the widespread pressures within communities to resist the closure of any medical facility once opened, particularly if there are no plans for the replacement of the facility in that locality.

5. To make allowance for the multiple location of facilities over time. If hospitals are to be located in space at successive time intervals then, for some given long term time period, which particular subset of all potential locations is required such that some overall criterion is minimized?

Scope of the Study

This study formulates a numerical allocation model to fulfill the objectives as stated above, and gives a partial application to one regionally defined sub network within the province of British Columbia. Fundamental to the question of hospital costs is the phenomenon of returns to scale which, in this field, are notoriously difficult to assess. A simple correlation of the relationship between the average cost of constructing and maintaining a bed and

the size of the hospital plant will actually suggest the opposite trend. The reality of the situation is more subtle since in practice there exists a hierarchy of services whereby larger hospitals, acting as referral centres, maintain specialized functions in addition to those lower order services which all facilities of a lower rank provide. This central place concept of a nested hierarchical structure of hospital services is fundamental to an understanding of scale economies for, as a rule, the more specialized a function is, the more complex is the equipment required, the more skilled its operators must be, and the less frequently it is in use. Consequently such services are costly to maintain relative to lower order functions. If these effects are held constant however, then, as Berry and others have demonstrated, for a given level of services unit bed costs will tend to decrease with increasing size of hospital (Berry, 1967).

In terms of the application of the model it has not proved possible to derive adequate cost curves since as intimated, this is a major undertaking in its own right. Furthermore, application of the algorithm to some desirable future locational pattern is dependent upon forecasting future demand functions for discrete patient areas and lies beyond the scope of this dissertation. The major component of the study will then, be the development of the model itself, and its testing will be confined to 1) an attempt

to reproduce the flow patterns now extant on the existing network and 2) the derivation of a hypothetical location system, with the same number of plants, which is more efficient in terms of certain key criteria than is the present one. This latter aim should provide some insights into the potential improvements which might be made within the current operating system.

Study Outline

This study consists of three additional chapters whose outline is given below. In Chapter II some contemporary trends in medical health planning are discussed together with some recent analyses of hospital location systems. The rationale for health planning is evaluated; some aspects of the movement of the patient through the medical care system are examined, and some implications of this knowledge in terms of improved medical planning are assessed. The role of the hospital in the system is discussed and a critical review of some previous medical location models is attempted.

Chapter III provides, first of all, a formal statement of the general plant location problem, of which hospital location, at least in rural areas or the inter urban context, may be considered as a particular case. Certain specific constraints which are of significance in the formulation of the general problem are incorporated into this statement. Secondly, a review of previous attempts

to solve the problem is provided. Finally, a detailed description of the algorithm itself is given, and some sample problems on a variety of networks are illustrated to demonstrate the flexibility of the model design for a fairly wide range of potential plant location problems.

Chapter IV partitions British Columbia into sub-regions of a size which can be computationally handled by the algorithm. One of these regions is then selected and analysed in terms of the limital objectives previously specified. The basic algorithm is subjected to some modifications which seem appropriate to the particular case under study, and the fundamental strengths and weaknesses of the formulation are discussed together with some suggestions for potential improvement and development.

CHAPTER II

REGIONAL HEALTH PLANNING: SOME PERSPECTIVES AND PROBLEMS

The Rationale for Regional Health Planning

The last decade and a half in North America has witnessed a surge of interest in the concept of regional planning as the appropriate milieu within which to institute much needed improvements in medical health. Prior to the Hill-Burton Act of 1946, the responsibility for the provision of medical care had essentially been in local community hands (McNerney, 1962). As long as the costs of medical care remained low relative to the demands of other sectors of the economy this responsibility did not prove to be a severe financial burden to most communities, and for the few for which it was (frequently rural areas), public opinion at large was not sufficiently agitated to demand corrective action at the national level. As these costs have risen sharply over the last twenty years however, under the stimulus of changing health care patterns and increasingly sophisticated medical technology, the disparities between the tax bases of rich and poor communities, defined all too often upon urban-rural lines, and the attendant differentials in the quality of available care which this has caused, have become both medically and politically intolerable.

In response to these needs increasing support from both provincial and national sources has now become the norm, and although local revenue resources will continue to be of some importance in the financing of local medical services for a considerable time to come, this role will inevitably be diminished in the future.

As local financial autonomy has declined so has the opportunity arisen for the rational planning of health facilities upon a more realistic geographical basis than was formerly feasible. The evils of unco-ordinated planning have been well documented in the health planning literature (see for example Letourneau, 1965; Cardwell and Kucka, 1961). The duplication of expensive equipment and personnel in adjacent or nearby hospitals; the installation of specialized care facilities at locations where the demand is not sufficient to warrant them; and a virtually de facto competitive spirit among some hospitals to install facilities with little regard to the overall needs of their dependent service areas, all bear testimony to the inefficient use of existing medical resources.

Concomitant with these abuses are a series of parallel problems which are a reflection of the maldistribution of resources over the continent at large. Inadequate and low quality facilities are common in many outlying and underprivileged areas of both Canada and the United States. The distribution of facilities and personnel appears, in many instances, to be more a reflection of the economic

status of a community than its actual health needs, and the organization of that health care suffers all too frequently from a closer correspondence with local political jurisdictions, than with the natural movement of people along well defined links of communication (Weinerman, 1951). Few organizations too, are as fragmented as the varied components of the health care system. Following Schaefer, Hilleboe defines five levels of organization at which this fragmentation occurs. First there is a horizontal fragmentation, caused by a partitioning of responsibility often ill defined, at the federal, state and local levels. Secondly, vertical fragmentation into a multiplicity of agencies, commissions, departments, divisions, bureaus, and offices, each of which occurs at the three horizontal levels of organization. Sectional arrangement of health programs among public, private and voluntary organizations provides a third level of fragmentation. Next is the fragmentation of health activities along professional and technical lines, and finally fragmentation exists in the financing of health services through a variety of insurance and public grant schemes (Hilleboe, 1968). The inevitable consequence of such divisions is to be seen in ill-defined roles, overlapping responsibilities and poor communication among the varied levels of the organization, all of which contribute to a quality of medical care, below that which is now technically possible.

In response to these problems and to the unmet needs which are evoked by them has arisen the concept of regional planning, or the regionalization of medical facilities as it has come to be known. A succinct definition of this concept has been given by Donabedion and Axelrod (1961):

> *. . . regionalization envisages the unified planning of a functionally differentiated and carefully co-ordinated system of hospitals serving an entire geographic region demarcated, not by narrow political boundaries, but according to established patterns of seeking and providing medical care in a manner analagous to trading areas. Under such a plan, standards of medical care could be considerably improved throughout the region without costly duplication or inefficient deployment of scarce resources and skills. There would be established appropriately located central or base hospitals to which other small hospitals in the region would turn for help and advice. The central hospital would, in turn, assume considerable responsibility for supporting continuing professional education and medical and allied services in the hospitals associated with it.*

Historical Antecedents for Regionalization

The earliest formal documentation of these concepts appears to have been contained in the Dawson Report, published in 1920, by the Ministry of Health in Great Britain. In this report, subsequently the basis for the organization of the British Health Service in 1947, was embodied the notion of a network of primary and secondary hospitals affiliated by strong links with a larger teaching hospital (Ministry of Health of Great Britain, 1920). The teaching hospital was envisaged as acting both as a referral centre for the smaller community hospital and as a base from which

medical, technical, and administrative assistance could be offered wherever necessary for the provision of higher quality care. The first expression of these ideals in North America came eleven years later in 1931 when the Bingham Associates Fund, partially under the stimulus of widespread public demoralization caused by the severe economic depression of 1929, organized the earliest system of regional health care in the United States. This program, based upon the New England Medical Centre in Boston, set out to offer a complete diagnostic and surgical service to the patients of referring physicians, and to provide a continuing post-graduate educational program for doctors in the affiliated network of smaller community hospitals, which covered most of Maine and much of northern Massachusetts (Lemboke, 1951).

But despite the promise of this early and successful beginning little more was achieved over the next fifteen years. The philosophy of individual responsibility in all spheres of life, not least in medical care, runs deep in North American life, and the strength of this tradition tended to foster an attitude of apathy, if not hostility, on the part of the general public towards wide-ranging regional programs (Sigmond, 1967). It was not until the passing of the Hill-Burton Act in 1946, that the stage became set for the subsequent widespread growth of regionally oriented health care programs.

The Hill-Burton Act was implemented as a direct consequence of a severe shortage of doctors and facilities in rural areas, and was focused principally upon efforts to attract physicians into these remote locations and to provide them with well equipped centres from which to operate. But the legislation has had a significant impact upon regional medical planning. Organization of the program at the state rather than the local level has strengthened the role of state health departments, a necessary prerequisite for the foundation of a wider and more functional pattern of health care. At the same time the delegation of the responsibility for fund raising to both state and community agencies has helped to stimulate interest in the health planning group and has served to dissipate some of the emotional opposition to the regional planning concept.

For all its strengths however there are certain inbuilt restraints in the Hill-Burton legislation which have prevented it from becoming the basis for a really effective system of medical regional re-organization. Most significant among these has been the reliance placed upon local participation in the financing of construction projects. Depending upon state and local per capita income levels this amount has come to vary from between one to two-thirds of the total cost of the project. Unfortunately those locations with the greatest need for new facilities have proved all too often to be those which are most economically depressed and unable to raise even the minimum revenues

necessary to qualify for federal aid. As a consequence there has been a tendency to withhold funds from precisely those areas in most urgent need of them (Battistella, 1966). Secondly, the emphasis of the program has been upon the construction of new beds to the exclusion of the development of broadbased co-ordinated services at different levels of state and federal organization. In McNerney's view "too many hospitals may have been constructed in communities where real need may not have been so much for in-hospital beds as for well-equipped diagnostic or emergency clinics." (McNerney, 1962) A final failure of the Hill-Burton program has been seen in its voluntary tripartite structure of federal, state and local governments which has proved to be an unwieldy means for regional reorganization. Progress has been retarded in many states by outdated governmental machinery which is unrepresentative of the needs of communities, and by the lack of viable health organizations professionally trained and empowered by sufficient authority to carry through reforms.

The failings of the Hill-Burton Act provided lessons which were not forgotten in the preparation of the Heart Disease, Cancer and Stroke Act of 1965. This piece of legislation, sometimes known as the Regional Medical Programs Act, marked the first significant attempt on a national scale in North America to organize health services into a regional framework. The act specifically encouraged the creation of regions whose boundaries were to be defined in

terms of medically functional criteria. As such, the delimitations of the regions have, of necessity, cut across local and state lines in some instances, thereby offsetting some of the major shortcomings of the Hill-Burton Act. Insofar as the program has been successfully implemented services have been organized in the interdependent hierarchical structure fundamental to the regionalization concept. Decisions concerning the precise nature and range of services offered within each region have been decentralized and are being undertaken by local medical planning groups which are, in theory at least, more sensitive to the needs of local areas. Finally, and perhaps more important of all, the cost of the program has thus far been underwritten by grants from the National Institute of Health, and is therefore independent of the financial status of each region and its ability or otherwise to pay for the services it requires (Sanazaro, 1967; Kissick, 1967).

Reports on the current status of the Regional Medical Programs Act indicate that a successful beginning has been made, but as with any program on this scale of magnitude and innovation, problems have arisen from time to time to impede its rate of progress. Commitment from many institutions has been on a tentative basis so far owing to the limited duration of current financial support for the program. Assurances of longer support are seen as being vital to the achievement of the long term objectives of the plan (Stewart, 1967). The program has also been adversely

affected by a serious shortage of medical manpower in the United States. These shortages have placed restraints on the rate of implementation of some of the program's goals. Neither it is known to what extent this program's requirements for manpower have diminished the supply available to other deserving medical care schemes. This limitation has resulted necessarily in a relatively greater emphasis being given to medical training activities in the initial operating phases of the plan, and has created the false impression in some quarters that the Regional Medical Programs are primarily designed for this task.

In the Canadian context the concept of regional planning has been firmly endorsed by the report of the Royal Commission on Health Services and has been implemented to some extent ever since the establishment of the federal governments grant-in-aid program for construction in 1948. In the past however, regionalization, which is organized on the provincial level in Canada, has suffered from too much centralization at the top and a lack of sensitivity to local needs. In Wahn's view an inherent defect has been that "those who ultimately have the obligation to implement the plan are not actively involved in the preparation of the plan" (Wahn, 1967).

More recently a movement towards decentralization, in the planning function at least, has taken place, and both British Columbia and Ontario have now established a number of regional planning areas. The British Columbia plan came

into effect in 1967 with the creation of twenty-seven regional hospital districts throughout the province. In design these districts are seen to have several interdependent functions. Each area has established an Advisory Planning Committee whose task is to review the projects for construction and expansion submitted by the management boards of individual hospitals in the district, and to fit them into a priority plan (which may include recommendations for the creation of new facilities) which is then submitted to the central planning authority for approval. One major purpose of the plan has been to provide a grant-in-aid scheme which offers a more equitable way of sharing construction costs over the entire area which uses a hospital's facilities. As presently constituted the provincial government pays sixty per cent of construction costs in districts where the levy is up to four mills and eighty per cent in regions where the assessment is greater than this. The remainder is accounted for by district and federal contributions. Organization of services on this basis thus allows the local share of construction costs to be raised on a broader tax base than was formerly possible. Some commentators see possible long term weaknesses in the financing arrangements. In particular the 60:40 per cent basis for cost sharing is not considered adequate by some planners especially in the case of the larger referral hospitals, which treat many patients from beyond their district boundaries, but receive no extra recompense for this service (Anon, 1967).

Some Inherent Problems in Regional Planning

It is too early as yet to evaluate critically the successes of regional planning on this continent. Changes in the basic philosophy of organizational structure on this magnitude inevitably give rise to problems unforeseen and to roles as yet not firmly defined. To some degree this has already become apparent. Apart from such technical considerations as the prior delimitation of regional areas (see Klarman, 1964) experience with area wide planning agencies has frequently demonstrated a number of conflicting interests and purposes.

Co-operation among agencies and other planning bodies is clearly fundamental to successful planning on the regional level but where, as is often the case, different agencies or even departments within the same planning board, have different technical backgrounds and orientations, there has been a tendency to define and identify community health service needs in different ways. This in turn has encouraged selective attention upon the means to solve particular problems, rather than the prior first step of critically evaluating the underlying problems themselves (Arnold and Hink, 1968).

A second group of problems reflects a lack of definition in some instances of the roles which each of the various components of health care is to play in the overall context of health planning. The focus of the present concept of regional planning on a hierarchical structure of

hospital referrals needs to be extended to include all public health organizations and other related bodies, whose interests have a bearing upon both preventive and curative medicine. The total systems approach to health planning has been advocated by several authors (see for example Weinerman, 1969). Hilleboe has extended the idea further by urging the development of strong links where possible with other community planning objectives. In his view health planning is inseparable from these other goals. The question of the total resources available to meet all planning demands is one which is often overlooked by health administrators. The construction or modification of health facilities needs to be matched against the capacity of available building resources, and estimated manpower requirements should be a reflection of available or potentially available educational facilities if realistic objectives are to be attained (Hilleboe, 1968).

The Role of the Modern Hospital

In any definition of roles that of the modern hospital in the organization of health care is clearly one of fundamental significance. Many of the problems which have resulted from the past construction of facilities designed for the treatment of specific diseases have long been recognized, particularly where the control of infectious ailments has been the major objective. The tuberculosis sanitorium, long a familiar sight upon the landscape, has been rendered

almost obsolete by medical advances over the last twenty years (Hoge, 1967). But geriatric and psychiatric needs are on the increase in modern society, and account for a proportionately larger share of medical resources with every passing decade. The role of the community general hospital with respect to these needs is still as yet, a matter of some debate.

The continued use of separate long term geriatric and psychiatric care is sometimes thought desirable on the grounds that large scale incorporation of these services into the structure of the general hospital will diminish its efficiency by virtue of diseconomics of scale, and will, at the same time, prove to be inaccessible to many of the patients homes, a factor of some significance where long term care is involved. Against this it may be said that the consequences of such segregation are potentially more serious, particularly in an age where modern transport has increased the range of acceptable travel distances. The most intractable problem associated with the maintenance of separate institutions for this type of care is that of obtaining suitable qualified staff. Geriatric and Psychiatric hospitals have for long been regarded as depressed areas by both the public and the medical profession. Appointments to these positions are not, by and large, competitive, and staffing shortages remain a perennial factor, with all the attendant effects on the level and quality of medical care that this implies. Moreover segregation from

the district hospital centre tends to depress the interests of medical research into these ailments, and the isolation of the psychiatric patient suffers the further disadvantage of perpetuating public prejudices about the particular nature of this disorder (McKeown, 1966). Patients segregated upon the basis of such disease characteristics seldom, if ever, remain homogeneous with respect to their medical needs. Chronically ill patients may develop symptoms of acute illness while geriatric patients will frequently show signs of mental disorders. The net effect is that the institution in question either finds itself responsible for classes of patients whose needs are beyond its resources or must provide those resources itself, thereby encouraging a further duplication of facilities and personnel.

The alternative approach, which is now favoured by many medical administrators, is to abolish separate institutions for these types of care and to provide the whole spectrum of care in the community general hospital. Proponents of this policy argue that long stay units at the district hospital need not be inordinately expensive since chronically and psychiatrically ill patients need not be admitted to and retained in wards designed for the more expensive types of acute care. Because of the relative ease with which patients can be transferred within a single hospital complex, there seems to be no inherent reason why special long stay units cannot be constructed at lower

capital and maintenance costs for this specific purpose. Such units might further serve a double function by acting as convalescent homes for patients who no longer need acute care but who must remain hospitalized for a rehabilitation period (Bridgman, 1967).

The Role of Social Medicine

Thus far this discussion has centred upon the provision and organization of the curative aspects of hospital care. The emphasis on curative medicine, particularly in the hospital setting, made sense in the past when acute disease of the infectious variety was the overriding medical problem, and chronic disease was of much less significance owing to the shorter span of life which then prevailed. But this situation is no longer common, at least in technically developed parts of the world, and the role of public health programs and social and preventive medicine which has received less attention in the past, is now seen as being of potentially greater consequence in the future. The need for alternative approaches to in-patient hospital care takes on an added significance if the opinion of those who believe that hospital use is directly related to the supply of beds can be given credence (Roemer, 1961). Although Rosenthal has taken a contrary viewpoint in this controversy, Feldstein supports Roemer's findings, and in his analysis of hospital utilization in England, is drawn irresistably to the conclusion that the existing supply of beds is the

correct one, whatever its level may be (Rosenthal, 1964; Feldstein, 1967).

The search for alternatives to hospital care requires some prior distinction of the concepts of "need" and "demand." If demand can be loosely defined as the actual receipt of medical care the concept of need is clearly more ephemeral in nature. Whether or not vast health needs actually exist as has sometimes been claimed is actually a debatable point. Indeed many physicians who practice among insured urban populations are of the opinion that the demands for health services often exceed the true needs (Task Force Report, 1970). But whatever the truth of the matter is, it seems clear that one of the primary aims of preventive medicine is to keep patients, where possible, from reaching the level of in-patient hospital care.

Preventive medicine has already achieved some striking successes in the control of infectious diseases. Vaccination programs, conducted on a mass basis, have virtually eliminated many of the contagious infections which took such a heavy toll in the past. Mass screening programs have offered additional opportunities for preventive medicine. But although they have proved themselves in the control of preliminary tuberculosis, there is much less concensus on their current overall worth as a widespread tool for the early detection of many types of disease. Many of these programs have proved to be expensive propositions when conducted upon a large scale, and they have not always proved to be as medically reliable as their proponents have

claimed. It seems clear that more cost effectiveness studies are required in this field and, at the very least, it is felt that they should be directed more selectively towards high risk groups in the population (Task Force Report, 1970).

One of the recurring problems which faces social medicine is that the public perception of what constitutes a serious health hazard is not always coincident with that of the medical profession. Cigarette smoking, often cited as a case in point, offers an excellent example of the type of conflict which can occur between, on the one hand, the financial benefits bestowed by a particular industry on the national treasury, and the aims and aspirations of public health programs on the other. James has aptly summarized this kind of ambiguity in attitudes by remarking that public pressure for the control of many medical problems is apt to set quite different priorities from those of the health professional. "Diseases," he comments, "appear to run an unpopularity contest before the public and it is rarely settled by incidence statistics" (James, 1967).

One of the keys to greater success in preventive medicine would seem to lie in an increased understanding of the complex processes by which individuals are motivated to seek health care and by the way in which patients move through the medical care system. In considering some of the mechanics of this process it is known that an intricate network of formal and informal interactions exists between

patients or potential patients on the one hand and the providers of health care, the physicians and other medical personnel, on the other. The events which characterize these interactions occur in particular settings in which attitudes which reflect social, economic and cultural attributes have a profound influence on the outcomes (Donabedion, 1968). Many studies have been made of the utilization of health services by patients of widely varying backgrounds. From these certain generalizations may be deduced which show in effect that facilities are utilized most frequently by the young middle-aged, by females rather than males, and by those who have achieved a higher educational and socio-economic status. By corollary those individuals of lower educations attainments and increased age would appear to react more slowly to these symptoms and to delay in seeking help (Odoroff and Abbe, 1957; Feldstein, 1966). These conclusions would seem to be equally valid for those who are using preventive and detection services and for those who are seeking actual diagnosis and treatment. Friedson, Suchman and others have undertaken more sophisticated analysis in which linkage mechanisms between social characteristics and individuals perceptions of their symptoms and their responses to them are proposed (Suchman, 1965; Friedson, 1961). Rosenstock who has reviewed some of these studies comments upon the specific limitations of their focus which is confined to health behaviour undertaken in response to specific symptoms. In his words:

> *The findings are thus of unknown relevance to the situation confronting the person who must decide whether to seek preventive or detection services before the appearance of events that he interprets as symptoms.* (Rosenstock, 1966)

The author then goes on to propose a model which attempts to explain health behaviour in individuals who believe themselves to be presently free of any definitive symptoms. He hypothesizes that the decision to seek preventive medicine in such a situation is dependent upon the extent to which an individual perceives his susceptibility to some particular condition; the extent to which he perceives its possible occurrence as being serious to him; and the degree to which he feels that some preventive test is appropriate for him insofar as it will diminish his perceived susceptibility. Once these conditions are obtained, then some cue of stimulus, either internally, or externally such as an impact from some communication media, is required to trigger an appropriate reaction (Rosenstock, 1966).

The validity of Rosenstock's model remains largely unknown at the present time, since there has been little empirical evidence thus far to substantiate these hypotheses. Such evidence as is available seems to suggest, as Rosenstock himself notes, that the behaviour patterns implicit in the health belief model tend to be exhibited more among the upper socioeconomic groups than the lower. It is precisely among these lower status groups however, that social medicine has been least effective to date, and where the need is

consequently greater. The classical referral pattern whereby patients seek the services of a physician and are referred by him where necessary to a specialist, or the lay referral system in which patients refer themselves directly to a specialist are less frequently seen in these social groups (White, 1967). The problem is further compounded by a relative inaccessibility of all types of medical care to these people. The decline of the general practitioner in North American medical practice is a well known phenomenon. A recent study by Dorsey of the changes in the distribution of physicians in Boston between 1940 and 1961 revealed a decline in the ratio of physicians per 100,000 population for all socio-economic areas of the city. For social classes 3-5, defined in the study as the lower status groups, this trend is particularly disturbing since the areas occupied by these groups have relatively low physician ratios to begin with (Dorsey, 1968). If, as in Friedson's view, convenient access to medical care is important to patients especially in terms of routine primary care, then the inference is that the urban poor will, in many cases, either ignore their initial symptoms, or refer themselves to the emergency department of the nearest hospital, a medically undesirable habit which appears to be very much on the increase in nearly every major community on the continent (Friedson, 1961; Coleman, 1967).

The decline of the general practioner and the flight of other physicians and specialists to the suburbs or to

offices located adjacent, in many instances, to hospital complexes, has left a medical vacuum for many potential patients in the slums and ghetto areas of the larger cities. The concept of the neighbourhood health centre or clinic has been proposed as an answer to this medical gap. Yerby has described the essential features of these centres. They are community oriented in the sense of being located with respect to significant concentrations of the lower income groups; are open day and night; provide basic diagnostic and theraputic services; are staffed not only by doctors, laboratory technicians and frequently required specialists, but also by public health nurses and social workers; and have a direct ambulance link with some hospital for immediate referral where necessary (Yerby, 1967). It is hoped that such centres will ultimately diminish unnecessary hospital in-patient admission, and will do much to reduce the pressure on hospital emergency departments which are not thought to be the appropriate milieu for primary health care.

Insurance Coverage and the Utilization of Medical Care

Interest in alternative approaches to hospital care has been further encouraged by speculation as to the potential long term effects of medicare and government sponsored insurance plans on in-patient hospital utilization. There is a considerable amount of evidence to show that medical services are used more frequently by those who carry health

insurance. Indeed the corollary of this—that people may be deterred from obtaining necessary health care by virtue of their lack of coverage is obviously the reason behind the recent expansion of the role of governments in the medical insurance field (Task Force Report, 1970). Studies in the United States suggest that hospital admission rates are much higher among the insured than the uninsured (Somers and Somers, 1962). Whether or not this actually leads to over-utilization is a debatable issue among medical administrators since the hospital length of stay is generally shorter for the insured person. In Canada the relationship between hospital admission rates and insurance coverage appears to be less clear. Andersen and Hull state that the introduction of universal hospital insurance into this country has not produced a marked increase in the rates of admission (Andersen and Hull, 1969). Peterson on the other hand asserts that unnecessary admissions are occurring among some patients admitted purely for diagnostic purposes (Peterson, 1967). It may be (although there is no direct evidence as yet) that this apparent contradiction reflects a slight but perceptible change in the physician culture of Canada which traditionally, appears to have been hospital oriented. Thus where unnecessary hospital admissions for diagnostic purposes occur, they may, to some extent, be counterbalanced by the movement towards group practice which, as the Task Force has tentatively indicated, results in a reduction of hospital usage.

Medical Care in the Rural Setting

Many of the difficulties which beset health administrators in the provision of medical care in metropolitan areas are magnified in the context of rural life. Rural health care is burdened by some particular problems which are peculiar to its setting. Although the mortality rates for rural inhabitants are less than for those of their urban counterparts morbidity during life appears to be greater, especially for chronic diseases (Roemer and Anzel, 1968). The provision of medical care for migrant agricultural workers and for Indians, who do not fit easily into the mainstream of North American life, presents almost unique difficulties. The benefits of good occupational health services are generally denied to rural workers. Fundamental to rural areas too, is the low population density which is reflected in greater distances between patients and the services they might expect to receive. The provision of these services is further compounded by the very low density of health personnel per unit area. The lack of health manpower in rural areas is a perennial problem. The ratio of medical professionals to population is less in the rural setting for every category of health personnel (Mott and Roemer, 1968). For the most part small rural communities have fewer advantages to offer the young physician when competing for his services in the open market. In particular the amenities available, both technical and social, are fewer (McNerney and Riedel, 1962). Moreover, even if it

were possible to attract personnel into these areas in large numbers, there is considerable doubt as to whether it would be worth the cost in terms of the degree to which they could be fully employed. Greater financing is then, clearly not the answer in itself. What is required for these areas is a flexible and mobile system of medical care which can be adequately incorporated into the concept of the regionalization of fixed facilities as previously described.

To some extent these goals have already been achieved. A number of schemes have been proposed and implemented. Flying doctor services and the use of helicopters in emergency situations have been comprehensively documented in the health planning literature (Duncan, 1966; Jacobs and McLaughlin, 1967). The training of para-medical personnel, who possess a limited number of the skills of the qualified physician, have been suggested for use in extremely remote locations (World Health Organization, 1954). In more accessible areas mobile health units have been advocated as offering an excellent method of providing care to communities too small to maintain a permanent health facility (Bodenheimer, 1969; Kilsdonk, 1966). Such units have been valuable in the treatment of endemic infectious diseases in Africa, and as a preventive medical aid, particularly for tuberculosis screening, in the more developed nations. Physicians in rural areas have been encouraged to pool their resources, records and equipment in particular central locations, and to maintain a series of simple satellite offices in

surrounding communities at which they spend a few hours per week at specified days and times (Canadian Medical Association, 1967).

Although in all these ways a measure of much needed flexibility has been introduced into the rural health system there still remains the fundamental problem of the provision of a number of fixed facilities, such as clinics or hospitals, which can serve as a basis for the organization of rural medical care, and to which patients can be sent when these other measures prove inadequate to their needs. Bearing in mind the costly financial burden of maintaining hospitals in the rural context it is perhaps surprising that so little objective analysis has been undertaken on the multiple location of medical facilities in this setting. As recently as 1966 Lord Llewewyn-Davies could say,

> *Very little study has been made with regard to the location of health buildings . . . the lack of study of this topic is striking when the literature of urban planning and development is examined.* (Lord Llewelyn-Davies, 1966)

Since that statement was made a few researchers have turned their attention to this problem and a brief review of their models now follows.

Some Previous Models of Health Facility Location

At least three studies in recent years have considered methods by which the locational efficiency of existing or potential hospital networks might be measured. In the first of his two studies on this problem Schneider considers the

case of urban hospitals with particular reference to Cincinnatti (Schneider, 1967). He postulates that the locational imbalance of any one hospital can best be assessed by analyzing the net pulling force of hospital in-patient flows. Given discrete residential locations to which a weight, representing numbers of patients is attached, he seeks to find among them the point of minimum aggregate travel by means of the Kuhn-Kuenne algorithm (Kuhn and Kuenne, 1962). The difference between this computed centre and the actual location is interpreted as a measure of the locational inefficiency of the hospital in terms of the area it serves.

Several aspects of this conceptualization deserve further consideration. The Kuhn algorithm requires Euclidean distance measures as its inputs, and while these may be appropriate in the urban context it is unlikely that they will be so in a rural setting. There is no guarantee either that the optimal solution as computed by this method will be constrained to a settlement or even adjacent to a road. Finally this approach can consider only one hospital at a time although heuristics have been developed by Cooper for applying the algorithm to multiple plant location problems (Cooper, 1967).

In a later model Schneider extends his analysis to locational imbalances in hospital systems. For each medical category the centre of minimum total travel based on patient residences is computed as before. Next this centre is found

with respect to hospital locations, the weights in this instance being the hospital admission figures. The differences between the two points is taken as an initial measure of imbalance in the system to which is added the disparity between the relative dispersions (standard distances) about the two centres, under the assumption that the greater the degree of patient dispersal relative to hospital dispersal, the greater the spatial inefficiency of location. Although this formulation has merit over the previous one it suffers from the disadvantage, as noted by the author, of exaggerating the standard distance measures in cases where patient demand is clustered into two or more spatially distinct areas (Schneider, 1968). Moreover, while this approach will give insights into imbalances in the present system it has less to say on where the best set of locations might be if more than one is desirable. It is clearly possible to postulate new locations and re-cycle the algorithm in an attempt to reduce the imbalance measure, but if the number of possible sites is large a combinatorial problem may be encountered which has no feasible computational solution.

Gould and Leinbach studied the problem of rural hospital locations in Guatemala. From five possible locations they set out to find those three which would minimize total patient distance travelled. They based their approach upon the classic transportation model. Setting initial hospital capacity to be equal at each site, they evaluated every possible combination of three out of the five potential

locations, assigning the flows from each patient area to the current set of hospitals in solution by means of a linear programming algorithm. By inspecting the best solution so obtained, inefficient cross flows were eliminated by adjusting the hospital size inputs and re-cycling the algorithm until a satisfactory flow pattern emerged (Gould and Leinbach, 1966). In fact, if an all combinations approach is adopted then the linear programming aspect of this solution procedure is quite unnecessary. All that is required is to assign the demand at each patient area to the closest one of the three hospitals currently in the solution. This procedure will give the optimal solution, and provide in addition the appropriate hospital size which will not necessarily be found by the Gould/Leinbach method. However, since the number of combinations rises very rapidly both as a function of the number of locations required and of the number of potential locations, this approach will rapidly become infeasible for even moderately sized problems.

Morrill and Kelley present a different model whose basic objective is to measure the efficiency of the location of capacity over an operating system of hospital sites. The authors make use of a modified gravity interactance algorithm in which patient flows from areas to hospitals are conceived to be not only an inverse function of distance, but are also related directly to hospital size and to the sizes and distances away of intervening opportunities (Morrill and Kelley, 1970). This conceptualization has behavioural

merit in the sense that size can be considered to be a crude reflection of the hospital referral function and as such patients will not necessarily be constrained to admittance at their nearest facility. As with the Huff (1963) and Lackshmanan and Hansen (1966) model initial use is made of the interaction formulation to predict flows between residences and facilities. The divergence of the predicted from actual hospital capacity is considered by the authors to be a measure of the imbalance of the location of capacity. The particular contribution of Morrill and Kelley is in the algorithm by which hospital capacity is relocated so that aggregate patient travel is reduced but in a manner which is still consistent with the basic structure of the interactance model. In their words

> *An allocation can be considered optimal if the model is interpreted as maximizing the joint satisfaction of a heterogeneous population at minimum distance, but where satisfaction requires the use by some people, perhaps part of the time, of a variety of opportunities.* (Morrill and Kelley, 1970; p. 286).

The re-allocating algorithm adjusts the sizes of the hospital by reducing capacity at underpredicted locations and increasing it at overpredicted ones. This step which amounts to using the predicted capacities as inputs, is repeated until some acceptable level of convergence is attained between two successive iterations.

It should be noted however, that after the initial allocation the calculated divergence may be as much due to imbalances in the gravity model as to actual imbalances in

the location of capacity. Furthermore the reduction in aggregate patient travel which might be achieved will not necessarily be one in which the capacity of location has been optimally redistributed. Several equilibrium states may exist and it is possible to devise network structures where this occurs to within quite low tolerance levels (under 5 per cent), and where the algorithm converges from a given starting point to some local minimum. An alternative method suggested here is to begin the cycling with an initial set of hospital sizes based not on the current operating levels but upon a set of values derived from allocating each area of patient demand to its nearest treatment centre. It is conjectured that this procedure will tend to give a better redistribution of capacity since the direction of adjustment to achieve consistency with the gravity structure is always away from the minimum distance allocation. This suggests that the equilibrium obtained upon termination of the iterations will tend to be nearer the desired goal than any other feasible solution. Experiments with a very limited number of small hypothetical examples have borne this out in some instances, and in other cases where both methods converged to the same approximate solution the proposed algorithm required fewer iterations to converge, particularly where the starting point was badly out of balance.

It is possible to test the viability of new facilities with the Morrill-Kelley algorithm by introducing

potential sites into the network and re-allocating flows according to the interactance mechanism in much the same way as was proposed by (Huff, 1966) for retail outlets. This model cannot however, test for the direct efficiency of location in the sense of solving for the optimal system of locations with respect to all possible sites. It is this particular problem with which this study is principally concerned.

A General Approach to Rural Hospital Location Systems

As suggested previously in this chapter the interrelationships among the many components of the health care system through which the patient receives medical care are very complex and not fully resolved as yet. The path of some patient to treatment at any hospital may be the culmination of a complicated chain of events. The role of accessibility in this process has received considerable documentation in the past (Morrill and Earickson, 1970; Marrinson, 1966). Even in metropolitan regions, where the pattern of cross flows can be expected to be complex, distance minimizing tendencies exert a significant influence (Lubin et. al., 1966). This principle is however, frequently violated by other offsetting factors. For the physician with hospital privileges the choice of preferred office location will tend to be influenced by accessibility to the hospital in which he practices, and where this coincides with a business district a considerable proportion of

the clientele may be expected to be in attendance during their working hours. Since the choice of hospital for treatment is normally dictated by the physician's preference this cannot always be expected to coincide with the patient's general area of residence. Case mix differences are also of importance particularly with respect to the less frequently utilized spectrum of services. Referrals to specialists and to hospitals for specialised treatments will clearly tend to offset the distance minimizing principle. Other hospitals which have particular religious or ethnic affiliations may draw patients from any part of the city. Finally there are a whole range of less well defined variables related to the quality of available services and the prestige of institutions and practitioners, which can and do complicate the whole pattern of patient flows.

Such intervening opportunities are, as is to be expected, much less in evidence in the rural context, where distances between communities resolve the problem into a study of inter rather than intra urban locations. Apart from specialist referrals (less than 10 per cent of all patients) the accessibility variable appears, with some exceptions, to be much the most dominant force. Since this is generally so the problem of optimizing hospital locations in rural areas can be conceptually reduced to one of finding the medians of a weighted graph subject to certain side constraints. Such a model is devised in the following chapter and tested in the fourth. The testing procedure

for assessing the locational efficiency of a set of sites over a given network is to measure the differences in total distance travelled between two solutions, one constrained to the existing system of locations and the other unconstrained. If a more realistic trip distribution pattern is desired, one in which patient areas are not allocated in block to a single centre, then these solutions should be modified by an interactance model. These various applications are undertaken in Chapter IV, together with a practical comparison of the two methods previously discussed for redistributing capacity over an operating location system.

CHAPTER III

AN ALGORITHM FOR ESTIMATING THE MEDIANS OF A WEIGHTED GRAPH SUBJECT TO SIDE CONSTRAINTS

Problem Statement

If hospital location can be regarded as a special case of the generalized plant location problem then the solution requires the determination of the number, sizes and locations of plants which will service at a minimum overall cost, a given set of demands at discrete points over space. The unit transport costs or distances between each demand point and potential location site must be known in advance, and if the number of specified plants is not known at the outset it will be assumed that a production cost function is associated with each potential plant location.

Because of the economies of scale inherent in plant production costs this problem, both theoretically and computationally, is one of considerable complexity. At the present time no undisputed exact analytical solution has been developed for the case where the plant cost functions are continuous and concave over the whole of their feasible range. It is known however, that under scale economies unit production costs will rise as the number of plants

increases, whereas conversely, the transportation costs will fall (Figure 1). In the twin cost plant location problem the goal therefore, is to achieve an equilibrium among these two opposing sets of functions such that the total cost level is minimized.

The real nature of many plant location problems is such that it is frequently desirable to impose, where possible, a further set of conditions and constraints upon the system. Such constraints can take a wide variety of forms and may include restrictions placed upon the size of plant operation, upon the maximum flow allowable over any one link, or upon some predetermined maximum distance any demand point may be from some supplying source. The number of potential locations may be less than the number of network nodes, or again there may be a priori reasons for forcing certain locations into the solution as, for example, when extensions to an existing distribution system are required. Multiple product allocation problems, stochastic demand functions and warehouse locations, intermediate between plant sources and customers, represent further potential variations upon the general formulation.

In this chapter a numerical method is developed for obtaining optimal or near optimal solutions to the plant location problem, subject where applicable, to a specific subset of all feasible constraints. Some advantage is taken in this model of an ingenious partitioning heuristic given

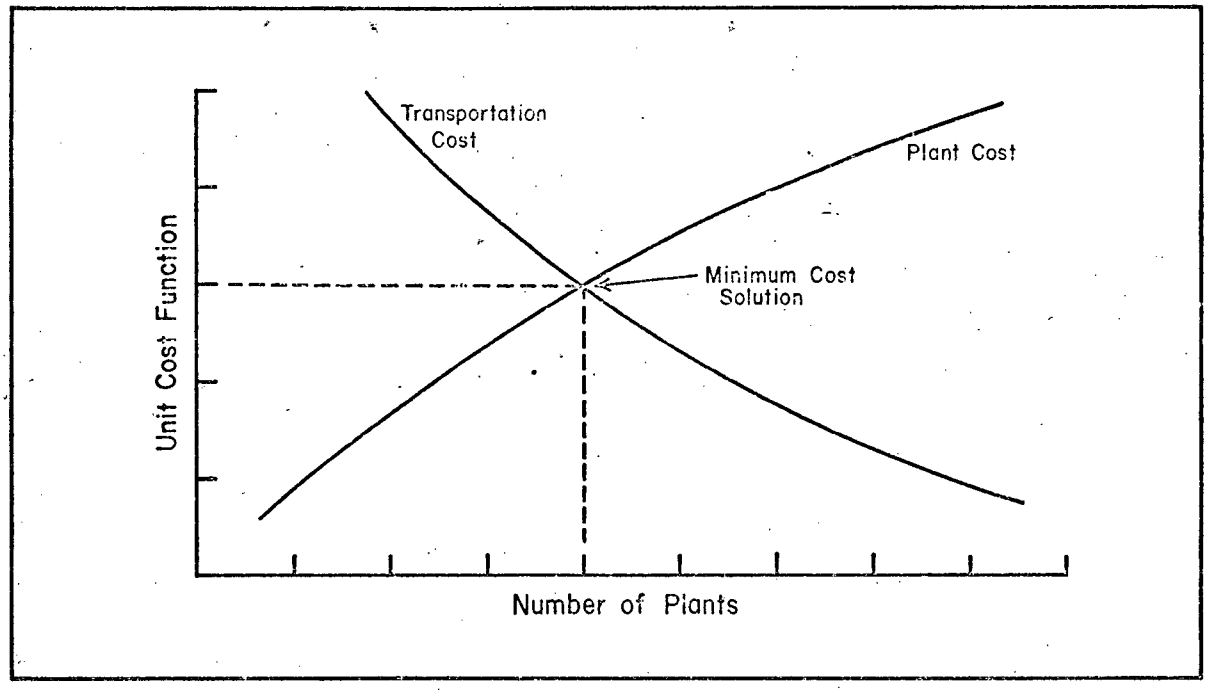


Figure 1
Relationship Between Plant and Transportation Cost Function

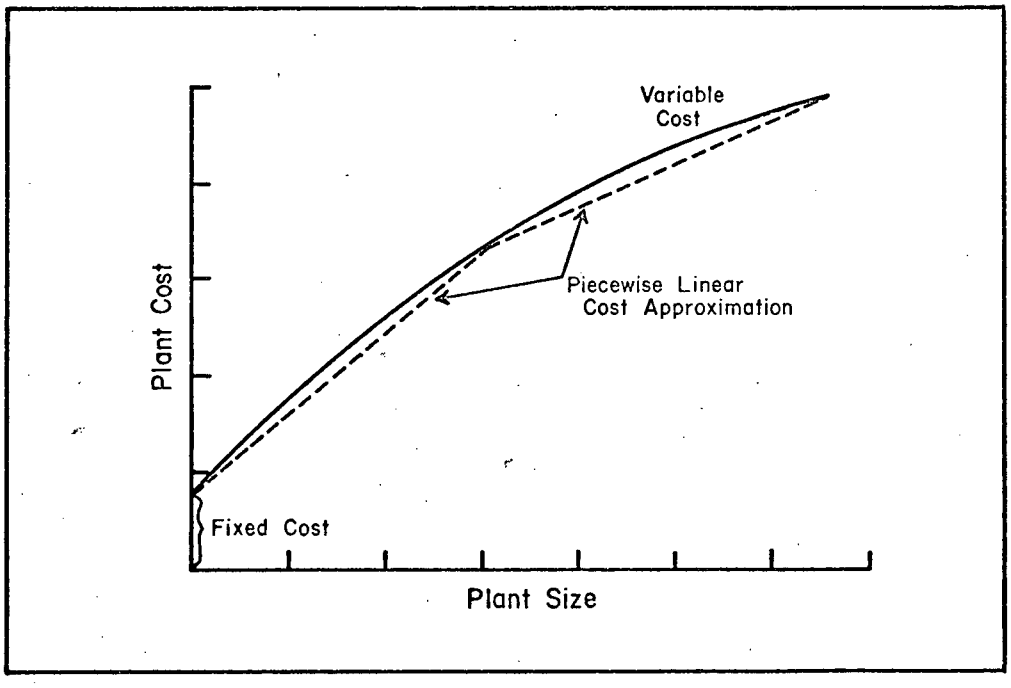


Figure 2
Piecewise Linear Approximation of a
Concave Plant Cost Function

by (Singer, 1968) whose contribution will subsequently be reviewed. The particular problem specification to be considered here can be expressed as follows:

Given;

(1) The location of each destination.

(2) The demand at each destination for a single product commodity considered as a fixed non-stochastic variable.

(3) A vector of potential plant site locations.

(4) A subset of locations to be forced into the solution—if necessary.

(5) An upper bound on the size of each potential plant site—if optioned.

(6) A constraint on the maximum distance any destination may be from some supply source—if optioned.

(7) A set of unit transport costs (currency, time or distance) between each destination and each potential supply point which, since they are averaged unit values, may represent non-linearities.

(8) A vector of production functions—if optioned—of any type, which may or may not be identical for each prospective site.

Then determine;

(1) The number of supply sources required (unknown only if plant costs are optioned).

(2) The location of each supply source.

(3) The size of each supply source.

(4) The allocation of destinations to each supply point.

(5) The amount supplied from each plant source so chosen to each destination so allocated (may be fractional only if plant size upper bounds are imposed).

Such that the overall cost is minimized.

Mathematically this is a mixed integer programming problem of the form:

$$(1) \quad \text{Min} = s = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij} D_{ij} \lambda_i + \sum_{i=1}^{m} f_i (T_i) \lambda_i$$

subject to

$$(2) \quad X_{ij} \geq 0 \qquad (i=1,\cdots,m \; ; \; j=1,\cdots,n)$$

$$(3) \quad \sum_{i=1}^{m} X_{ij} = B_j \qquad (j=1,\cdots,n)$$

$$(4) \quad \sum_{j=1}^{n} X_{ij} = T_i \qquad (i=1,\cdots,m)$$

$$(5) \quad \sum_{i=1}^{m} T_i = \sum_{j=1}^{n} B_j = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij} \qquad (i=1,\cdots,m \; ; \; j=1,\cdots,n)$$

$$(6) \quad T_i \leq C_i \qquad (i=1,\cdots,m)$$

$$(7) \quad D_{ij} \leq V_j \qquad (i=1,\cdots,m \; ; \; j=1,\cdots,n)$$

$$(8) \quad \lambda_i = 1 \qquad (i=1,\cdots,k)$$

$$(9) \quad \lambda_i = \begin{matrix}(0)\\(1)\end{matrix} \qquad (i=k+1,\cdots,m)$$

Where

n = The number of destinations

k = The number of forced plants in solution (if any)

m = The total number of potential plant sites (including forced nodes)

X_{ij} = The commodity flow from plant i to destination j

T_i = The volume of plant i

D_{ij} = The cost of transporting one unit of X_{ij} from plant i to destination j

C_i = The plant size upper bound constraint on plant i

B_j = The demand at destination j

V_j = Maximum distance (transport cost) constraint on destination j

f_i = Plant cost function at plant i

λ_i = A zero one variable; a one indicating a plant site is in the solution, a zero indicating it is not.

Formulation of the Problem in Terms of a Weighted Graph

For the purpose of subsequent exposition this mathematical specification of the problem may be rephrased in terms of a weighted graph. Consider a connected graph G with n vertices labelled v_i, $i = 1,\cdots,n$ and k branches or links. Each link has a length >0 associated with it such that this length is the distance from some v_i to some v_j. In addition each node has attached to it a weight b_i where $b_i > 0$ and is the demand at v_i. If $b_i > 1$ for at least one

node on the network a weighted graph exists; otherwise the graph is said to be non-weighted. The basic problem, without constraints, is to find a set of p nodes on the graph such that the weighted distances from this set to all other nodes on the graph is a minimum over any such set of p.

Designate any set of p nodes on G $(x_1, x_2, \cdots, x_p)$ as X_p and let $D = |d(v_i\ v_j)|$ an n by m distance matrix in which each element $d(v_i\ v_j)$ is the value of the shortest path from vertex i, $i=1,\cdots,n$ to vertex j, $j=1,\cdots,m$, where m is the number of potential plant sites $m \leq n$. From $|D|$ generate a new matrix $|W|$ such that $w(v_i\ v_j) = d(v_i\ v_j)b_i$ over all i and j. This weighted distance matrix $|W|$ is then, equivalent to the multiplication of each row i in $|D|$ by the corresponding weight b_i of the vertex v_i. For each node v_i let

$$(10) \qquad w(v_i\ X_p) = \min \left[w(v_i\ x_1), w(v_i\ x_2), \cdots, w(v_i\ x_p) \right]$$

that is, find for each node its minimum weighted distance to some member of the set X_p. The required set of p nodes X_p^* on G is the p median of the network if, for every other possible set of X_p on G.

$$(11) \qquad \sum_{i=1}^{n} w(v_i\ X_p) \geq \sum_{i=1}^{n} w(v_i\ X_p^*)$$

Some Previous Approaches to the Problem

As Scott has recently provided an excellent critique of plant location algorithms only a short survey of the current methods will be reviewed here (Scott, 1970). There are, briefly, two possible approaches to the problem; exact solution techniques and heuristic numerical methods.

(a) Exact Solution Methods

If plant costs are ignored then the basic location problem for any fixed value of p can be neatly structured and solved in a linear programming format (Revelle and Swain, 1970). The inclusion of production costs adds considerably to the complexity of the problem, and as previously noted, the value of p is not known at the outset. An exact solution method for cases where the plant costs are continuously concave over their whole range is given by Rutenberg (Rutenberg, 1968), but it remains to be conclusively shown that this algorithm will always converge to the optimal solution. Efroymson and Ray (1966) take a different, and computationally faster approach by approximating the concave cost function with piecewise linear segments (Figure 2), and solving again in an extended programming form. As a general rule the initial linear programming solution will not meet the integer restriction which requires each potential plant site to have a value of zero or one depending upon whether it is, or is not, active in the final solution. If each variable not satisfying this conditon is now given

one of these values, and if every alternative is considered, a combinatorial tree can be constructed which represents the entire space of all feasible solutions. The generation of such a tree establishes a set of terminal states, each one of which is an integer solution. In practice it is not possible to search all of these, and the approach taken by Efroymson and Ray is to devise a branch-bound algorithm which will ignore evaluation of all such alternatives as lie along those branches of the tree whose descendent solutions are known, a priori, to be infeasible by virtue of some previously established lower bound integer solution.

The general nature of branch-bound algorithms can be readly illustrated with reference to Figure 3, although this is not precisely the method used by Efroymson and Ray. Consider a problem with n destinations and m possible plant sites where $m \leq n$. Let m have the value of 3 and designate the 3 possible sites as X_1 X_2 and X_3. Any subset of this set may contain the optimal solution. Solve the problem first as an integer unconstrained linear program with a minimum cost of, say, equal to 70. All subsequent integer restricted solutions must be at least as great as this. Set X_1 equal to 1 and solve again to obtain a solution of 75. At this stage assume that neither X_2 nor X_3 has integer coefficients. Branch next on $X_2 = 1$ and obtain a bound of 78. Set $X_3 = 1$ and solve again with $S = 85$. This is one terminal state on the tree, an integer feasible solution.

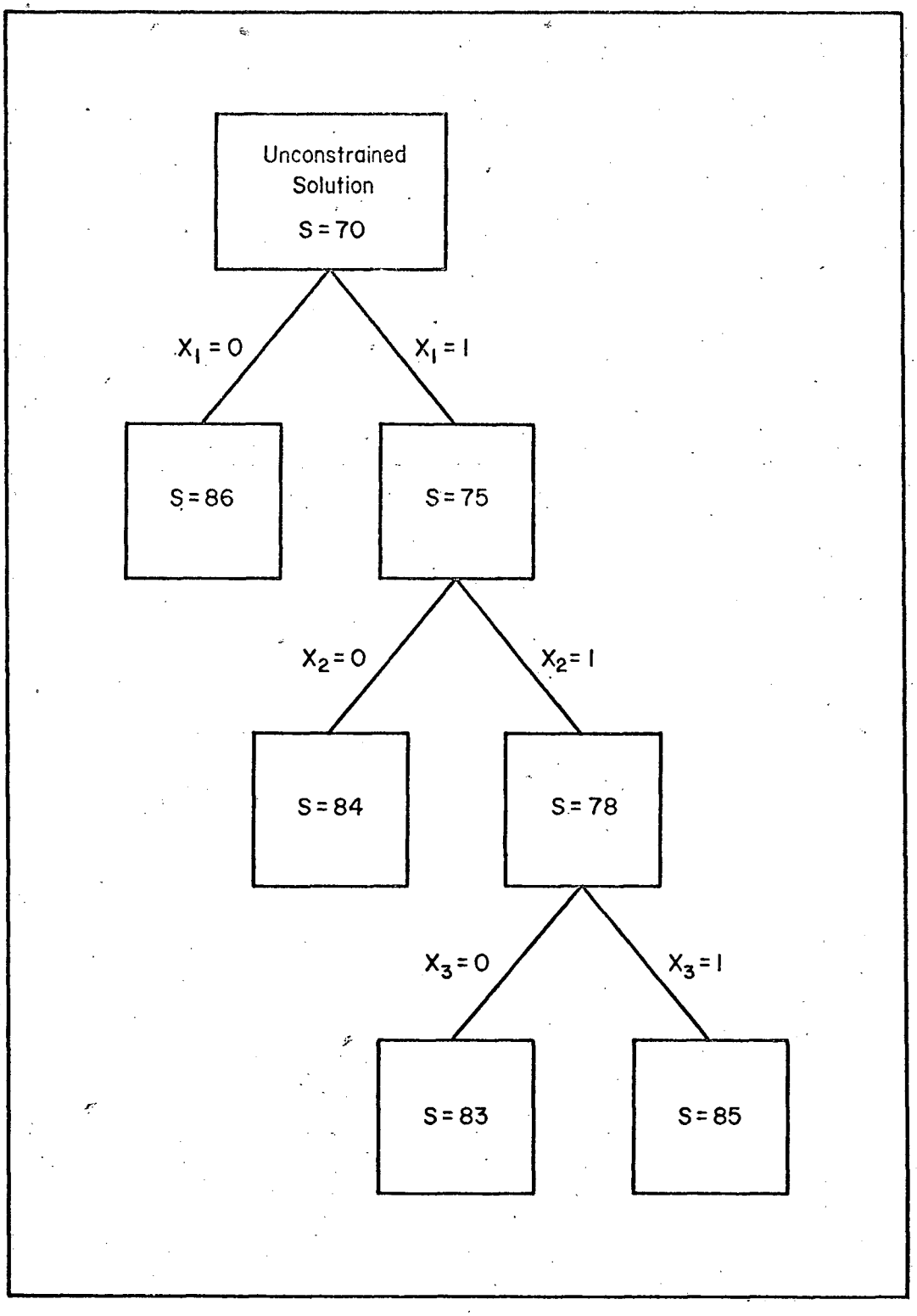


Figure 3
Principles of the Application of a Branch-Bound Algorithm to Plant Location

Now backtrack down the tree and set $X_3 = 0$, keeping $X_1 = 1$ and $X_2 = 1$. A new lower bound with the objective function $S = 83$ is established. Next relax the explicit integer constraint on X_3, continue down the tree and set $X_2 = 0$. In solving this linear program with $X_1 = 1$, $X_2 = 0$ and X_3 unconstrained the obtained solution is 84. It is not necessary therefore to continue the search along this branch of the tree since restraining X_3 to either 1 or 0 must result in a solution $\geq$ to 84, the bound obtained when X_3 was unconstrained. When the constraints are relaxed on both X_2 and X_3, and X_1 is set =0, the unconstrained solution (in terms of X_2 and X_3) again exceeds the lower bound on the best previously established integer solution for which $S = 83$, and any further evaluation in this direction is therefore quite unnecessary (see also Scott, 1969; White, 1969).

The disadvantages of the exact solution approach are generally acknowledged to be two-fold. Although the linear approximation of the concave cost curve may be achieved to any desired degree of accuracy, computer storage requirements tend to become excessive as the number of segments increases. It is more difficult to generalize upon the computational efficiency of branch bound algorithms since if a good integer solution is quickly found the amount of searching along the branches of the tree can be greatly curtailed, and the algorithm may rapidly converge to the

global minimum. As a rule however, these models tend to become computationally over-extended for the larger sized problems, and this tendency is considerably enhanced with each additional constraint that is placed upon the system.

(b) Heuristic Solutions

Heuristic algorithms generate, by contrast, solutions characterized by a much greater degree of computational efficiency at the expense of known accuracy in the final product. The two approaches can be used to complement each other if the branch-bound algorithm can be structured in such a way as to take advantage of the heuristic solution as an initial starting bound. In this manner the number of alternatives evaluated by the exact solution method may be greatly reduced if the heuristic solution is near optimal. It is even possible, since the concave cost functions need not be approximated in heuristic models, that this approach may actually find a lower cost solution.

Kuhn and Hamburger (1963) suggest an algorithm in which it is assumed that the best p sources will be contained among the best p+1 locations. This method locates the pth plant at that site which offers the largest cost savings over the set of sources contained in the p-1 solution. The value of p is incremented sequentially until such time as the p+1 solution exceeds in cost the pth solution, at which point the algorithm terminates. At each

stage in the cycling procedure some modification of the current location system is allowed for by a bump-shift routine which eliminates all such plants in solution as have become uneconomical as a result of the generation of subsequent plants.

Feldman et. al. (1966) consider the problem from the reverse order and begin with all in potential plant sites in solution. At each step one plant is eliminated until no further cost reduction is attained. This is the point at which scale economies achieved by increasing plant size as the number of sources diminishes is offset by greater transportation costs which accrue from the smaller number of plants now in solution. An initial estimate of plant size, necessary for the beginning solution, is derived from each plant's local customer set defined as being those destinations to which each plant is closest to in terms of transportation costs alone.

The basic disadvantage of these methods is that on any cycle in which the value of p changes the set of current sources may tend to be too dependent upon the vector of plants defined in the antecedent step. Two other fundamental approaches to the problem have been advocated in which for any value p the network may be independently broken into p descendent subgraphs. There are the vertex substitution method and the partition approach.

In the first of these a set of p points is chosen (often arbitrarily) as the initial source configuration.

Each node v_i is now assigned to that member of the set to which it is closest and a lower cost bound S, the objective function is calculated for this set of p, X_p. Some vertex v_j, not a member of X_p is now selected and in turn each $v_i \varepsilon X_p$ is replaced by v_j. If a lower bound of S is found during any one of these p substitutions then this set replaces X_p. If not, then some other v_j is chosen and the process is repeated. The algorithm terminates when a complete cycle has been made during which no reduction in S is found. Variations on this method has been proposed by Teitz and Bart (1968), and by Scott (1969). Note that this method involves a fairly extensive search pattern, which, while falling far short of a direct enumeration of all p combinations on the graph, is likely to become computationally prohibitive for the larger scale problem with several hundred nodes.

For networks of this magnitude the partitioning approach appears to offer the best alternative for estimating the required p median. Such methods can be generalized in terms of three basic steps.

i) Apply some criterion to partition the graph into some initial set of X_p nodes. Call this set the beginning set $X_{\bar{p}}$.

ii) Next every other node or destination is assigned to some member of $X_{\bar{p}}$ by (10). Each subgraph G_i, $i = 1,\cdots,p$ now has an associated subset of nodes $v_r \varepsilon G_i$.

and for each such subgraph G_i there exists some node v_{ki} (not necessarily unique) for which

$$(12) \qquad \sum_{r \in G_i} w(v_r\ v_{ki}) = \min = S_i$$

where v_{ki} is designated as the median of subgraph G_i and S_i is the value of v_{ki}, or the aggregate weighted distance from v_{ki} to all other nodes in G_i where this value is at a minimum for one such node (v_{ki}) in G_i. The value

$$(13) \qquad S = \sum_{i=1}^{p} S_i$$

over all possible partitionings of G into p subgraphs is the value whose overall minimum is sought. To test, at any stage, whether at least a local minimum obtains the following condition,

$$(14) \qquad w(v_r\ v_{ki}) \geq w(v_r\ v_{kj}) \quad j=1,\cdots,p-1\ ;\ j \neq i$$

is examined for each G_i and for each node $v_r \in G_i$.

If this condition holds then v_r may be switched from G_i to G_j, notably to that G_j for which $w(v_r\ v_{kj})$ is a minimum if v_{kj} is not unique. The value of S so obtained will be less than the previous value of S if (14) is a strict inequality, and will be less than or equal to it if (14) is satisfied as an equality. Step (ii) is reiterated until

such time as

$$w(v_r\ v_{ki}) \leq w(v_r\ v_{kj}) \tag{15}$$

is met for all v_r over all G_i. Alternative solutions may be obtained in instances where condition (14) exists as an equality, and will occur when the median of some subgraph(s) is not unique and condition (15) is met for all the combinations of the p medians of the subgraphs.

iii) The procedures of step ii) ensure at least a local minimum which may or may not be the optimal solution. Most partitioning methods contain some means of testing this lower bound by perturbing the solution; that is switching nodes according to some rule and re-calculating step ii).

The actual performance of any partitioning algorithm is clearly dependent upon the logical strength of the heuristics adopted for steps i) and iii). Although random starts will provide fast solutions they are generally much less desirable than systematic partitioning procedures. A heuristic given by Cooper consists of beginning with the set of all potential sites and eliminating one member at each step until only p nodes are left (Cooper, 1967). The elimination criterion (for Cooper's graph is non-weighted) is to take the closest pair of points on any iteration and to drop that one whose estimate of the single p median of the network is the higher. After completion of

step ii) the perturbation test on the lower bound first finds some G_i such that for any two nodes v_i and v_j, $d(v_i\ v_j)$ is a maximum over all G_i. This median is then dropped to be replaced by both v_i and v_j. Since X_p now contains p+1 members the two closest medians are selected and in turn each one is dropped while step ii) is repeated.

The Singer Algorithm

Singer has developed an alternative algorithm of very considerable merit in which he presents an ingenious method for obtaining $X_{\bar{p}}$ based upon the attainment of a set of p nodes which are the maximum weighted distance apart. Let $|A|$ be an n by n (or m by m if the number of potential sites is less than n) symmetric matrix in which each a_{ij} element is the weighted distance apart of some v_i and some v_j, $v_i \neq v_j$. Then set $a_{ij} = d(v_i\ v_j)\ b_i\ b_j / (b_i + b_j)$ which is equivalent to $a_{ij} = w(v_i\ e) = w(v_j\ e)$ where e is the fulcrum between each of the pairwise weighted vertices. For the non-weighted graph in which all $b_i = 1$, each a_{ij} value is simply equal to one-half of the shortest distance between each pair of vertices (Singer, 1968).

The aim behind this and most other partitioning criteria is to obtain that set of $X_{\bar{p}}$ nodes which, ceteris paribus, are the least likely in the final solution to be in the same subgraph set. Since node weights are traded against the distances between nodes, the Singer criterion

provides a logically sound springboard from which to begin the search procedure. In matrix $|A|$ the maximum a_{ij} element will designate the two nodes v_i and v_j which are the maximum weighted distance apart. For $p > 2$ $X_{\bar{p}}$ is the required set of p nodes if, and only if, the minimum pairwise a_{ij} values among all pairs of nodes v_i, $v_j \varepsilon X_p$ is a maximum over all such minimum values in every other set X_p in G.

A computational method for obtaining $X_{\bar{p}}$ when $p>2$ (evaluation of all combinations of X_p being generally infeasible) is given by Singer and may be illustrated for the case of $p = 3$.

i) Let a_{ij} be the maximum value in row i in the A matrix.

ii) Scan down the rows of $|A|$ and choose some row r for which maxmin $(a_{ri} \times a_{rj})$ is satisfied. That is pick the row in which the minimum of a_{ri} or a_{rj} is a maximum over all r(r=n) rows. The associated node v_r is the tentative third node.

iii) Designate a_s as the minimum of $(a_{ij} \times a_{ri} \times a_{rj})$.

iv) Repeat steps ii) and iii) for all pairs of nodes v_i and v_j in row i for which $a_{ij} \geq a_s$. The set of p nodes for which the minimum value of a_s is a maximum is the candidate set from this row.

v) Repeat steps i) to iv) for all rows and choose that set of 3 nodes $(v_i\ v_j\ v_r)$ for which a_s, as defined, is a maximum over all rows.

Generalizing to $p > 3$, find in each row i the p-2 maximum elements. There are p-1 indices associated with these elements, that is p-2 indices which correspond to the columns in $|A|$ containing the maximum elements, plus the index of the row itself. It is possible to speed up Singer's computational procedure by omitting step iv) until all rows of the matrix have been scanned at least once. This increases the probability of finding an initially good value for a_s. However in the author's experience even this improvement can be computationally exacting if not infeasible, should the actual required maximum value of a_s happen to be relatively small, or should a good upper bound for it not be established early in the search. Consider the example of a 100 node network with p=6. For any row i after one complete scan there may be many nodes, in terms of step iv), which exceed the current value of a_s. This implies therefore that there may be many combinations of p-1 nodes (the first member of which is always the row index) that require reiteration of steps i) to iii). If the value of p is large in both absolute terms and perhaps more significantly relative to m, then the number of combinations to be evaluated can easily become computationally prohibitive.

An alternative method for obtaining a good estimate of X_p^- in terms of the Singer criterion and which avoids the above difficulties may now be given. The principle is

to add nodes one at a time applying the maxmin criterion at each step. Satisfaction of this criterion will determine the next vertex to be added. Note that in the Singer method p - 1 elements are chosen prior to the application of the maximum criterion which is utilized only to find the pth node for the candidate set from that row. In adding each new node by this suggested method it is not necessary in updating a_s to re-scan the columns of $|A|$ belonging to those nodes already assigned to the candidate set, if the following procedure is adopted. For each row i (let p = 3 for illustration).

i) Scan the ith column in $|A|$ and store all a_{ji} values, $j = 1,\cdots,m$, $j \neq i$, in some among say $|BC|$. Pick the maximum a_{ji} value $\equiv a_s$ and store the associated node v_r where r denotes the row index in which a_s was found. The candidate set row has two members, v_i and v_r.

ii) Branch next on v_r alone; that is scan only column r in $|A|$. Consider again all elements a_{jr}, $j = 1,\cdots,m$, $j \neq i$ and $j \neq r$. That is ignore the row indices of all the current members of the candidate set. Now if some $a_{jr} < bc_j$ then replace the jth element in $|BC|$ with that a_{jr} value. Thus the values now stored in $|BC|$ of those nodes not yet in the beginning set are always the minimum values in $|A|$ with respect to those

members which are currently in $X_{\bar{p}}$. For any node v_j then, $v_i \neq v_j$ and $v_j \neq v_r$ its value in $|BC|$ is always $\min(a_{ji}, a_{jr})$.

Now search among BC excluding bc_i and bc_r and find some bc_j = max, storing v_ℓ the associated node. Let $a_s = bc_j$ max = the minimum of $(a_{ir}, a_{\ell i}, a_{\ell r})$. This is the maximum of the minimum pairwise values of any other ℓ with i and r in $|A|$.

iii) If $p > 3$ repeat step ii) until all p vertices have been chosen, the candidate set $X_{\bar{p}}$ for this row. Let $a_m = a_s$ the maximum value over p elements (a_s always decreases with each additional node).

iv) Repeat steps i to iii) for all i rows. If the last value for a_s for any row is $> a_m$ replace a_m with a_s and store this set of nodes in preference. If $a_s = a_m$ store both sets of nodes if they differ, for these are alternate sets of $X_{\bar{p}}$.

This partitioning heuristic may be illustrated with reference to the network displayed in Figure 4. In this and all subsequent example the node references are designated as v_i, $i = 1,\cdots,n$, the b_i value of each vertex is encircled, and the distance between each pair of nodes is the figure lying adjacent to each link. The distance of each vertex with itself is assumed to be zero and the short distance and A matrices for this

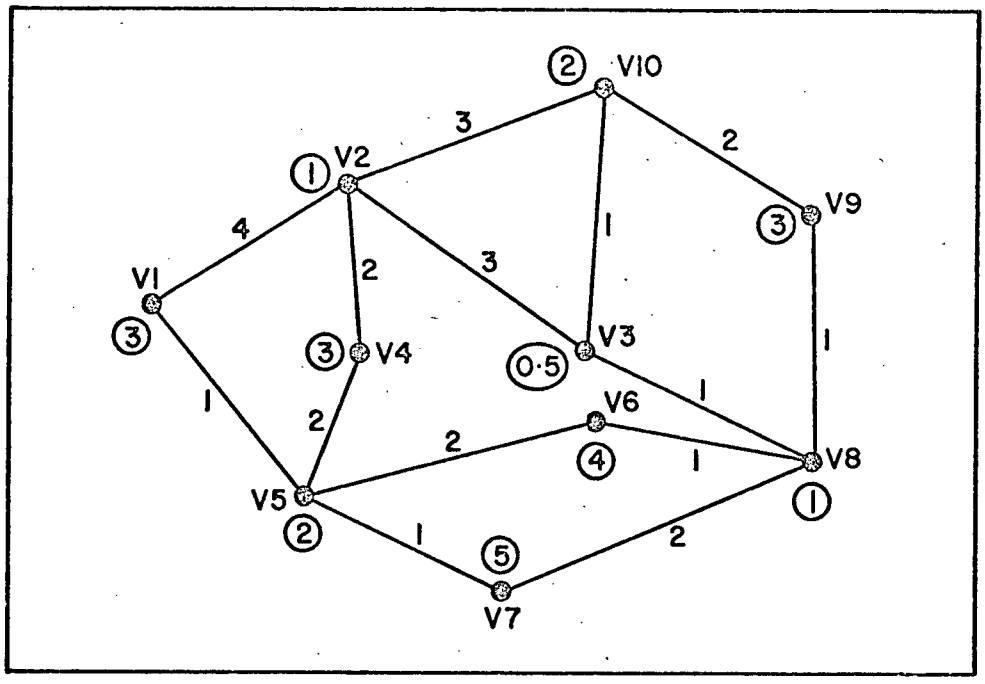


Figure 4

Table 1

SHORT DISTANCE MATRIX FOR FIGURE 4

0.0	4.0	5.0	3.0	1.0	3.0	2.0	4.0	5.0	6.0
4.0	0.0	3.0	2.0	4.0	5.0	5.0	4.0	5.0	3.0
5.0	3.0	0.0	5.0	4.0	2.0	3.0	1.0	2.0	1.0
3.0	2.0	5.0	0.0	2.0	4.0	3.0	5.0	6.0	5.0
1.0	4.0	4.0	2.0	0.0	2.0	1.0	3.0	4.0	5.0
3.0	5.0	2.0	4.0	2.0	0.0	3.0	1.0	2.0	3.0
2.0	5.0	3.0	3.0	1.0	3.0	0.0	2.0	3.0	4.0
4.0	4.0	1.0	5.0	3.0	1.0	2.0	0.0	1.0	2.0
5.0	5.0	2.0	6.0	4.0	2.0	3.0	1.0	0.0	2.0
6.0	3.0	1.0	5.0	5.0	3.0	4.0	2.0	2.0	0.0

Table 2

THE |A| MATRIX FOR FIGURE 4

0.0	3.0	2.1	4.5	1.2	5.1	3.7	3.0	7.5	7.2
3.0	0.0	1.0	1.5	2.7	4.0	4.2	2.0	3.8	2.0
2.1	1.0	0.0	2.1	1.6	0.9	1.4	0.3	0.9	0.4
4.5	1.5	2.1	0.0	2.4	6.9	5.6	3.8	9.0	6.0
1.2	2.7	1.6	2.4	0.0	2.7	1.4	2.0	4.8	5.0
5.1	4.0	0.9	6.9	2.7	0.0	6.7	0.9	3.4	4.0
3.8	4.2	1.4	5.6	1.4	6.7	0.0	1.7	5.6	5.7
3.0	2.0	0.3	3.8	2.0	0.8	1.7	0.0	0.8	1.3
7.5	3.8	0.9	9.0	4.8	3.4	5.6	0.8	0.0	2.4
7.2	2.0	0.4	6.0	5.0	4.0	5.7	1.3	2.4	0.0

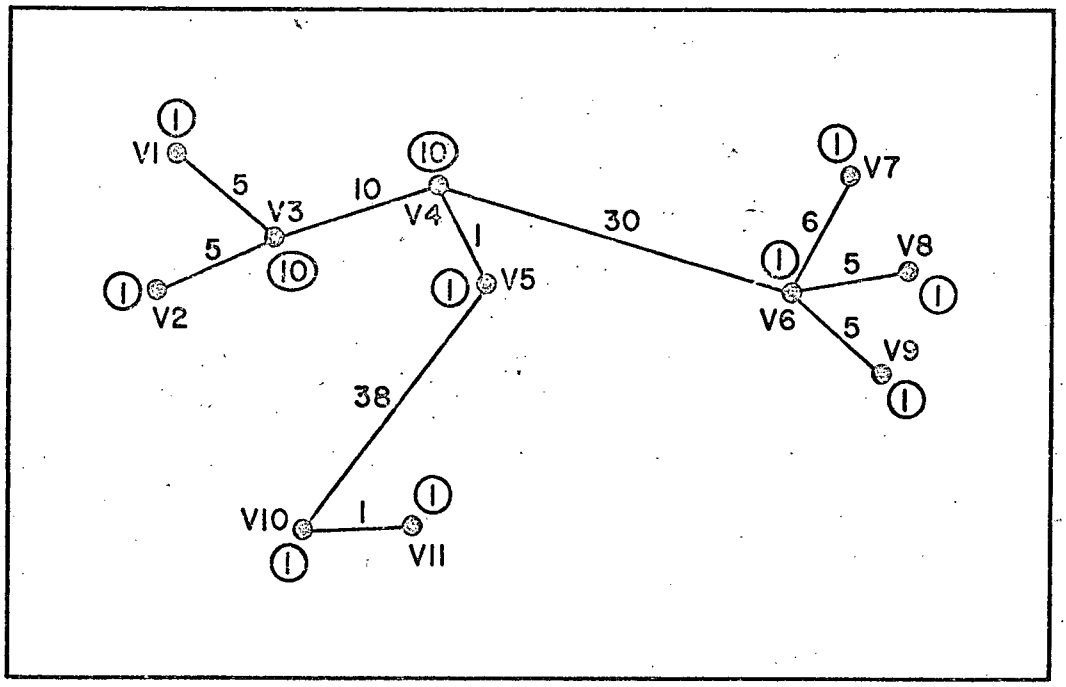


Figure 5

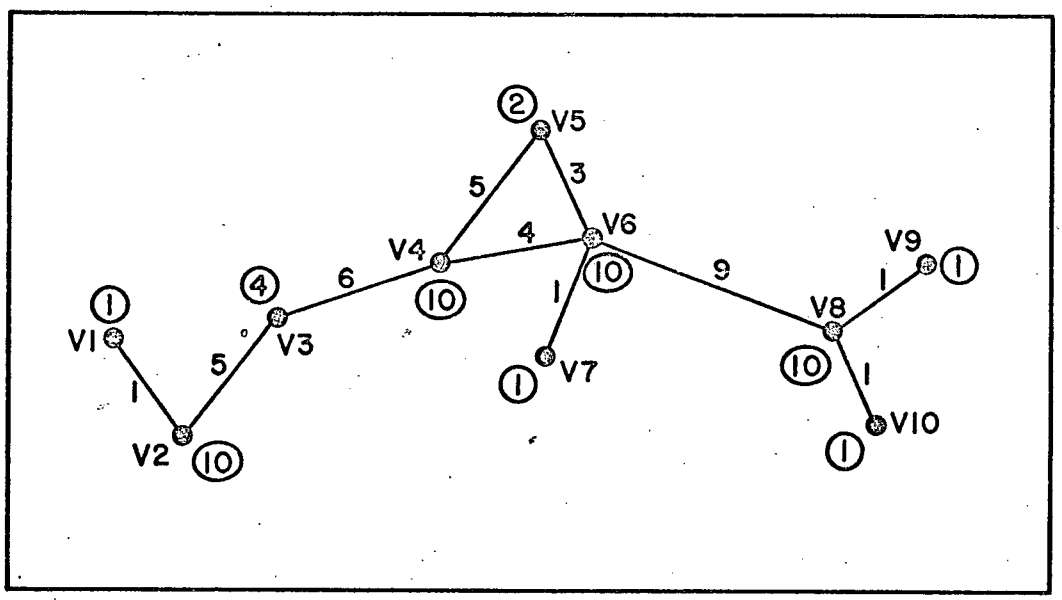


Figure 6

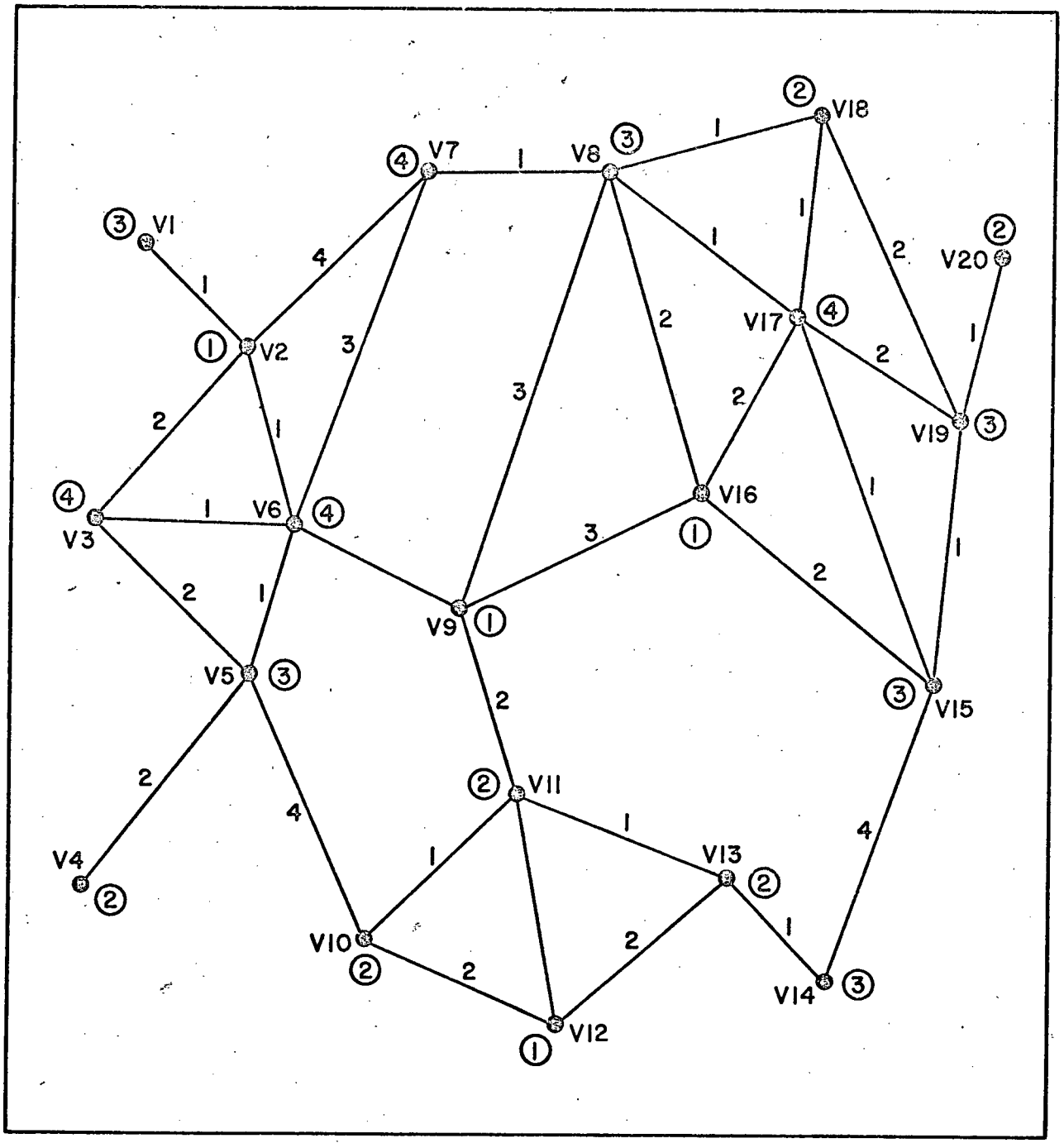

V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19
V20

Figure 7

PROBLEM NAME ..... FIGURE 8(A) 2 MEDIAN SOLUTION FOR FIGURE 4

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	5	14.00	18.00	0.00	18.00	1	2	4	5	7	3.0	1.0	3.0	2.0	5.0
2	8	10.50	11.50	0.00	11.50	3	6	8	9	10	0.5	4.0	1.0	3.0	2.0
NETWORK		24.50	29.50	0.00	29.50										

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	5	13.00	14.00	0.00	14.00	1	4	5	7		3.0	3.0	2.0	5.0	
2	8	11.50	15.50	0.00	15.50	2	3	6	8	9	1.0	0.5	4.0	1.0	3.0
						10					2.0				
NETWORK		24.50	29.50	0.00	29.50										

PROBLEM NAME ..... FIGURE 8(B) 3 MEDIAN SOLUTION FOR FIGURE 4

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	4	4.00	2.00	0.00	2.00	2	4				1.0	3.0			
2	5	10.00	8.00	0.00	8.00	1	5	7			3.0	2.0	5.0		
3	8	10.50	11.50	0.00	11.50	3	6	8	9	10	0.5	4.0	1.0	3.0	2.0
NETWORK		24.50	21.50	0.00	21.50										

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	4	4.00	2.00	0.00	2.00	2	4				1.0	3.0			
2	7	10.00	8.00	0.00	8.00	1	5	7			3.0	2.0	5.0		
3	8	10.50	11.50	0.00	11.50	3	6	8	9	10	0.5	4.0	1.0	3.0	2.0
NETWORK		24.50	21.50	0.00	21.50										

PROBLEM NAME ..... FIGURE 8(C) 3 MEDIAN SOLUTION FOR FIGURE 5

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	3	22.00	10.00	0.00	10.00	1	2	3			1.0	1.0	20.0		
2	4	25.00	137.00	0.00	137.00	4	5	6	7	8	20.0	1.0	1.0	1.0	1.0
						9					1.0				
3	10	2.00	1.00	0.00	1.00	10	11				1.0	1.0			
NETWORK		49.00	148.00	0.00	148.00										

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	3	22.00	10.00	0.00	10.00	1	2	3			1.0	1.0	20.0		
2	4	25.00	137.00	0.00	137.00	4	5	6	7	8	20.0	1.0	1.0	1.0	1.0
						9					1.0				
3	11	2.00	1.00	0.00	1.00	10	11				1.0	1.0			
NETWORK		49.00	148.00	0.00	148.00										

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED				VOLUME SUPPLIED TO EACH MEDIAN			
1	3	22.00	10.00	0.00	10.00	1	2	3		1.0	1.0	20.0	
2	4	23.00	80.00	0.00	80.00	4	5	10	11	20.0	1.0	1.0	1.0
3	6	4.00	16.00	0.00	16.00	6	7	8	9	1.0	1.0	1.0	1.0
NETWORK		49.00	106.00	0.00	106.00								

Figure 8

graph are given in Tables 1 and 2 respectively. For the two median solution of this network the members of the set $X_{\bar{p}}$ for which a_{ij} is a maximum are v_4 and v_9 with a value of 9. For the three medians of this graph the initial partitioning is based upon each of the three sets of $X_{\bar{p}}$ $(v_4\ v_7\ v_9)$, $(v_4\ v_6\ v_7)$; or $(v_4\ v_7\ v_{10})$. The minimum pairwise distance between any two nodes in these three sets is a maximum over all such sets in the network with a value of 5·625.

The lower bound perturbation test suggested by Singer is to array first the subgraphs in decreasing order of the values S_i. Then for each subgraph G_i, that node $v_i \in G_i$ whose weighted distance to the median of some other subgraph G_j is least, i.e. for which $w(v_i\ v_{kj})$ is a minimum, is switched to subgraph G_j and the resultant subgraphs are modified until condition (15) is satisfied. This procedure is repeated p times moving one node on each occasion. If a lower bound is found on any iteration then the whole testing process is again re-cycled. The solutions to the two and three medians of Figure 4 were actually obtained without recourse to the perturbation test and are shown in Figures 8a and 8b.

Some Comments on These Algorithms

One considerable advantage of Singer's algorithm is that for each value of p required, the graph is repartitioned. This means that the $X_{\bar{p}} + 1$ or $X_{\bar{p}} - 1$ beginning set is not necessarily dependent upon the set of $X_{\bar{p}}$ present in the pth or antecedent step.

It is not too difficult in practice however, to design networks in which neither the Singer nor Cooper criterion provides a good initial starting basis. Consider the network shown in Figure 5 for which the three median solution is required. Both of these methods (if weighted rather than simple distances are considered for Cooper's criterion) give a set of $X_{\bar{p}}$ containing nodes v_3, v_4 and v_{11}, which converge to a local minimum of S = 148·0. As it happens Cooper's perturbation test finds the global optimum (see Figure 8c) whereas Singer's test does not. In this respect Cooper's method seems to be consistently the better of the two approaches. Singer's perturbation routine suffers from the further disadvantage that at least p re-assignments and modifications must be made on each testing cycle, so that for big networks where p is also large, this task could conceivably become over-extended.

The basic problem with both the partitioning and perturbation heuristics of these and other algorithms is that nodes are evaluated individually and without sufficient consideration of the weights and distances away of their immediate neighbours. As a consequence biases can occur towards the inclusion of solitary nodes which lie far removed at the periphery of the network. This circumstance is sometimes prevalent with networks of a tree or dendritic design. Some clustering procedure which considers the relationship of each node to its surrounding neighbours is clearly desirable if this effect is to be diminished. In deriving $X_{\bar{p}}$ it is always possible to artificially increase the attraction of some particular site by adding to its weight the demands of all such surrounding nodes, not themselves contenders for a plant location, which are closer to it than to some other potential site. This procedure can be adopted wherever the number of network nodes is greater than the number of feasible sites, but where this is not so some other procedure is required.

Some Alternative Partitioning and Perturbation Methods

The problem of node agglomeration can be approached by adopting the following procedure which will be known hereafter as the weight shift elimination method. The details for deriving $X_{\bar{p}}$ with this heuristic may be given as

i) For each node v_i on the network find some other node v_j for which $d(v_i\ v_j)$ is a minimum. Store all v_i in some array say $|AB|$, all v_j in $|AC|$ and all $d(v_i\ v_j)$ in array $|AD|$. Let some vector $|B|$ contain the list of node weights for each potential site b_i, $i=1,\cdots,m$.

ii) Scan array $|AD|$ and find some element, the rth element, for which $d(v_i\ v_j)\ b_i = w(v_i\ v_j)$ is a minimum. Designate the associated node in $|AB|$ as v_r and that in AC as v_s.

iii) Eliminate node v_r from $|AB|$. Rewrite arrays $|AB|$, $|AC|$ and $|AD|$ omitting the rth element in each and reduce n to n-1 which is the number of nodes now extant in $X_{\bar{p}}$.

iv) Let b_r be the weight of node v_r and b_s the weight of node v_s. Then change b_s so that $b_s = b_s + b_r$. In this step advantage is thus taken of the weight of the node v_r which is dropped on each iteration. Specifically its weight is added to the weight of that node v_s to which it was closest before elimination. Each time $d(v_i\ v_j)\ b_i$ is calculated as in step ii) the value of b_i in the computation may therefore be greater than the actual weight of node v_i on the network. As nodes are progressively eliminated the weight transfers are cumulative, the weight of each dropped node consisting of the sum of all weights previously assigned to it in turn by nodes which were dropped

on an earlier iteration. The net result of this cumulative weight shifting is to provide in effect a measure of node agglomeration.

v) In the final step array $|AC|$ is searched and for all nodes v_j in $|AC|$ for which $v_j = v_r$ calculate a new value $d(v_i\ v_j)\ v_j \neq v_r$. Replace the appropriate elements in $|AC|$ and $|AD|$ with the new values of v_j and $d(v_i\ v_j)$ respectively. In other words a new nearest neighbour v_j is found for all nodes v_i whose closest node was v_r before its elimination.

Steps ii) to v) are repeated until $X_{\bar{p}}$ contains p nodes. This is the candidate set of beginning nodes based upon this heuristic. If the three median problem of Figure 5 is reconsidered the set of $X_{\bar{p}}$ derived by this procedure is v_3, v_4 and v_9 which is clearly a better initial feasible solution than those given by the other methods.

For perturbing the lower bound solution the following heuristic is suggested. Drop in sequence from one to three of the current medians and replace them in turn with some other potential plant site not presently in the lower bound solution. From among these nodes v_i, $i=1,\cdots,m-p$ there exists one such member v_r for which

$$\max \sum_{i=1}^{p} \sum_{j \in G_i} w(v_j\ v_r) - w(v_j\ v_{ki}) \tag{16}$$

where v_j is a member of G_i such that $w(v_j\ v_r) < w(v_j\ v_{ki})$. This ensures the choice of that node for which the sum of the differences of the weighted distances of all nodes which are nearer to it than to their assigned median, is a maximum.

This node is then substituted in turn for the following three medians, assuming a different median in each case. First drop the median of that G_i for which S_i is a minimum before (16) is executed. The second candidate median for elimination belongs to that subgraph whose S_i value is at a minimum after completion of (16). Finally the third candidate median is the one for which $w(v_{ki}\ v_{kj})$ is least. This is the median whose weighted distance to some other median is a minimum over the network.

An Algorithm for Estimating the Medians of a Weighted Graph

This algorithm utilizes from one to four methods (at the user's discretion) for obtaining the initial partitioning of the graph. The best descendant solution from any one of these methods is retained as the estimated minimum bound. The first partitioning heuristic makes use of the Singer criterion as previously described. It is worth noting that since a set of $X_{\bar{p}}$ is derived for each row, for large scale problems with many hundreds of nodes computation can become exacting. For such cases an alternative procedure is to obtain just one candidate set which is

resolved from branching on that pair of nodes for which a_{ij} is a maximum over all such elements in $|A|$. A second partitioning heuristic consists of adding nodes one at a time according to the maxmin principle such that v_r, the next node added, is that one for which the minimum of $w(v_r X_{\bar{p}})$ is a maximum over any such v_i, where $X_{\bar{p}}$ contains nodes previously assigned to the candidate set and $v_i \notin X_{\bar{p}}$. This is really a sort of second-hand version of the Singer criterion, its main benefit being to provide a quick alternative set of $X_{\bar{p}}$ for the larger scale problem. The third partitioning heuristic is the weight shift elimination method, and the fourth is similar to the third but with step iv) omitted, and is therefore tantamount to a weighted version of Cooper's elimination procedure.

Any one or all of these partitioning heuristics may be used depending on the size of the network and the amount of computing time available. The weight shift elimination method gives, in the author's experience, the best initial starting solution. The short version of the Singer criterion can be recommended for the large network problem, since the derivation of $X_{\bar{p}}$ based on this heuristic is computationally faster than for any of the other alternative methods.

With the presentation of partitioning and perturbation methods now complete a formal description of the basic algorithm may now be given.

Step 1). Input the data consisting of a short path cost matrix and a vector of node demands, the b_i values. Appendix 3 gives a Fortran IV computer program (the Dantzig algorithm) for calculating all the shortest paths through a network. The minimum lower bound value S is initialized with an arbitrarily high number, and some index, say I, is set equal to zero.

Step 2). Partition the graph according to any or all of the heuristics just given. If, for methods one and two, more than one initial set of $X_{\bar{p}}$ is found, only the first two are retained. The user may wish to supply, in addition, his own set of beginning weights and/or an initial guess at the optimal solution. Let J = the total number of $X_{\bar{p}}$ sets.

Step 3). Increment I by one. Consider the Ith set of $X_{\bar{p}}$ nodes and let these be designated as the initial p medians of the network.

Step 4). Assign nodes to the current set of medians by equation (10), find a new set of medians as in (12) and an objective function ϕ as in (13). Repeat this step for all combination of medians in the current list, and for a limited number of combinations of shared nodes if any, (i.e. nodes equidistant to medians). Store the current best solution.

Step 5). Compare ϕ and S. If $\phi < S$, let $S = \phi$ and go to step 4. If $\phi = S$ print the results for all combinations if any, and go to step 6.

Step 6). Drop one of the specified current best medians and replace it with a substitute as previously defined. Go to step 4. Repeat this step dropping the other two designated medians if different from the first and from each other. Go to step 7 after evaluation of all substitutes.

Step 7). If I = J the algorithm has completed its search for condition (11) and terminates. Otherwise proceed to step 3.

Restrictions on Potential Plant Sites

In some specifications of the plant location problem not every node need be considered as a potential plant site. Many locations because of their remoteness or lack of amenities can be safely excluded from consideration. The required set of p nodes is therefore to be chosen from among some subset m of all n nodes. Computationally this restriction is desirable since it has the advantage of both allowing larger networks to be handled and of considerably speeding execution times particularly with respect to step 2 in the main algorithm.

In other variations of the problem some expanding operation may be faced with the task of adding new facilities to an existing distribution system where k plants are already in operation. Conceptually p-k new plants are needed from among the m potential sites where $m \leq n$.

The heuristics for handling these situations require first that the search for the set(s) of $X_{\bar{p}}$ be constrained

PROBLEM NAME ..... FIGURE 9 2 MEDIAN SOLUTION (LIMITED POTENTIAL SITES) FOR FIGURE 4

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	4	14.00	30.00	0.00	30.00	1 2 4 5 7	3.0 1.0 3.0 2.0 5.0
2	6	10.50	14.00	0.00	14.00	3 6 8 9 10	0.5 4.0 1.0 3.0 2.0
NETWORK		24.50	44.00	0.00	44.00		

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	4	9.00	15.00	0.00	15.00	1 2 4 5	3.0 1.0 3.0 2.0
2	6	15.50	29.00	0.00	29.00	3 6 7 8 9	0.5 4.0 5.0 1.0 3.0
						10	2.0
NETWORK		24.50	44.00	0.00	44.00		

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	4	12.00	26.00	0.00	26.00	1 2 4 7	3.0 1.0 3.0 5.0
2	6	12.50	18.00	0.00	18.00	3 5 6 8 9	0.5 2.0 4.0 1.0 3.0
						10	2.0
NETWORK		24.50	44.00	0.00	44.00		

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	4	7.00	11.00	0.00	11.00	1 2 4	3.0 1.0 3.0
2	6	17.50	33.00	0.00	33.00	3 5 6 7 8	0.5 2.0 4.0 5.0 1.0
						9 10	3.0 2.0
NETWORK		24.50	44.00	0.00	44.00		

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	4	11.00	21.00	0.00	21.00	2 4 5 7	1.0 3.0 2.0 5.0
2	6	13.50	23.00	0.00	23.00	1 3 6 8 9	3.0 0.5 4.0 1.0 3.0
						10	2.0
NETWORK		24.50	44.00	0.00	44.00		

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	4	6.00	6.00	0.00	6.00	2 4 5	1.0 3.0 2.0
2	6	18.50	38.00	0.00	38.00	1 3 6 7 8	3.0 0.5 4.0 5.0 1.0
						9 10	3.0 2.0
NETWORK		24.50	44.00	0.00	44.00		

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	4	9.00	17.00	0.00	17.00	2 4 7	1.0 3.0 5.0
2	6	15.50	27.00	0.00	27.00	1 3 5 6 8	3.0 0.5 2.0 4.0 1.0
						9 10	3.0 2.0
NETWORK		24.50	44.00	0.00	44.00		

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	4	4.00	2.00	0.00	2.00	2 4	1.0 3.0
2	7	20.50	40.50	0.00	40.50	1 3 5 6 7	3.0 0.5 2.0 4.0 5.0
						8 9 10	1.0 3.0 2.0
NETWORK		24.50	42.50	0.00	42.50		

Figure 9

to those nodes designated in the subset m. If $k > 0$ then k nodes are forced into the solution, and the search for the remaining p-k nodes is conducted among that subset m-k, where the k designated nodes are some of those originally specified in subset m. This restriction will ensure that each node labelled k (where $k > 0$) will be assigned to a separate subgraph, and that the remaining p-k subgraphs will contain at least one member of subset m-k. The n nodes or markets are then assigned by (10), and the choice of median for each G_i is constrained to nodes v_k in those subgraphs where such a k labelled node is present, and from among such m-k members as are present in each subgraph where no v_k node exists.

These concepts may be illustrated with reference to Figure 4. Let m, the subset of potential plant sites, contain members v_1 v_4 v_6 v_7 v_9 and v_{10} with the added restriction that $v_4 = v_k$, the designated forced node. Then, for the two median of this graph $X_{\bar{p}}$ contains v_4 and v_9 by the first heuristic and v_4 and v_7 by the third, and the final solution is shown in Figure 9.

Plant Size Upper Bounds

Zangwill (1968) has shown that under conditions of unlimited plant size each destination will, in the optimal solution, be supplied by one and only one source. In practice however, it may happen that scale diseconomies in plant production costs or restrictions on the availability of

labour or other inputs fundamental to the production process necessitate the imposition of upper bounds on the potential capacity of some prospective location site. Such plant site constraints can imply that in the global minimum one or more markets may be supplied from more than one source.

The capacity restricted plant location problem has received little previous attention. Efroymson and Ray ignore this constrained situation while Rutenberg's approach is valid only for approximating plant size upper bounds. Among the numerical solutions only Cooper (1961) has considered this restriction and his heursitics will be examined here in some detail. Cooper's algorithm covers some four basic steps as follows:

i) The problem is solved first for the unrestricted case. Associated with each of the resulting medians is a subset of demand which may exceed the median plant size upper bound (capacity deficit) or may be less (capacity surplus).

ii) Next, in Cooper's words "let I- be any set with a capacity deficit and let I+ be any set with a capacity surplus. Choose a destination point v_r in the set I- such that the difference in the distances from v_r to the source of I- and from v_r to the source of I+ is a minimum."

iii) As much as possible of the demand of v_r is now re-allocated from I- to I+ and a new set of medians is computed for each subgraph.

iv) Each subgraph is next examined for any member of the set which may be closer to some other median with a capacity surplus, and if this condition holds, further re-allocation occurs. Steps i) and iv) are repeated until all deficits have been eliminated.

Notwithstanding the good results seemingly obtained by Cooper with this method two disadvantages of the formulation are readily apparent. The first of these is simply that there seems to be no guarantee that the medians of the unconstrained solution can together, satisfy the total network demand. Neither is there any certainty that each subgraph contains at least one potential median which will ensure this occurrence. If the upper bound constraints are sufficiently restrictive it is not impossible for the only p nodes capable of satisfying total node requirements, to be clustered together within the same subgraph upon termination of the unconstrained solution.

The second disadvantage is a direct consequence of the fact that step iv) in Cooper's algorithm only re-allocates node demands to medians with excess capacity. Under these circumstances the arbitrary order in which the I- and I+ sources are apparently chosen can be of considerable significance. Consider the network given in Figure 6 for which a 4 median solution is required. Let the plant size upper bound for each of the 10 plants be in sequence 8, 13, 10, 12, 11, 11, 7, 15, 8 and 9 units.

Table 3

POSSIBLE SOLUTIONS TO THE 4 MEDIAN PROBLEM OF FIGURE 6 BY COOPER'S HEURISTICS

a) Unconstrained Solution

Median	Associated Node	Node Volumes	Plant Size	Upper Bound	I-	I+	Cost
V_2	V_1, V_2, V_3	1,10,4	15	13	-2		21
V_4	V_4	10	10	12		+2	0
V_6	V_5, V_6, V_7	2,10,1	13	11	-2		7
V_8	V_8, V_9, V_{10}	10,1,1	12	14		+2	2
		TOTAL	50	50	-4	+4	30

b) Plant Size Constrained Solution with V_4 and V_6 as the first I+ and I-1 sources respectively.

V_2	V_1, V_2, V_3	1,10,2	13	13		11
V_4	V_4, V_5	10,2	12	12		10
V_6	V_6, V_7	10,1	11	11		1
V_8	V_3, V_8, V_9, V_{10}	2,10,1,1	14	14		40
		TOTAL	50	50		62

In the unconstrained solution (Table 3) medians v_2 and v_6 have capacity deficits and medians v_4 and v_8 capacity surpluses. If the I- source chosen is v_6 and the I+ source is v_4 then, by the definition of step ii), node v_5 is switched from v_6 to v_4 so that the capacity constraints are explicitly satisfied at both of these plants. Median v_2 still exceeds its upper bound so that on the next iteration two units from node v_3 are switched to median v_8. Since each upper bound is exactly satisfied no further re-allocation occurs and the algorithm terminates with a solution of S = 62·0. This is not however the global minimum, for if median v_2 is chosen as the first I- source instead of v_6 then the optimal bound reduces to S = 50·0 (Figure 10a).

Two new sets of heuristics may now be presented which seem to offer consistently better results. In the first of these, the constrained solution approach, the capacity restrictions are satisfied at each step so that a feasible solution is ensured at every stage in the computation. Let the upper bound imposed on each potential plant site be labelled c_i, $i = 1,\cdots,m$. Then in any feasible solution, the restriction $t_i \leq c_i$ $i = 1,\cdots,p$ must apply where t_i is the current plant size for median v_{ki}. Now proceed in the following steps.

i) First ensure that for a given value of p there are p plants whose combined upper bounds are at least

equal to

$$\theta = \sum_{j=1}^{n} b_j ,$$

the total demand on the network. Specifically find the p potential plant sites whose c_i values are a maximum. Test whether the summation of these c_i values $\geq \theta$. If so, then a feasible solution exists.

ii) Among the m plant sites find a lower c_i value ϕ such that any p combination of c_i values exceeding ϕ will, when summed, be $\geq \theta$. Of course there may (will probably) exist combinations of p nodes some of whose individual c_i values are less than ϕ which are also feasible with respect to θ. The computation of ϕ will act as an insurance policy to keep the solution feasible, if necessary, at some stages in the ensuing computation.

iii) Break open the network according to any of the partitioning heuristics previously given. Retain this set of $X_{\bar{p}}$ only if the sum of the corresponding c_i values exceeds θ. Otherwise repeat the step restricting the choice of $X_{\bar{p}}$ from among only those nodes whose c_i values are $\geq \phi$. This procedure ensures the choice of an initial set of medians capable of satisfying the total network requirements.

iv) Define some vector say z, which contains the current amount of unfilled storage available at each median in solution. Initially the z values will be set equal to c_i, $i = 1,\cdots,p$.

v) Now allocate the demand of individual nodes to some one or more of the current median sets. The following procedure will provide a good basic start. Define for each such node v_i, $i = 1,\cdots,n$ some value x such that $x_i = d(v_i\ v_{k1}) - d(v_i\ v_{k2})$ where v_{k1} and v_{k2} are the closest two medians for that v_i respectively. Store all such x_i values and rank them in descending order of magnitude. The associated labelled nodes (corresponding to the x_i values) will now be assigned in this order, each one, or as much as possible of the demand at each one, to its particular v_{k1} median. After each assignment the value of the appropriate z element is reduced according to the amount allocated. If z = 0 for any median the upper bound constraint has been explicitly satisfied and no more destination requirements can be supplied from that source. The array x acts in effect as a penalty vector such that the node first assigned is always that one whose non-allocation to its designated nearest median might prove, should z become zero, to be most expensive in terms of transportation costs. Whenever some z value is reduced to zero then this step is repeated for all nodes whose demand is not as yet totally assigned, omitting from consideration the associated median and all others whose available capacity has been totally filled by previous allocations.

vi) This step reconsiders the allocations defined in the previous stage and makes readjustments where necessary. Consider each subgraph G_i in turn and search for some node $v_r \varepsilon G_i$, which is nearer to the median of some other subgraph G_j. Symbolically if there exists some v_r such that

$$d(v_r\ v_{ki}) > d(v_r\ v_{kj})$$

then let

$$y_1 = d(v_r\ v_{ki}) - d(v_r\ v_{kj})$$

where y 1 is the potential saving associated with the transfer of one unit of v_r from G_i to G_j. Now examine the subset of nodes $\varepsilon\ G_j$ for some node v_ℓ which is closer to v_{ki} than is v_r. If $d(v_\ell\ v_{ki}) < d(v_r\ v_{ki})$ then calculate

$$y_2 = d(v_\ell\ v_{ki}) - d(v_\ell\ v_{kj})$$

where y_2 is the potential loss (in terms of v_ℓ) of transferring back one unit of v_ℓ from G_j to G_i. Now compute $y = y_1 - y_2$ and if $y > 0$ then the savings to be made from re-allocating v_r (or that portion of the demand of v_r in G_i) from G_i to G_j more than offset the losses of returning a like amount of the demand of v_ℓ from G_j to G_i. The actual amount re-allocated will clearly depend upon the relative sizes of the demands at v_r and v_ℓ. If b_r = the demand of v_r in G_i and b_ℓ = the demand of v_ℓ in G_j then when $b_r \leq b_\ell$ an amount equal to b_r is switched from

G_i to G_j. Where $b_r > b_\ell$ however, the amount transferred is equal only to b_ℓ. In each case of course an equal amount is returned from G_j to G_i (by route v_ℓ to v_{ki}) to maintain the equilibrium of supply and demand. This step is repeated until no further readjustments are possible, and the choice of v_r and v_ℓ on any cycle is determined by the maximum value of y.

The concepts discussed so far can be briefly illustrated with reference again to Figure 6. At termination of step iii) $X_{\bar{p}}$ consists of v_2 v_4 v_6 and v_8 whose plant sizes together satisfy θ, the total network demand. By the details of step v) the node order for assignment is v_1, v_2, v_8, v_9, v_{10}, v_4, v_6, v_7, v_5, v_3 since the corresponding values of x are, in descending order, 11, 11, 9, 9, 9, 4, 4, 4, 2, and 1. Node v_1 is thus assigned first since the difference in the distances to its nearest two medians (v_2 and v_4) is 11, which is a maximum, (though not unique), for any node on the network. The initial basic start given by step v) is shown in Table 4 (b) with $S \cong 62{\cdot}0$. In examining this tableau note that 2 units of v_3 are assigned to G_4. The medians of all other three subgraphs are actually closer to v_3 than is v_8, the median of G_4. Consider G_2 which contains nodes v_4 (the median) and v_5. To transfer one unit of v_3 from v_8 to v_4 will reduce the cost (distance) from 19 to 6 so that $y_1 = 13$. Transferring one unit of v_5 back to G_4 from G_2 increases

the cost of supplying that unit from $5(d(v_5\ v_4))$ to $12(d(v_5\ v_8))$ so that the loss $y_2 = 12 - 5 = 7$. But the overall gain associated with this re-allocation is positive, $y = 13 - 7 = 6$ for each unit, and since two units can be transferred the total transport cost (distance) can be reduced by 12. Since no further readjustments can be made this is the optimal allocation (Table 4c, Figure 10a).

Note that steps v) and vi) are in fact actually a restatement of the classical transportation problem because the total network demand θ can always be made exactly equivalent to the combined capacity of the medians in solution by the introduction of slack variables. The procedure of v) will always serve to provide a good initial basis, and although the method of step vi) for re-allocating flows will normally be less efficient than existing algorithms which employ the method of shadow costs, the problem of degeneracy does not appear to occur so that coding is considerably simplified.

vii) Let some value t_i be the total subgraph demand so that

$$\sum_{i=1}^{p} t_i = \theta$$

the overall network requirement. The medians of each subgraph are now re-calculated according to (12) subject to further restriction that only those potential sites $v_i \in G_i$ whose upper bounds c_i are $\geq t_i$ will be considered as candidates. This constraint ensures choice from among only

Table 4

SOLUTION TABLEAU FOR FIGURE 6

(a) Distance (Transportation Cost) Matrix

G		V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	Capacity	
1	V_2	1	0	5	11	16	15	16	24	25	25	13	
2	V_4	12	11	6	0	5	4	5	13	14	14	12	
3	V_6	16	15	10	4	3	0	1	9	10	10	11	
4	V_8	25	24	19	13	12	9	10	0	1	1	15	
Demand		1	10	4	10	2	10	1	10	1	1	50	

(b) Initial Allocation (Step v)

G		V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	Capacity	Cost
1	V_2	1	10	2								13	11
2	V_4				10	2						12	10
3	V_6						10	1				11	1
4	V_8			2					10	1	1	14	40
Demand		1	10	4	10	2	10	1	10	1	1	50	62

(c) Optimal Allocation (Step vi)

G		V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	Capacity	Cost
1	V_2	1	10	2								13	11
2	V_4			2	10							12	12
3	V_6						10	1				11	1
4	V_8					2			10	1	1	14	26
Demand		1	10	4	10	2	10	1	10	1	1	50	50

PROBLEM NAME ..... FIGURE 10(A) 4 MEDIAN SOLUTION (PLANT SIZE CONSTRAINTS) FOR FIGURE 6

****

PLANT LOCATION-ALLOCATION 4 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	2	13.00	11.00	0.00	11.00	1 2 3	1.0 10.0 2.0
2	4	12.00	12.00	0.00	12.00	3 4	2.0 10.0
3	6	11.00	1.00	0.00	1.00	6 7	10.0 1.0
4	8	14.00	26.00	0.00	26.00	5 8 9 10	2.0 10.0 1.0 1.0
	NETWORK	50.00	50.00	0.00	50.00		

****

PROBLEM NAME ..... FIGURE 10(B) 2 AND 3 MEDIAN SOLUTIONS (PLANT SIZE CONSTRAINTS) FOR FIGURE 4

CAPACITY CONSTRAINT CRITERION PREVENTS A 2 LOCATION SOLUTION

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	5	9.00	10.00	0.00	10.00	1 4 5 7	3.0 3.0 2.0 1.0
2	8	8.00	10.00	0.00	10.00	6 7 8	4.0 3.0 1.0
3	10	7.50	13.50	0.00	13.50	2 3 7 9 10	1.0 0.5 1.0 3.0 2.0
	NETWORK	24.50	33.50	0.00	33.50		

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	1	7.50	11.00	0.00	11.00	1 2 7	3.0 1.0 3.5
2	5	9.00	12.50	0.00	12.50	4 5 6 7	3.0 2.0 2.5 1.5
3	8	8.00	9.00	0.00	9.00	3 6 8 9 10	0.5 1.5 1.0 3.0 2.0
	NETWORK	24.50	32.50	0.00	32.50		

****

PROBLEM NAME ..... FIGURE 10(C) 2 MEDIAN SOLUTION (MAXIMUM DISTANCE CONSTRAINT) FOR FIGURE 4

****

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	1	14.00	25.00	0.00	25.00	1 2 4 5 7	3.0 1.0 3.0 2.0 5.0
2	6	10.50	14.00	0.00	14.00	3 6 8 9 10	0.5 4.0 1.0 3.0 2.0
	NETWORK	24.50	39.00	0.00	39.00		

****

NO SOLUTION FOUND ON THIS ITERATION SATISFYING MAXIMUM DISTANCE CONSTRAINT FOR A 2 LOCATION SOLUTION

Figure 10

those nodes capable of satisfying the subgraph's demand. Since from the previous steps there is at least one such member in each G_i the solution will always remain feasible.

viii) Steps iv) to vii) are reiterated until one full cycle fails to reduce the objective function S.

In the second set of heuristics, the unconstrained solution approach, the objective is to solve first the minimum transport cost solution without plant size constraints, but restricting choice of medians to some set of p, known a priori, to be capable together of satisfying θ. This solution is then subsequently modified to satisfy the plant upper bound restrictions. These goals are simply achieved as follows:

1) In partitioning the network choice of nodes is exclusively restricted to those potential sites whose c_i values exceed ϕ.

2) Each node on the network v_i, $i = 1,\cdots,n$ is assigned by (10) to its closest member of $X_{\bar{p}}$. Upper plant bound violations are ignored for the moment.

3) The search for the median of each subgraph is again restricted this time to those members $v_j \in G_i$ for which $c_j \geq \phi$. From the previous restriction on step 1) there must be at least one such node in each subset.

4) Steps 2 and 3 are repeated and the lower bound test applied until a stable solution is attained. (N.B. In each of the two sets of heuristics nodes are only

considered for median replacement during the lower bound test, if and only if, it is known that they can guarantee the continued feasibility of the solution).

5) Using this local minimum (which may or may not be optimal) as a new starting basis steps iv) to viii) of the previous heuristic set are activated until no further improvement is found. This solution chosen is, of course, the best descendent one from any of the two given methods if both are optioned.

It is possible to solve some highly constrained problems with these sets of heuristics. Consider again the graph of Figure 4 in which, for this example, the m potential plant sites are restricted to v_1, v_3, v_5, v_6, v_8 and v_{10} and, as a further proviso, node v_8 is to be forced into the solution. Let the upper bounds on each of the m sites be set equal to 7, 7, 9, 5, 8 and 7·5 respectively. For the 3 median of this network the beginning set $X_{\bar{p}}$ consists of either v_1, v_5, v_6 or v_3, v_5, v_6. Each of the two heuristic sets converges to a local minima of S = 33·5 with nodes v_5, v_8 and v_{10} as medians (Figure 10b). By the lower bound test v_{10} is dropped from solution and node v_1 is introduced as the new candidate median. The solution attained on termination is v_1, v_5 and v_8 with S = 32·5 (Figure 10b) which appears to be the optimal lower bound. It seems fair to say that this result is not obvious from inspection.

Maximum Distance Constraint

For this option the heuristics presented may be unsuccessful since solutions can sometimes exist which are not found during computation. The suggested method is simply to constrain the search for the median of each subgraph G_i, $i=1,\cdots,p$ to those nodes $v_r \varepsilon G_i$ for which $d(v_j v_r) \leq V_j$ for all $v_j \varepsilon G_i$, where V_j is the maximum allowable distance of v_j to some median on the network. For any potential median v_r not satisfying this condition then there exists at least one other node v_j whose distance to v_r is greater than the stipulated maximum. This amount to satisfying equation (12) subject if possible to

(17) $$d(v_j \; v_{ki}) \leq V_j \quad \text{for all} \quad v_j \; \varepsilon \; G_i$$

As an example reconsider Figure 4 and place the following maximum distance constraints on each node 0·5, 4, 3, 7, 2, 0·5, 3, 4, 8 and 5. Subject to these constraints the 2 median solution consists of v_1 and v_6, which are in fact, the only two vertices capable of satisfying condition (17). (Figure 10c). In the event, on any iteration, that p medians satisfying the constraint cannot be found, the least cost transport solution is chosen instead for the next cycle, and a message is printed to this effect. In extreme cases the algorithm may therefore terminate without printing a feasible solution even though one exists. One partial way out of this impasse might be to choose, for all subgraphs where the constraint cannot be currently satisfied, that node $v_j \varepsilon G_i$ whose value is a

minimum in the set, and continue from there. However even this scheme will not guarantee success and is not as yet incorporated into the computer code for this algorithm (see Appendix 5). It is possible, even desirable, to force into the solution those nodes whose distance constraints are very restrictive, so that the liklihood of not obtaining a feasible solution is sharply reduced. For many networks this procedure will provide the best alternative, since there seems to be no obvious way of ensuring even an initial set of $X_{\bar{p}}$ which is feasible. In other cases it may well be that the problem is soluble only if structured as a linear program.

Plant Costs

For the plant location-allocation problem which requires the simultaneous minimization of both plant and transport costs, the value of p for which this is a minimum will not usually be known at the outset. In the method given here the general strategy will be to calculate the total cost function for several fixed arbitrary values of p, choose the two solutions within which the optimum must lie, and beginning with the lower value of p, increment it sequentially by one until the p+1 solution exceeds the pth solution.

As with the plant size constraint options two sets of heruistics will be given one of which will yield a better

estimate of the true lower bound. In the first of these two methods the procedure is as follows:

i) Solve first the minimum transportation cost solution. Determination of this p median will automatically define the least cost allocation of markets to sources, so that the plant size t_i of each source is also given. From this the associated plant costs can be readily established. This is the approach taken by Singer but his algorithm terminates at this point presumably under the assumption that the minimum transportation cost solution will also define, for any p, the total least cost bound. While this may generally be true it is not difficult to visualize situations in which scale economies at some plants may be more than sufficient to offset a non-optimal transportation cost solution.

ii) This step differs from the previous one only in terms of the median calculation. Specifically, choice of median on each subsequent iteration is now determined by an appraisal of both plant and transportation costs together, rather than on the latter basis alone. Thus for each G_i, S_i now equals

(18) $\sum_{j \varepsilon G_i} w(v_j \; v_{ki}) + f_i(t_i)$ where

f_i is some function of t_i, the current source size, such that $f_i(t_i)$ is the plant cost for that source at its present level of operation. In practice this function must be supplied for each potential site by the user as a subroutine.

iii) After repeating step ii) to achieve convergence there still remains the possibility that further advantage of the plant economies of scale can be gained by judicious manipulations of the current vector of allocations. A test is therefore made in which markets (or fractions of markets if size constraints dictate) are flipped one by one from each source currently assigned to it to those which are not. Let y_1 be the decrease in total costs (over some G_i) achieved by removing some market v_r from median v_{ki}. Similarly let y_2 be the increase in total costs (over some G_j) effected by the addition of v_r to some v_{kj}. If for any median v_{kj}, $y_2 < y_1$ then some reduction in total cost (over the network) is feasible. Choice of v_r and v_{kj} is dependent upon the maximum difference $y_1 - y_2$ (if any) found. This step is repeated until no further improvement is found, subject to the maintenance of at least one unit of demand at each source.

The second set of plant cost heuristics differs from the first in that nodes are assigned to medians on the basis of at least a partial evaluation of plant (in addition to transportation) costs. Consistent with the approach taken throughout this chapter the suggested procedure will be outlined in steps.

1) For each node v_j, $j=1,\cdots,n$ find the current median v_{ki}, $i=1,\cdots,p$ for which $d(v_j\ v_{ki})$ is a minimum.

2) Each subgraph G_i, $i = 1,\cdots,p$ now has a subset of nodes $v_j \varepsilon G_i$ which satisfy Step 1). Rank each member of each subset in ascending order in terms of $d(v_j\ v_{ki})$.

3) Assign first that member of G_1 which is closest to v_{k1}. Secondly assign to v_{k2} that member of G_2 which is closest to v_{k2} and so on until p such assignments have been made. This operation ensures the allocation of at least one destination to each source and yields a beginning plant size for each median.

4) Tentatively assign to v_{k1} the member in the set of G_1 which is second closest to v_{k1}. But before this allocation is confirmed the test given in step iii) of the previous heuristic set is made. Repeat this procedure for all remaining node assignments, the choice of order being determined by the ranking of the nodes in each subset. Each set of p assignments will then, take one member from each subset, but the test is activated only for the second and subsequent series of allocations. On any one series of such assignments after the initial allocations have been made it is feasible that some subsets may become exhausted of members. On termination all sets will of course be empty.

5) The medians of each G_i are now computed according to the total cost criterion of step ii), and steps 1 to 5 are again repeated until the lowest bound is found.

Applying these heuristics again to Figure 4 with plant costs set equal on any iteration to $10\sqrt{t_i}$ (t_i = current size of plant i) at nodes v_1, v_4, v_6, v_9 and v_{10} and to $7\sqrt{t_i}$ for the other five potential sources, the lowest calculated bound of S = 86·8 occurs when p = 3, with medians v_1, v_4, v_6 or v_1, v_4, v_9 in solution (Figure 11a). With the same plant cost functions but size upper bounds of 4, 6, 7, 10, 12, 11, 8, 13, 9 and 10 respectively imposed upon each node, the three median solution of the graph consists of v_4, v_5 and v_6 with S = 92·3 (Figure 11b).

Dynamic Location-Allocation Systems

The heuristics for systems of static location-allocation problems given previously in this chapter are readily adaptable to the dynamic sphere of operations. This extension demands the sequential partitioning of a network over discrete intervals of time. Redefining t_j as some time period j = 1, 2,···,r, then given some long term planning horizon t_r, and a budget constraint which allows for the construction of one or more facilities in any time interval t_j, the problem is to decide in which locations and in what order the plants should be built such that the overall costs (aggregate distances) are at a minimum over all periods under question.

This is a combinatorial programming problem which can always be explicitly solved by direct enumeration of

PROBLEM NAME ..... FIGURE 11(A) MINIMUM COMBINED PLANT-TRANSPORTATION SOLUTION FOR FIGURE 4

PLANT LOCATION-ALLOCATION 1 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	6	24.50	59.00	34.65	93.65	1	2	3	4	5	3.0	1.0	0.5	3.0	2.0
						6	7	8	9	10	4.0	5.0	1.0	3.0	2.0
NETWORK		24.50	59.00	34.65	93.65										

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	1	14.00	25.00	26.19	51.19	1	2	4	5	7	3.0	1.0	3.0	2.0	5.0
2	6	10.50	14.00	22.68	36.68	3	6	8	9	10	0.5	4.0	1.0	3.0	2.0
NETWORK		24.50	39.00	48.87	87.87										

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	1	14.00	25.00	26.19	51.19	1	2	4	5	7	3.0	1.0	3.0	2.0	5.0
2	9	10.50	14.00	22.68	36.68	3	6	8	9	10	0.5	4.0	1.0	3.0	2.0
NETWORK		24.50	39.00	48.87	87.87										

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	1	10.00	12.00	22.14	34.14	1	5	7			3.0	2.0	5.0		
2	4	4.00	2.00	14.00	16.00	2	4				1.0	3.0			
3	6	10.50	14.00	22.68	36.68	3	6	8	9	10	0.5	4.0	1.0	3.0	2.0
NETWORK		24.50	28.00	58.82	86.82										

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	1	10.00	12.00	22.14	34.14	1	5	7			3.0	2.0	5.0		
2	4	4.00	2.00	14.00	16.00	2	4				1.0	3.0			
3	9	10.50	14.00	22.68	36.68	3	6	8	9	10	0.5	4.0	1.0	3.0	2.0
NETWORK		24.50	28.00	58.82	86.82										

PROBLEM NAME ..... FIGURE 11(B) 3 MEDIAN SOLUTION (PLANT SIZE CONSTRAINTS) FOR FIGURE 4

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	4	6.00	6.00	17.15	23.15	2	4	5			1.0	3.0	2.0		
2	6	10.50	14.00	22.68	36.68	3	6	8	9	10	0.5	4.0	1.0	3.0	2.0
3	7	8.00	6.00	28.28	34.28	1	7				3.0	5.0			
NETWORK		24.50	26.00	68.11	94.11										

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	4	4.00	2.00	14.00	16.00	2	4				1.0	3.0			
2	5	10.00	8.00	31.62	39.62	1	5	7			3.0	2.0	5.0		
3	6	10.50	14.00	22.68	36.68	3	6	8	9	10	0.5	4.0	1.0	3.0	2.0
NETWORK		24.50	24.00	68.31	92.31										

Figure 11

all possibilities. In practice this is normally quite infeasible, and consistent with the heuristic approach taken here, far fewer evaluations will be needed. With respect to the particular formulation under consideration some additional assumptions will be made. First the network of all n destinations will be assumed to be unchanged over all time periods in question. Next plants (medians) will be added one at a time during each temporal interval, and thirdly once chosen, each such source will be assumed to remain fixed at its original location over all succeeding periods. Actually the first two assumptions can be relaxed with only minor modifications to the existing program, but the third remains more restrictive reflecting as it does the question of relocation costs and the troublesome problems of locational obsolescence.

Of the two possible approaches to this problem the first makes use of a forward recursive procedure in which, beginning with time period t_1, a single median is located which is optimal with respect to all weighted distances over the network. A second facility is added next in t_2 which minimizes the sum of the aggregate flows subject to the maintenance of plant 1 at its previously located site. For any time period t+1 then, the next source to be added is that one which minimizes the function

$$(19) \qquad S = \sum_{i=1}^{n} \sum_{j=1}^{t} X_{ij} D_{ij} + \sum_{i=1}^{n} X_{it+1} D_{it+1}$$

where t labels the t plants established in the preceeding time periods 1 through t_j and X_{ij} and D_{ij} are as previously defined.

Consider this approach in terms of the twenty node network of Figure 7 where the long term planning goal t_r=3. The single median of this weighted graph is period t_1 is either v_8 or v_9 with S = 184·0. Branching first on median v_8, node v_6 is added in t_2, and either v_{11} or v_{13} in t_3 with S = 68·0. The cumulated costs over all three periods total to 360·0. Alternative branching on v_9 when t=1 adds v_{17} in the second period and v_6 in the third with S = 81·0 and cumulated costs equal to 380·0 (see Figures 12 and 13).

Although this approach is always optimal with respect to period t_1, on subsequent iterations over periods $t_2, \cdots, r$, the probability is very high that it will become increasingly sub-optimal (Teitz, 1968). To guarantee the true global minimum it is necessary to resort to discrete dynamic programming methods where the recursion procedure will be backwards rather than forwards in time. The adoption of a backwards recursion depends upon the principle that a dynamically optimal system should always, at the fulfillment of the long term planning horizon, have a

PROBLEM NAME ..... FIGURE 12 FORWARD RECURSIVE DYNAMIC SOLUTION FOR FIGURE 7

****

PLANT LOCATION-ALLOCATION 1 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	9	50.00	184.00	0.00	184.00	1	2	3	4	5	3.0	1.0	4.0	2.0	3.0
						6	7	8	9	10	4.0	4.0	3.0	1.0	2.0
						11	12	13	14	15	2.0	1.0	2.0	3.0	3.0
						16	17	18	19	20	1.0	4.0	2.0	3.0	2.0
NETWORK		50.00	184.00	0.00	184.00										

****

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	9	28.00	85.00	0.00	85.00	1	2	3	4	5	3.0	1.0	4.0	2.0	3.0
						6	9	10	11	12	4.0	1.0	2.0	2.0	1.0
						13	14				2.0	3.0			
2	17	22.00	30.00	0.00	30.00	7	8	15	16	17	4.0	3.0	3.0	1.0	4.0
						18	19	20			2.0	3.0	2.0		
NETWORK		50.00	115.00	0.00	115.00										

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	9	28.00	85.00	0.00	85.00	1	2	3	4	5	3.0	1.0	4.0	2.0	3.0
						6	9	10	11	12	4.0	1.0	2.0	2.0	1.0
						13	14				2.0	3.0			
2	17	17.00	18.00	0.00	18.00	7	8	15	16	17	4.0	3.0	3.0	1.0	4.0
						18					2.0				
3	19	5.00	2.00	0.00	2.00	19	20				3.0	2.0			
NETWORK		50.00	105.00	0.00	105.00										

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	9	28.00	85.00	0.00	85.00	1	2	3	4	5	3.0	1.0	4.0	2.0	3.0
						6	9	10	11	12	4.0	1.0	2.0	2.0	1.0
						13	14				2.0	3.0			
2	17	14.00	15.00	0.00	15.00	7	8	16	17	18	4.0	3.0	1.0	4.0	2.0
3	19	8.00	5.00	0.00	5.00	15	19	20			3.0	3.0	2.0		
NETWORK		50.00	105.00	0.00	105.00										

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN		PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED					VOLUME SUPPLIED TO EACH MEDIAN				
1	6	17.00	20.00	0.00	20.00	1	2	3	4	5	3.0	1.0	4.0	2.0	3.0
						6					4.0				
2	9	11.00	31.00	0.00	31.00	9	10	11	12	13	1.0	2.0	2.0	1.0	2.0
						14					3.0				
3	17	22.00	30.00	0.00	30.00	7	8	15	16	17	4.0	3.0	3.0	1.0	4.0
						18	19	20			2.0	3.0	2.0		
NETWORK		50.00	81.00	0.00	81.00										

****

Figure 12

PROBLEM NAME ..... FIGURE 13 FORWARD RECURSIVE DYNAMIC SOLUTION (NODE 9 FORCED) FOR FIGURE 7

****

PLANT LOCATION-ALLOCATION 1 MEDIAN SOLUTION

MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1 8	50.00	184.00	0.00	184.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
					6 7 8 9 10	4.0 4.0 3.0 1.0 2.0
					11 12 13 14 15	2.0 1.0 2.0 3.0 3.0
					16 17 18 19 20	1.0 4.0 2.0 3.0 2.0
NETWORK	50.00	184.00	0.00	184.00		

****

PLANT LOCATION-ALLOCATION 1 MEDIAN SOLUTION

MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1 9	50.00	184.00	0.00	184.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
					6 7 8 9 10	4.0 4.0 3.0 1.0 2.0
					11 12 13 14 15	2.0 1.0 2.0 3.0 3.0
					16 17 18 19 20	1.0 4.0 2.0 3.0 2.0
NETWORK	50.00	184.00	0.00	184.00		

****

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1 6	28.00	73.00	0.00	73.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
					6 9 10 11 12	4.0 1.0 2.0 2.0 1.0
					13 14	2.0 3.0
2 8	22.00	35.00	0.00	35.00	7 8 15 16 17	4.0 3.0 3.0 1.0 4.0
					18 19 20	2.0 3.0 2.0
NETWORK	50.00	108.00	0.00	108.00		

****

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1 6	25.00	55.00	0.00	55.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
					6 9 10 11 12	4.0 1.0 2.0 2.0 1.0
					13	2.0
2 8	25.00	53.00	0.00	53.00	7 8 14 15 16	4.0 3.0 3.0 3.0 1.0
					17 18 19 20	4.0 2.0 3.0 2.0
NETWORK	50.00	108.00	0.00	108.00		

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1 6	18.00	22.00	0.00	22.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
					6 9	4.0 1.0
2 8	22.00	35.00	0.00	35.00	7 8 15 16 17	4.0 3.0 3.0 1.0 4.0
					18 19 20	2.0 3.0 2.0
3 11	10.00	11.00	0.00	11.00	10 11 12 13 14	2.0 2.0 1.0 2.0 3.0
NETWORK	50.00	68.00	0.00	68.00		

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1 6	17.00	20.00	0.00	20.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
					6	4.0
2 8	22.00	35.00	0.00	35.00	7 8 15 16 17	4.0 3.0 3.0 1.0 4.0
					18 19 20	2.0 3.0 2.0
3 11	11.00	13.00	0.00	13.00	9 10 11 12 13	1.0 2.0 2.0 1.0 2.0
					14	3.0
NETWORK	50.00	68.00	0.00	68.00		

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1 6	18.00	22.00	0.00	22.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
					6 9	4.0 1.0
2 8	22.00	35.00	0.00	35.00	7 8 15 16 17	4.0 3.0 3.0 1.0 4.0
					18 19 20	2.0 3.0 2.0
3 13	10.00	11.00	0.00	11.00	10 11 12 13 14	2.0 2.0 1.0 2.0 3.0
NETWORK	50.00	68.00	0.00	68.00		

****

Figure 13

locational structure which is equivalent to that of a static system where $p = t_r$ the long range goal. Hence at the outset it is necessary to minimize the following function.

$$S_r = \sum_{i=1}^{n} \sum_{j=1}^{r} {}_{j\epsilon M} X_{ij} D_{ij} \tag{20}$$

where M is the set of all potential plant sites. This is the solution for the static case when p=r, and is tantamount to (1) without all the side constraints. Whether this solution is achieved by the heuristic methods given previously in this chapter or by some exact branch-bound algorithm, it now remains only to obtain an orderly sequencing of the plants given by (20) such that the long term cost accumulation is minimized.

It is this sequencing solution which is normally sought by dynamic programming, and the approach will usually be feasible up to a limit of about nine or ten time periods depending upon the size of the network. Beyond this, as will become apparent, dynamic programming tends towards infeasibility since computation increases very rapidly as a non-linear function of r. Thus at any stage t backwards from the solution of (20) there are $\binom{r}{t}$ or $r! / t!(r-t)!$ combinations of possible locations. Over all time periods t_j, $j = 1,\cdots,r$, there are then

(21) $$\sum_{t=1}^{r} \binom{r}{t}$$ such solutions to be evaluated

Clearly the amount of calculation involved will be a function of both the size of the network (since each node must be assigned to one of the t locations), and of the size of r which is the maximum number of periods involved. If r = 10 for example, then by (21) the number of locational combinations which must be evaluated is in excess of 1000. Moreover once these solutions have been obtained there still remain some r! possible ways to order them, one of which will be the optimal sequence required. For r = 10 the number of such possible sequences or permutations is over one million. Although dynamic programming will actually evalute for fewer sequences; by virtue of the computational effort implied in (21), r need never be very large before the problem in practice becomes hopelessly overextended. The presentation of a dynamic program for the optimal sequencing of plant locations over time will be deferred to the next section. The remainder of this one will be devoted to exploring heuristics for the very long term planning horizon problem when r is in excess of say nine or ten.

The general procedure is actually simple and will be to recurse backwards taking advantage of the solution of each preceeding step. Beginning with the solution to

(20) define R as the initial subset of t = r medians so obtained. Then, working backwards, and using the heuristics previously given for the estimation of the p median of a weighted graph solve for any subsequent time period t_i, t = r - 1, r - 2,···,1, the following expression

$$S_t = \sum_{i=1}^{n} \sum_{j=1}^{t}{}_{j \varepsilon R} X_{ij} D_{ij} \tag{22}$$

That is the solution is constrained to a search among the members (medians) of the set R which, at any stage, contains t+1 medians.

Although it is always possible to find counter-examples, this method will usually give satisfactory results. Referring back to the network of Figure 7, the results of the backward recursion can be seen in Figure 14. At completion of stage 1, t=3 and the minimum cost for three median solution of v_6, v_{11} and v_{17} is $S_3 = 63 \cdot 0$. During periods t_2 and t_3 medians v_{11} and v_{17} are dropped leaving v_6 with $S_1 = 60 \cdot 0$. The cumulated costs for all three periods are

$$S = \sum_{j=1}^{3} S_i = 350 \cdot 0 \tag{23}$$

which is less than either of the two results given by the forward recursive method.

PROBLEM NAME ..... FIGURE 14 BACKWARD RECURSIVE DYNAMIC SOLUTION FOR FIGURE 7

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	6	18.00	22.00	0.00	22.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
						6 9	4.0 1.0
2	11	10.00	11.00	0.00	11.00	10 11 12 13 14	2.0 2.0 1.0 2.0 3.0
3	17	22.00	30.00	0.00	30.00	7 8 15 16 17	4.0 3.0 3.0 1.0 4.0
						18 19 20	2.0 3.0 2.0
	NETWORK	50.00	63.00	0.00	63.00		

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	6	17.00	20.00	0.00	20.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
						6	4.0
2	11	11.00	13.00	0.00	13.00	9 10 11 12 13	1.0 2.0 2.0 1.0 2.0
						14	3.0
3	17	22.00	30.00	0.00	30.00	7 8 15 16 17	4.0 3.0 3.0 1.0 4.0
						18 19 20	2.0 3.0 2.0
	NETWORK	50.00	63.00	0.00	63.00		

****

PLANT LOCATION-ALLOCATION 3 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	6	18.00	22.00	0.00	22.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
						6 9	4.0 1.0
2	13	10.00	11.00	0.00	11.00	10 11 12 13 14	2.0 2.0 1.0 2.0 3.0
3	17	22.00	30.00	0.00	30.00	7 8 15 16 17	4.0 3.0 3.0 1.0 4.0
						18 19 20	2.0 3.0 2.0
	NETWORK	50.00	63.00	0.00	63.00		

****

PLANT LOCATION-ALLOCATION 2 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	6	25.00	55.00	0.00	55.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
						6 9 10 11 12	4.0 1.0 2.0 2.0 1.0
						13	2.0
2	17	25.00	45.00	0.00	45.00	7 8 14 15 16	4.0 3.0 3.0 3.0 1.0
						17 18 19 20	4.0 2.0 3.0 2.0
	NETWORK	50.00	100.00	0.00	100.00		

****

PLANT LOCATION-ALLOCATION 1 MEDIAN SOLUTION

	MEDIAN	PLANT SIZE	FREIGHT COST	PLANT COST	TOTAL COST	NODES SERVICED	VOLUME SUPPLIED TO EACH MEDIAN
1	6	50.00	187.00	0.00	187.00	1 2 3 4 5	3.0 1.0 4.0 2.0 3.0
						6 7 8 9 10	4.0 4.0 3.0 1.0 2.0
						11 12 13 14 15	2.0 1.0 2.0 3.0 3.0
						16 17 18 19 20	1.0 4.0 2.0 3.0 2.0
	NETWORK	50.00	187.00	0.00	187.00		

****

Figure 14

In some instances it may be possible to combine the results of both the forward and backward recursive solutions. Let t_j (if it exists for a particular case) be some time period, $0 > t_j \geq r$, in which the solutions of both methods are identical. Then the sequencing of the forward recursive method from period t_1 to t_j must always be at least as good as that given by the backward recursive method over the same time intervals. Alternatively, for any time period t_r back down to t_1, the optimal sequencing can always be explicitly solved by the dynamic programming methods of the following section providing r is not too large.

A Dynamic Program for the Optimal Sequencing of Plant Locations Over Time

The application of dynamic programming to plant location through time has received surprisingly little attention in the past. Bellman (1965) has demonstrated how this approach may be applied to the case of static systems but, to the author's knowledge, only Scott (1969) has explicitly considered the problem in its temporal context. Scott has provided a careful analysis of the potential advantages of dynamic programming over forward recursive procedures, but suggests that it is necessary to evaluate all r! possible sequences to find the global minimum, after solving for the solutions implicit in equation (20) and (21). Actually, as will be demonstrated, far fewer evaluations are needed. Indeed if

$r = 10$, only a few thousand calculations are required to solve the minimum cost sequence compared to several million if all possible orderings are examined. As it happens the computational difficulties associated with applying dynamic programming to plant location over long periods of time lie less in the sequencing problem than in the seemingly unavoidable need to evaluate a minimum cost allocation for all network nodes with respect to all $\binom{r}{t}$ sets of possible locations at each period t in the analysis.

The fundamental recursive relationship of discrete dynamic programming is due originally to Bellman (1957). Improvising on a general format given by Wagner, (1969) this relationship may be explicitly expressed in terms of the plant location problem as follows:

$$(24) \qquad f_k(s) = \min \left[S_s + f_{k-1}(j) \right] \text{ for all } k=1,2,\cdots,r$$

where

(a) $f_k(s)$ is a minimum cost (aggregate distance) policy decision taken in state s with k more plants to locate.

(b) S_s is the cost of serving the network in state s with k locations in solution and

(c) $f_{k-1}(j)$ is the optimal aggregate location costs back out of state j with k-1 plants to locate i.e.

optimal from state j back through all antecedent plant sequencing decisions in all previous stages to the commencement of stage 1 where r locations are in solution.

Since the application of this recursion is backwards in time any state j will contain one more location than any entering state s, so that at the beginning of stage k=1 (time period t=r) all pre-determined r locations are present, and at the termination of stage k=r (time period t=1) each state j will contain only one location. At every stage K in the dynamic process $\begin{pmatrix} r \\ r-k \end{pmatrix}$ new states enter into the solution, each such state s being a unique combination of the original r locations with r-k terms. For each one the decision $f_k(s)$ taken with respect to some particular state j with r-k+1 plants (of which the locations belonging to state s must be a subset) is such that the act of transferring to or routing through state j constitutes a least cost optimal plant sequence for state j back to stage one. Since the function $f_{k-1}(j)$ embodies the minimum cost plant sequence for some state j back through all preceeding stages to k=1, the decision sought for states at stage k is simply the minimum of $S_s + f_{k-1}(j)$ over all j for which state s is a subset. This explanation is aptly summarized by the principle of optimality which states that:

> *An optimal policy must have the property that regardless of the route taken to enter a particular state, the remaining decision must constitute an optimal policy for leaving that state.* (Wagner, 1969, p. 257)

The choice of some state j for any state s constitutes the decision by which state s is left.

These concepts can be clarified with reference to the hypothetical example of four stages shown in Table 5. Assume for some network a four location solution given by (20) with nodes 1, 2, 3 and 4 as the medians of the graph and S_4, the aggregate cost of supplying each destination from its closest median, as 10. The objective is to order the sequencing of the plants over time such that the cumulative long term cost function over all periods is minimized. In Table 5 there are four matrices each corresponding to one stage in the analysis with the columns referencing some state j and each row some entering state s. For any stage k the number of locations in state s will indicate the time period r-k, while those for state j have r-k+1 terms, each j being the signpost to the s=j entering state of period t+1. The number of entering states at any step will be all the r-k combinations of r, i.e. $\binom{r}{r-k}$ while for any stage k+1 the j states will be the entering states of the kth stage.

At the commencement of stage 1, state j is the median series 1, 2, 3 and 4, and the entering states are 123, 124, 134 and 234, the combinatorial series down the

rows of Table 5(a). Let the minimum cost S_s of supplying the network for each one of these combinations or locations be $S_{123} = 16$, $S_{124} = 18$, $S_{134} = 15$ and $S_{234} = 19$ respectively. The cost entry in the first cell (16 + 10) represents the cumulated costs of serving the graph with medians 123 in time period 3 and 1234 in the fourth or final period. The inset box on the left contains the integer 4, which labels the median which will be added to the entering state 123 on its exit from this stage should it prove subsequently to be part of the optimal sequencing route. The inset box on the right of each cell contains a counter which references the row in the preceeding stage matrix wherein can be found that member of the left inset box which is to be added in the next time period t + 1. The optimal decision policy for any entering state with respect to its feasible j exits is stored in the decision cell, the penultimate right most column of each matrix. Since for stage 1 there is only one exit state j no choice is involved in the contents of this decision cell.

For stage k=2 the entering states (there are $\binom{r-k}{k}$ of them) represent all the possible locational combinations of time period 2, and the leaving states indicate the routes to all the combinations of entering states in period t=3 (k=1). Consider the value of $S_{12} = 25$, the minimum cost of supplying all network destinations from medians land 2. For this entering state either median 3 or 4 will be added at the

Table 5

OPTIMAL SEQUENCING OF PLANT LOCATIONS
FOR A HYPOTHETICAL EXAMPLE

(a) k=1 t=3 to 4

Exit State j

		1 2 3 4	Decision	Cost
	123	[4] 16+10 [1]	[4] 26 [1]	16
Entering	124	[3] 18+10 [1]	[3] 28 [1]	18
State s	134	[2] 15+10 [1]	[2] 25 [1]	15
	234	[1] 19+10 [1]	[1] 29 [1]	19

(b) k=2 t=2 to 3

Exit State j

		123	124	134	234	Dec.	Cost
	12	[3] 25+26 [1]	[4] 25+28 [2]	X	X	[3] 51 [1]	25
	13	[2] 27+26 [1]	X	[4] 27+25 [3]	X	[4] 52 [3]	27
Ent.	14	X	[2] 28+28 [2]	[3] 28+25 [3]	X	[3] 53 [3]	28
St. s	23	[1] 24+26 [1]	X	X	[4] 24+29 [4]	[1] 50 [1]	24
	24	X	[1] 26+28 [2]	X	[3] 26+29 [4]	[1] 54 [2]	26
	34	X	X	[1] 27+25 [3]	[2] 27+29 [4]	[1] 52 [3]	27

(c) k=3 t=1 to 2

Exit State j

		12	13	14	23	24	34	Dec.	Cost
	1	[2] 34+51 [1]	[3] 34+52 [2]	[4] 34+53 [3]	X	X	X	[2] 85 [1]	34
Ent.	2	[1] 33+51 [1]	X	X	[3] 33+50 [4]	[4] 33+54 [5]	X	[3] 83 [4]	33
St.s	3	X	[1] 35+52 [2]	X	[2] 35+50 [4]	X	[4] 35+52 [6]	[2] 85 [4]	35
	4	X	X	[1] 32+53 [3]	X	[2] 32+54 [5]	[3] 32+52 [6]	[3] 84 [6]	32

(d) k=4 t=0 to 1

		1	2	3	4	Decision	Cost
Entering States	0	[1] 0+85 [1]	[2] 0+83 [2]	[3] 0+85 [3]	[4] 0+84 [4]	[2] 83 [2]	0

Table 5 cont'd.

termination of the stage. If the choice is via state j, containing plants 1, 2 and 3, then from the previous stage k-1 = 1, it is known that the combined costs of solutions 123 in t_3 and 1234 in t_4 are 26 as indicated by the decision cell in row 1 of stage 1. Similarly if the routing is through state j_2 = 124, the cost with respect to this state is 28. Therefore, if locations 1 and 2 prove later to be in the optimal sequence in time period 2, the least cost decision, f_2s, will be to add median 3 rather than 4 at the termination of this stage, since 25 + 26 < 25 + 28, governs the minimum order for this state s back to t = 4 (k=1). The decision cell contains the value of 51, the cumulative costs of the three period solutions of 1,2; 1,2 and 3; and 1,2,3 and 4. The left inset box contains a 3, the requisite branching decision, and the right inset box the value of 1 which is the column index of state j = 123. Note that since the j states 134 and 234 do not include both medians 1 and 2 as subsets, they are not feasible exit paths for state s=1,2.

At stage k=4 no states actually enter into the solution since the time period reference is r-k=0. But from this state there are four possible exits, medians 1, 2, 3 or 4 which may be entered at commencement of the next time period t=1 (k=3). It is worth emphasizing that at any step in the dynamic process the only information required from the previous stage is the values contained in the decison cells of that matrix. This illustrates the

key principle of dynamic programming that for any state s the least cost path with k stages remaining is dependent only upon the result of the immediate decision, and some appropriate optimal policy established with k-1 stages left.

Upon completion of all stages the value of the minimum cost sequencing, given in the single decision entry of Table 5(d), is 83. To trace back the actual sequence which reflects this optimum through all four time periods is now an easy matter. The left inset box of the decision cell of 5(d) indicates that median 2 should be the first to be located in period t_1. The right inset box contains the integer 2 and gives the appropriate decision cell to reference in stage 3. Entrance to this cell in row 2 of 5(c) leads to the selection of median number 3 as the next plant to be sited. Continuing back in this manner results in the addition of median 1 at the commencement of t=3, and finally source number 4 completes the ordering in the last time period. The optimal sequence therefore locates plants 2, 3, 1 and 4 in that order. The cost of the allocations in each time period are stored in those cells of the cost column which are appropriate for the entering states, and for this example are 33 + 24 + 16 + 10 = 83.

A computational short cut can now be introduced for finding all the feasible paths of the matrix for some stage

k. Remember that a feasible path exists only if state s is a subset of state j. The number of such paths for any k is given by

$$\begin{Bmatrix} r-k+1 \\ r-k \end{Bmatrix} \times \begin{Bmatrix} r \\ r-k+1 \end{Bmatrix}$$

which compares to

$$\begin{Bmatrix} r \\ r-k+1 \end{Bmatrix} \times \begin{Bmatrix} r \\ r-k \end{Bmatrix}$$

total elements in any matrix. The number of active cells down each column is

$$\begin{Bmatrix} r-k+1 \\ r-k \end{Bmatrix}$$

and across each row is k. In the example of Table 5 feasible paths account for half or more of all cells for any stage in the cycle. For the larger problem however, this ratio may be very much less. If r = 10 then the total number of feasible routes (about 4000) over all r time periods is less than one fourtieth of all possible paths. Because k will be at least equal to (and may be very much less) than j at any stage, it is computationally inefficient to evaluate by direct comparison each member of some entering state s against every member of each exit state j to test for feasibility. Even if the execution for any row can be terminated

upon counting k active cells, more than 50 per cent of all infeasible paths will still need be examined during the analysis. Since between half a dozen to more than one hundred operations (depending upon the sizes of r and k) may be necessary for the complete evaluation of each cell, significant computational savings will be realised if all feasible paths can be directly obtained. This task, in which the direction of the analysis will be down columns instead of across rows, can be accomplished as follows.

Consider state j=123 in stage k=2 of Table 5(b). There are three descendent subsets of r-k terms each, 12, 13 and 23 respectively. These combinations occupy rows 1, 2, and 4 of the six cells in column 1, each one of which references a unique descendant of all the $\binom{r}{r-k}$ possible combinations of entering states for that stage. Furthermore every one of the descendants can be determined in strict lexicographical order. If each one of the three subsets of j=123 can be given a value which corresponds exactly to the rank of that combination in terms of all $\binom{r}{r-k}$ descendants, then the cell which each should occupy will be uniquely determined. The rank of a combination can be obtained by means of the following method.

Let $\binom{r}{m} = \binom{r}{r-k}$ be the number of combinations of locations or entering states at each stage in the analysis. Let $(b_1, b_2, \cdots, b_m)$ denote one of these combinations of

the r integers, and further define $(a_1\ a_2 \cdots, a_m)$ as a combination of r integers 0, 1,$\cdots$,r-1 such that

$$0 \leq a_1 < a_2 \cdots < a_m \leq r \tag{25}$$

which implies that each element of the series $(a_1, a_2, \cdots, a_m)$ is one less than its corresponding counterpart in $(b_1, b_2, \cdots b_m)$. If R_v is defined as the required rank of the series $(b_1, b_2, \cdots, b_m)$ then

$$R_v = \binom{r}{m} - \sum_{j=1}^{m} \binom{r-1-a_{m-j+1}}{j} \tag{26}$$

This derivation adapted from (Lehmer, 1964) concludes this section on dynamic plant location over time.

CHAPTER IV

AN APPLICATION OF THE MEDIAN ALGORITHM TO RURAL HOSPITAL LOCATIONS IN A SUB-REGION OF BRITISH COLUMBIA

In the preceeding chapter a detailed description of an algorithm for estimating the medians of a weighted graph subject to side constraints was presented. In this chapter on application of the basic model (without constraints) to hospital locations in a rural sub-region of British Columbia is given. Although the algorithm is computationally feasible for large scale network problems, if judicious choice of the partitioning heuristics is made, only one part of the province will be subject to a detailed analysis here. The outline for this chapter is as follows:

1) A suitable sub-region for analysis is first chosen.

2) Specified medical data pertaining to this region is summarized in terms of a network of nodes and links where each node, the centre for some discrete surrounding area in space, represents in turn the demand over a one-year period for adult medical, adult surgical and obstetrical care, and each link represents the distance (normally road distances) between some pair of nodes. A subset of of these nodes is further designated as potential sites.

3) The algorithm is tested upon the network in accordance with the stated aims and objectives detailed in Chapter I.

4) A short discussion of the results follows together with an appraisal of the validity of the general model formulation for this type of location problem.

Choice of Regional Study Area

The objective of this section was to attain a partitioning of the province of British Columbia into subregions such that each one formed a relatively self-contained unit with respect to the magnitude and direction of the linkages and flows within its bounds. To avoid possible bias which might be present if the actual pattern of hospital flows cuts across any natural nodal region, traffic flow data, collected by the provincial government during the summer of 1966, was utilized as the criterion upon which the partitioning was based. Notwithstanding some bias inherent in the recognition that this data was and is collected primarily for the purposes of tourism, it was felt that this information was the best readily available source for forming the crudely defined nodal regions necessary for this stage of the analysis.

Traffic flow data was gathered for eighty control points distributed randomly throughout the province. Given the traffic flow across each of the designated control

points, the distance of the link between any two points, and an assessment of the average traffic speed over any link (taken somewhat arbitrarily to be 35 m.p.h., with 25 m.p.h. on gravel roads and 40-45 m.p.h. on freeways), an estimate of road traffic density, or more specifically the average distance between vehicles over any link, may readily be obtained. For each such control point the weighted average of this measure over all links connected to it was calculated. Given one such value for each control station a grouping of the points was sought, subject to a contiguity constraint, for which the average intra-group differences among the points might be minimized while maximizing the average inter-group dispersion. Although this objective function is not easily realised the well-known hierarchical step grouping algorithm of (Ward, 1963) will tend to provide a good estimate of the optimal grouping, particularly where the natural clustering is strong.

Application of this algorithm (for which an efficient computer code incorporating a dendogram is given in Appendix 4) to the vehicle spacing measure revealed a well defined regionalization with six groups remaining. These regions were identifiable as a) Victoria and its environs, b) north-central and north Vancouver Island, c) Vancouver and the lower mainland as far east as Hope, d) the Sechelt peninsula, e) the mainland north of Lytton to Prince George and

west to Prince Rupert and f) south-eastern British Columbia from Hope to the Alberta border, bounded on the north by the Trans-Canada highway and on the south by the boundary with the United States. The last of these regions, wherein contrasts exist between closely clustered sets of communities where opportunities for in-patient hospital care are varied, and other areas of more dispersed settlement where distances between towns correspondingly reduce such choices, seemed to offer the best opportunity for a rigorous testing of the model, and was accordingly chosen as the sub-area for analysis.

Generation of a Medical Demand Network for South-Eastern British Columbia

As shown in Figure 15 the defined study area of south-east British Columbia covers some twenty-nine school districts which are the smallest geographical base for which medical data can be assembled from collected information. From 1966 data made available by the British Columbia Hospital Insurance Scheme in Victoria, hospital patient separations (diagnosis on discharge) were obtained disaggregated by medical, surgical and maternity categories. For each school district the following information was compiled:

1) The number of separations for each of the three medical categories over the one year period (Table 6).

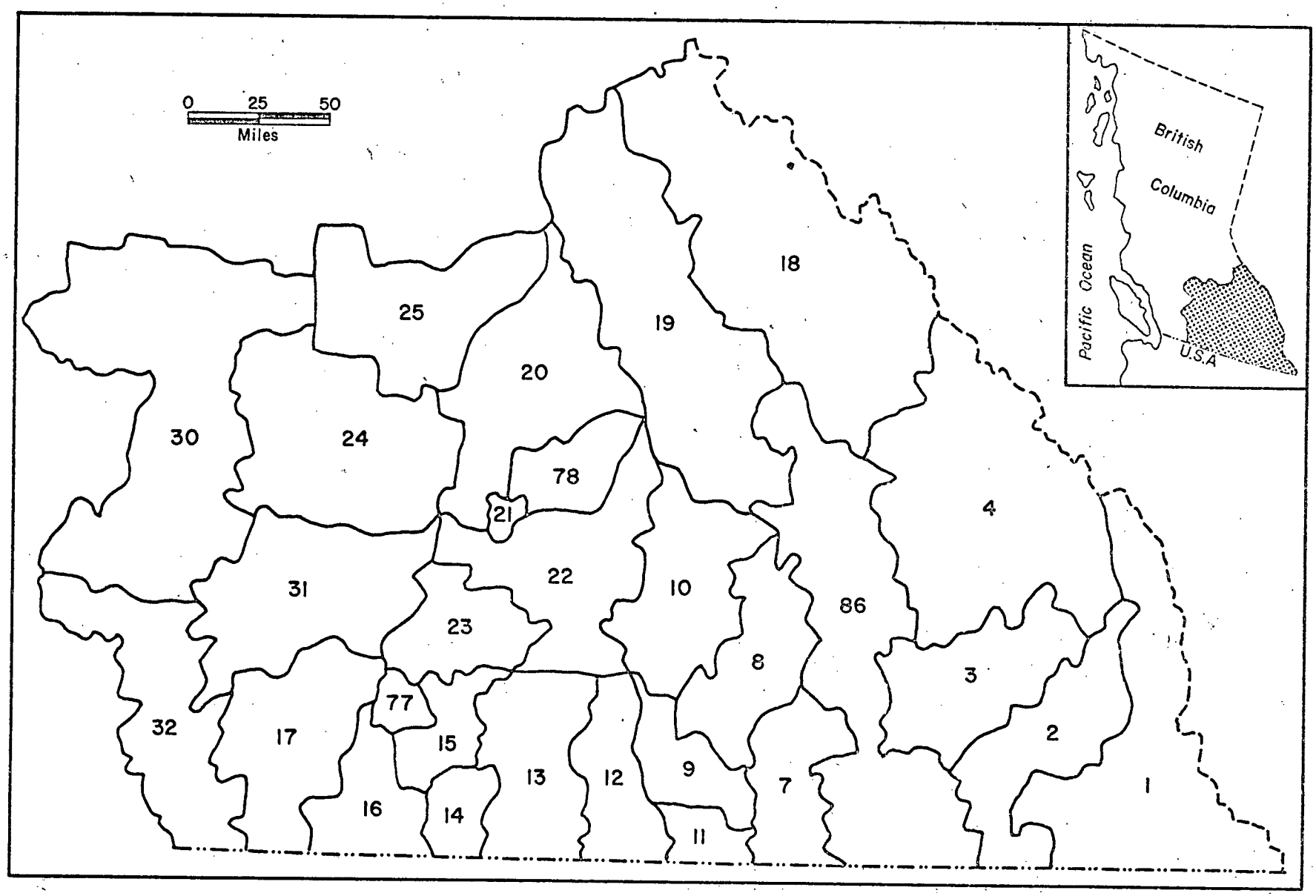


Figure 15

School District Locations in South-East British Columbia

Table 6

NUMBER OF SCHOOL DISTRICT SEPARATIONS BY MEDICAL CATEGORIES

S.D.	Population	Medical	Surgical	Maternity
1	6,624	521	473	126
2	11,051	645	625	267
3	8,665	389	825	150
4	4,461	352	314	93
7	16,011	875	915	305
8	3,336	256	255	71
9	11,319	533	790	270
10	3,269	164	215	63
11	24,138	984	1922	463
12	5,300	291	271	91
13	2,739	97	163	45
14	8,116	524	541	139
15	17,783	966	1140	324
16	2,438	194	203	37
17	2,998	252	215	46
18	5,696	314	265	172
19	8,237	382	379	197
20	11,911	1038	801	214
21	3,502	272	186	57
22	21,336	889	1112	355
23	33,730	1350	1770	556
24	39,249	1189	1661	832
25	1,895	61	90	62
30	6,519	497	462	184
31	7,465	419	513	196

Table 6 cont'd.

S.D.	Population	Medical	Surgical	Maternity
32	5,878	426	322	139
77	4,691	422	321	59
78	3,055	264	230	47
86	11,141	661	690	197

2) The number of separations discharged from each of the thirty-three hospitals operating in the sub-region, that is the patient flow from each school district to each hospital.

3) The total population of each school district.

Since the scale of most of the school districts proved to be too large to be of much practical value for analysis, it was necessary to disaggregate this information into a much finer net, one where each unit could be summarized by a single point or node. This was successively achieved in the following stages, each one necessarily entailing a certain measure of estimation at each step.

a) Using enumeration district population maps compiled from the National Census of 1966, each school district was disaggregated into these smaller geographical units for which the exact population figures were known. In a few instances the boundaries of the enumeration and school districts do not coincide and it was necessary to make an estimate of the proportion of enumeration district population which lay upon each side of the school district boundary. These estimates were adjusted to ensure that the sum of all enumeration district populations within the bounds of a given school district was set equal to the known population total of that school district.

b) The centres of population gravity for each of the enumeration districts were approximated with the aid of a

population dot distribution map of British Columbia prepared following the 1961 census by (Farley, et. al., 1961). In some cases depending upon the local distribution two or more points were assigned to the same enumeration district, while in others one point served to summarize the population of more than one district or parts of several, if population concentrations lay on either side of an enumeration district boundary. A set of geographical co-ordinates were then recorded for each of the 168 points compiled.

c) These co-ordinates together with their attached population weights enabled the centroids of the school districts themselves to be ascertained in terms of all the points lying within their boundaries. This analysis gave thirty-five such values of which six represented school districts surrounding the defined sub-region.

d) Incidence rates for each of the school districts were next compiled disaggregated by medical category. These rates were based upon the number of patient separations per 1000 population excluding all cases referred to hospitals beyond the region except those to Chilliwack, Lillooet and 100 mile house.

e) Finally an estimate of the demand or actual number of patient separations generated by each enumeration district was required. In utilizing the given co-ordinates of the school district population centroids, trend surface analyses were applied to all three sets of medical incidence data.

Since only thirty-five points were involved a sixth order trend surface analysis accounted in every case, as might be expected with so few values, for more than 90 per cent of the variation in each case. Attempts to use the sixth order trend equation as a means of estimating the incidence rate at each enumeration district centroid proved to be illusory however, owing to the extreme instability of the surfaces at their edges. The trends proved to be ill-defined up to the fifth order, little more than 50 per cent of the variation being explained at this level. As a consequence the generated surfaces did not provide a satisfactory approximation to the true ones so that the centroid estimations, particularly at the edges of the region, were subject to gross distortions. The required centroid incidence rates were finally interpolated from contour maps generated for each of the three sets of school district data. Given the enumeration centroid populations it was then a short step to calculate the estimated demand over each of the 168 enumeration nodes (Appendix 1). Figure 16 gives the position of all the centroids (nodes) with respect to their appropriate schools districts, together with the locations of the three hospitals on the periphery of the region.

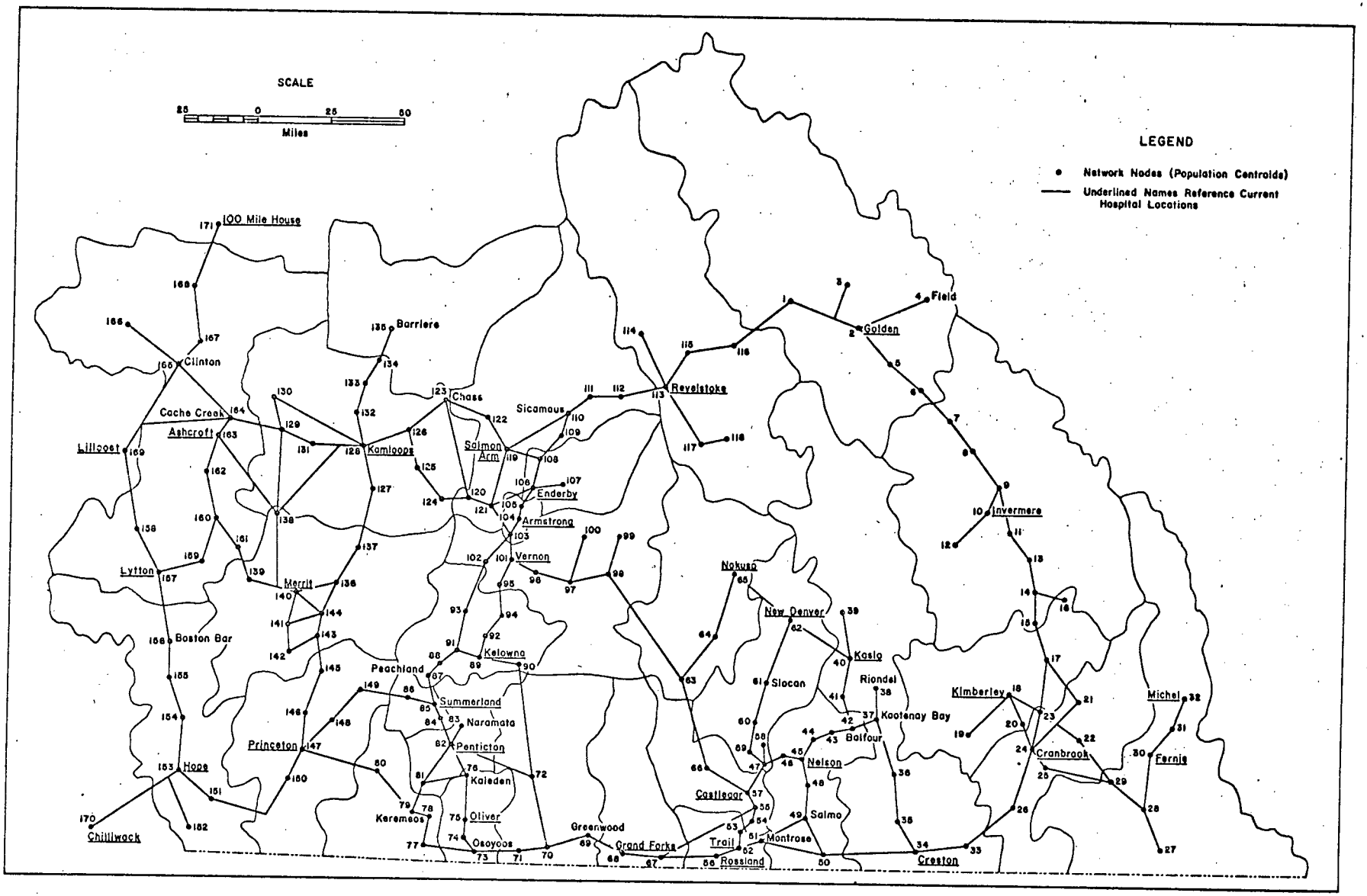


Figure 16

Medical Demand Network: Node and Potential Hospital Sites

A Measure of the Locational Efficiency of the Hospital System of South-East British Columbia

By the methods specified in the preceeding section three medical demand networks were condensed from population census data and hospital separation tapes. Figure 16 provides the cartographic representation of these networks, each one consisting of the same 171 nodes, of which 52 were assigned the status of potential hospital sites including the 33 facilities which are currently in operation in this region. The choice of each potential location was simply made on the basis of including nearly every node in the network whose population exceeded 1000 people.

A short path matrix (see Appendix 2) was next prepared between every node and each potential site, the distance between each node and itself being assigned, where applicable, a value of one. In the first three runs (one for each medical category) the solution was constrained to locations currently operating in and at the periphery of the area. These constraints were relaxed in the second series of analyses, the objective being to test the ability of the median algorithm to find sets of locations for which the total patient mileage on the network was less than that given by the constrained solutions.

The results of these analyses (presented in Table 7) show that the median algorithm solutions were consistently better than the constrained ones to the extent of 15 to

Table 7

COMPARISON OF DISTANCE MINIMIZING RESULTS: CONSTRAINED AND UNCONSTRAINED SOLUTION

Category	Total Patient Miles	
	Constrained	Unconstrained
Medical	89,021	76,494
Surgical	96,860	83,077
Maternity	32,288	26,149

20 percent in each case. Table 8 provides details on the different hospital sets obtained in each solution. Since the algorithm is heuristic in design there can be no guarantee that these patterns of locations are optimal, but a careful scrutiny of data and results suggest no obvious improvements and it is conjectured that the obtained solutions are at least close to the global minima. Note that the three peripheral locations of Chilliwack, Lillooet and 100 Mile House are dropped in the unconstrained solutions in each case. Actually, since these facilities are in viable locations with respect to the surrounding communities they serve outside the bounds of the specified region, the actual measure of locational inefficiency implicit in the results will be somewhat exaggerated.

Prediction of Actual Flow Patterns Within the Sub-Region

Although the results given by the median algorithm reflect favourably upon the intrinsic structural strength of the model formulation, its applicability to the problem at hand still requires some further justification. Implicit in the analysis so far has been the central position of distance, for it is upon this variable that the locational structure of the regional hospital system has been essentially assessed. Yet the significance of this factor as a determinant of the actual pattern of trip distributions still remains to be positively demonstrated for this

Table 8

COMPARISON OF CONSTRAINED AND UNCONSTRAINED LOCATION SYSTEMS

Hospital	Const. Solution	Unconstrained Solution		
		Medical	Surgical	Maternity
Golden	X	X	X	X
Invermere	X	X	X	X
Kimberly	X	X	X	X
Cranbrook	X	X	X	X
Fernie	X	X	X	X
Michel	X	X		
Creston	X	X	X	X
Kootenay Bay		X		
Kaslo	X	X	X	X
Nelson	X	X	X	X
Salmo			X	X
Trail	X	X	X	X
Rossland	X		X	
Castlegar	X	X	X	X
New Denver	X	X	X	X
Nakusp	X	X	X	X
Grand Forks	X	X	X	X
Greenwood				X
Osoyoos		X	X	
Oliver	X	X		
Keremeos		X	X	X
Penticton	X	X	X	X
Summerland	X	X	X	X
Kelowna	X	X	X	X

Table 8 cont'd.

Hospital	Const. Solution	Unconstrained Solution		
		Medical	Surgical	Maternity
Vermon	X	X	X	X
Armstrong	X			
Enderby	X	X	X	X
Revelstoke	X	X	X	X
Salmon Arm	X	X	X	X
Chase		X	X	X
Kamloops	X	X	X	X
Barriere			X	X
Merritt	X	X	X	X
Princeton	X	X	X	X
Hope	X	X	X	X
Lytton	X	X	X	X
Ashcroft	X	X	X	X
Clinton		X	X	X
Lillooet	X			
Chilliwack	X			
100 Mile Hs.	X			

example. To test this question the number of separations from each node at each of the 33 operating hospitals were aggregated by their school district designations, and then compared to the actual flow patterns condensed from the hospital tapes.

These results are shown in Table 9 in which the entries down each column are computed percentages obtained from solution of the expression $(a - b)/a \times 100$, where a is the number of separations from any school district and b is the sum of the deviations between actual and predicted separations over all hospitals. The overall results were encouraging to the extent that the range of prediction varied from a low of 72.9 per cent for surgical care to a high of 86.7 per cent for medical. These results tend to confirm that a fundamental role is played by accessibility in rural medical trip pattern generation, and so gives strength to the verification of the median algorithm as a tool for hospital location planning in the rural inter-urban context.

Not all flows over the region are distance determined however, as is shown by individual anomalies in the values of particular school districts. Consider the example of school district 8 where the computed values are 31 and -33.3 for medical and surgical care respectively. The negative values for surgical actually indicate that the sum of the flow deviations is greater than the actual number of

Table 9

FLOW PATTERN PREDICTION: DISTANCE MINIMIZING AND INTERACTANCE MODELS

S.D.	Distance Minimizing Model			Gravity Interactance Model		
	Medical	Surgical	Maternity	Medical	Surgical	Maternity
1	90·4	78·4	91·0	91·1	76·0	91·5
2	92·6	78·2	92·4	90·0	78·6	92·5
3	88·7	91·2	96·4	94·2	90·0	96·7
4	86·3	82·8	83·1	87·7	92·1	83·6
7	83·8	91·5	93·9	95·7	91·3	96·0
8	3·1	-33·3	32·9	73·0	61·7	53·0
9	83·5	45·3	86·5	80·5	48·4	87·3
10	41·5	-27·4	65·0	50·6	20·2	67·1
11	93·5	95·4	82·2	93·7	95·4	82·3
12	84·9	4·1	83·5	86·6	21·2	83·9
13	62·9	42·3	70·9	75·6	26·1	71·7
14	79·3	61·9	77·3	80·0	71·3	77·4
15	96·9	90·4	96·5	97·1	90·7	96·4
16	80·4	62·6	69·8	71·0	67·3	72·8
17	89·7	68·4	85·4	92·3	81·7	85·9
18	89·2	81·9	94·2	89·6	84·7	94·2
19	84·8	47·7	92·0	85·9	51·9	92·0
20	84·4	51·6	66·7	82·7	60·7	70·0
21	94·1	28·0	28·8	93·9	28·2	29·3
22	92·8	87·4	93·9	94·4	89·1	93·8
23	93·3	92·8	98·5	96·3	93·0	98·3
24	91·1	91·0	95·4	95·8	91·6	95·4
25	86·9	100·0	99·3	87·3	98·7	99·5
30	89·9	62·8	75·1	86·6	80·7	75·1
31	91·4	65·3	86·3	94·3	83·1	87·1

Table 9 cont'd.

S.D.	Distance Minimizing Model			Gravity Interactance Model		
	Medical	Surgical	Maternity	Medical	Surgical	Maternity
32	91·1	84·5	89·2	93·5	83·5	92·3
77	88·2	20·2	41·4	89·2	30·5	43·1
78	80·3	33·0	79·6	87·1	58·7	79·2
86	76·4	42·3	67·1	76·8	61·4	70·5
TOTAL	86·7	72·9	85·2	89·6	78·7	87·7

trips. This school district contains the following labelled nodes 47, 58, 59, 60, 61 and 62, of which only the last two are assigned to New Denver hospital. The other four are directed south in each case to Castlegar which is approximately only one mile nearer than the larger facility at Nelson. In fact more than 50 per cent of these separations were treated at the Nelson facility. This illustrates again the cardinal disadvantage of simple distance minimizing models in situations where, for any area of patient demand, more than one treatment centre provides a feasible alternative for care. This type of situation occurs comparatively infrequently in this particular application since the rural context of the study, where distances between communities are relatively great, tends to reduce the viable alternatives for patient care except in special cases where treatment demands referral to those larger hospitals where the requisite facilities are available.

Notwithstanding this caveat it is fair to say that, for the purposes of optimizing trip distribution prediction, the interactance model offers a potentially more fruitful approach. The structure of this model is behaviourally more realistic for it allows for the possibility of assigning patient demand from any one area to more than one hospital. A gravity model of the genre proposed by Huff, (1963) was accordingly structured to test for improved trip distribution prediction, though it should be emphasized

again at this point that the interactance model is not at all suited for optimizing locations over a network, which is the central concern of this study. The Huff model is of the general form

$$(27) \qquad T_{ij} = \left(\left(P_j^g / d_{ij}^m\right) / \sum_{i=1}^{n} \left(P_j^g / d_{ij}^m\right)\right) \cdot C_i$$

where T_{ij} is the flow from node i to hospital j

P_j is the size of hospital, measured here in terms of the number of separations treated.

d_{ij} is the distance from node i to hospital j

C_i is the estimated demand for treatment by medical category at node i and

g and m are best fitting exponential parameters determined empirically in this application.

The rationale for this model expressed as it is here in its simplest form is that, ceteris paribus, the trip distribution between any node and hospital can be expected to vary as a direct function of hospital size and as an inverse function of its accessibility measured in simple distance.

Application of this model to the data gave the following results. The parameter g was calculated to be 1·5, 2·3, and 1·4 for medical, surgical and maternity care respectively, while 5·5, 4·4, and 5·7 proved to be the best fitting values for m. Aggragating the nodes by school district gave predictions of 89·6, 78·7 and 87·7 per cent for each of

the three medical categories which, as Table 9 shows, proved to be a slight improvement on the overall performance of the distance minimizing model.

Inspection of the results for individual school districts shows striking improvements for school district 8, although for the surgical data the general increases were sometimes less spectacular, remaining, on occasion, below the thirty per cent predicted level. It is difficult to provide satisfactory explanations for these particular results without recourse to detailed knowledge of the medical patterns and linkages over these areas. One of the weaknesses of this interactance model application and indeed of most such algorithms where the parameters are empirically determined as is normally the case, is that there is usually little a priori theoretical justification which can be ascribed to the functions g and m. The measure for hospital size is at best a crude indicator and is certainly much less satisfactory than one which might be inclusive of the range and scope of facilities present at each institution. Note too that the distance measure gives no recognition at all to the particular professional linkages and patterns of referral which might prevail over certain parts of the region. Moreover it is somewhat limiting to suppose that the exponential effect of distance decay incorporated into this model is necessarily constant over its whole range. Many decay functions display a rapid decline over the first

few miles and amore gradual tailing off thereafter. Indeed the interesting analysis of (Morrill and Kelley, 1970) seems to bear this out for hospital usage, and the functions they tested were certainly of a more varied and complex form than the simple exponential effect used here.

Although all these considerations are indicative of real weaknesses in this particular formulation, the obtained results nevertheless suggest that interactance structures are appropriate to this type of problem in this setting. Where possible however, the determination of a priori parameters based upon in-depth behavioural studies of the medical patterns of a region is certainly to be preferred to a posteriori parameters on both theoretical and practical grounds. Such an approach would surely help to clarify particular idiosyncracies such as those reflected in the lower predicted values of Table 9. However the acquisition of such detailed knowledge is both expensive and time consuming and for this application has lain beyond the scope of the analysis.

A Method for Estimating Optimal Plant Size Over the Current System of Hospitals in the Region

As a corollary to the main aims of this chapter, a method for testing and relocating the imbalance of capacity over existing hospitals on a network is suggested by combined usage off the median and interactance models. In this problem as formulated by Morrill and Kelley (1970), and as described in Chapter II, the aim is to redistribute capacity

(hospital size) over a given set of centres so that patient travel is minimized subject to the behaviourally more realistic constraints of an interactance model structure.

The Morrill-Kelley method begins with some given measure of hospital size, allocates patients with the interactance model among the set of hospitals, calculates a new set of sizes based upon these allocations, and continues the iteration until an acceptable level of divergence is obtained between two successive cycles. The method proposed here differs from this procedure in that the set of initial hospital sizes is not taken as given, but is derived from some measure, either directly or indirectly, based upon the prior assignment of each patient demand area to its nearest facility. It was suggested in Chapter II that this method should tend to give a better redistribution, and experiments with some small sample networks have shown this to be so on occasion.

The results of an analysis using both methods on the generated medical and surgical data proved however, to be inconclusive in this respect, since both algorithms converged to approximately the same solutions. Actually the measured improvement was less than 10 per cent in both cases, although the proposed method required slightly fewer (3 to 4) iterations to converge. Perhaps these results are not too surprising on a network where nearly every

incorporated community has its own hospital and where, as a consequence, the distribution of plant capacity is near optimum at the outset.

Chapter V

SUMMARY AND CONCLUSION

Rapid changes in medical technology and in the material expectations of societies over the last decade have created demands for fundamental changes in the organization of health care. A review of the pertinent literature over this period reveals a quickening interest in the concept of regional planning as a suitable framework within which to institute policies, made urgent by the gross inflationary nature of medical costs, and by the growing awareness of the marked discrepancies which exist in the receipt of medical care among various segments of the population. The translation of these desires into operational planning tools has been the focus of much on-going effort in recent years. Among the spatial components of this field the problem of designing efficient hospital location systems for rural and developing areas, where budgetary limitations place severe constraints upon the number of available facilities, has been deemed a matter of some significance (Lord Llewelyn-Davies, 1966).

The results of this study strongly suggest that in the inter-urban context at least, the role of accessibility

is of fundamental importance in conditioning the magnitude and direction of many patient to hospital trips. This fact gives some prior justification for considering hospital location as a particular case of the more generalized problem of finding the medians of a weighted graph subject to certain side constraints. An algorithm for estimating these medians was presented in some detail in Chapter III. The heuristics devised tentatively appear to be at least as successful as those of other numerical algorithms, and are computationally more feasible than many for the large scale problem. It is conjectured, though by inspection rather than definitive proof, that they will generally lead to convergence with optimal or near optimal solutions.

Improvements in the median algorithm can arise only from the development of more efficient partitioning and perturbation techniques. The partitioning procedures of this model seem to be particularly successful for breaking open the graph in ways which will tend to lead to subsequent rapid convergence. The perturbation techniques can certainly be improved, but only at the expense of considerably expanding the search pattern to take account of particular, and perhaps unusual, network designs. The wisdom of such an expansion must however, always be judged against the prime advantage of heuristic over exact solution methods, that of more rapid computational convergence over extensive

network structures. The partitioning procedures alone should converge to results sufficiently good for the majority of practical applications.

The application of the median algorithm to the current hospital system of South-East British Columbia suggests that the current pattern of locations is not an optimal one. The inefficiency of the present locational structure with respect to the minimization of patient travel over the area, amounts to at least 10 per cent for each of the three types of care considered. The disadvantage of the algorithmic mechanism in restricting patient assignments to only one area, can be tempered by modifying the trip distribution with some suitably designed interactance model which, as demonstrated here and previously by Morrill and Kelley (1970), is behaviourally more realistic in this sense. If resources permit, then the a priori estimation of the interactance parameters is theoretically to be preferred to empirically determined functions.

Interactance models cannot however, be used to solve the quest for direct multi-location optimization on a connected weighted graph. Distance minimizing algorithm of the genre proposed here seem to offer at present the best potential tool for such problems, if accessibility can be shown to be a key variable and its minimization the prime goal. The operational algorithm developed in this

study should provide one decision-making basis from which to test the feasibility of proposed future hospital locations within a rural context, and under a wide variety of potentially measurable constraints.

LITERATURE CITED

Anon. "Regional Hospital Districts." British Columbia Medical Journal, 9 (1967), 473-4.

Andersen, R. and Hull, J.T. "Hospital Utilization and Cost Trends in Canada and the United States," supplement to Medical Care, 7 (1969), 4-23.

Arnold, M.F. and Hink, D.L. "Agency Problems in Planning for Community Health Needs," Medical Care, 6 (1968), 454-466.

Battistella, R.M. Directions in Regional Planning, Mimeographed, (1966).

Bellman, R. Dynamic Programming, Princeton, New Jersey: Princeton University Press, (1957).

Bellman, R. "An Application of Dynamic Programming to Location-Allocation Problems," SIAM Review, 7 (1965), 126-8.

Berry, R.E. "Returns to Scale in the Production of Hospital Services," Health Services Research, 2 (1967), 123-139.

Bodenheimer, T.S. "Mobile Units: A Solution to the Rural Health Problem," Medical Care, 7 (1969), 144-154.

Bridgman, R.F. "Integration of the Organization of Medical Care into Health and Town Planning," World Hospital, 3 (1967) 50-53.

Canadian Medical Association. Group Practice in Canada, Toronto: Ryerson Press, (1967).

Cardwell, R. and Kucka, K. "The Case for Co-ordinated Hospital Planning," Hospital, 35 (1961), 43-35.

Coleman, J.V. "Psychiatric Studies of Patient Needs in Emergency Services of General Hospitals," Medical Care, 5 (1967), 255-259.

Cooper, L. "Solutions of Generalized Locational Equilibrium Models," Journal of Regional Science, 7 (1967), 1-19.

Coughlin, R.E. Hospital Complex Analysis: An Approach to Analysis for Planning a Metropolitan System of Service Facilities, Unpublished Ph'd Dissertation, Philadelphia; University of Pennsylvania, (1964).

Donabedian, A. and Axelrod, S.J. "Organizing Medical Care Programs to Meet Health Needs," Annals of the American Academy of Political and Social Science, 337 (1961), 45-56.

Donabedian, A. "Promoting Quality Through Evaluating the Process of Patient Care," Medical Care, 6 (1968), 181-202.

Dorsey, J.L. "Physical Distribution in Boston and Brookline 1940 and 1961," Medical Care, 7 (1969), 429-440.

Duncan, J. "Flying Doctor Service of Africe," World Medical Journal, 13 (1966), 102-122.

Efroymson, M.A. and Ray, T.L. "A Branch-Bound Algorithm for Plant Location," Operations Research, 14 (1966), 361-368.

Farley, A. et. al. British Columbia: Population Distribution 1961. Department of Geography, University of British Columbia.

Feldman, E. et. al. "Warehouse Location Under Continuous Economies of Scale," Management Science, 12 (1966), 670-684.

Feldstein, M.S. Economic Analysis for Health Service Efficiency. Amsterdam: North Holland Publishing Company, (1967).

Feldstein, P.J. "Research on the Demand for Health Services," Milbank Memorial Fund Quarterly, 44 (1966), 128-165.

Friedson, E. Patients Views of Medical Practice, New York: Russel Sage Foundation, (1961).

Gould, P.R. and Leinbach, T.R. "An Approach to the Geographic Assignment of Hospital Services," Tijdschnift Voor Economique en Sociale Geografie, 57 (1966), 203-206.

Hakimi, S.L. "Optimum Location of Switching Centers and the Absolute Centers and Medians of a Graph," Operations Research, 12 (1964), 450-459.

Hilleboe, H. "Health Planning on a Community Basis," Medical Care, 6 (1968), 203-214.

Hoge, V.M. "Planning, Design and Construction of Hospitals-Analysis of Community Needs," World Hospitals, 3 (1967), 11-20.

Huff, D.L. "Defining and Estimating a Trading Area," Journal of Marketing, 28 (1964), 34-38.

Huff, D.L. "A Programmed Solution for Approximating an Optimal Retail Location," Land Economics, 42 (1966), 293-303.

Jacobs, A.R. and McLaughlin, C.P. "Analysing the Role of the Helicopter in Emergency Medical Care for a Community," Medical Care, 5 (1967), 343-350.

James, G. "Health Services and Community Needs," Medical Care, 5 (1967), 62-69.

Kilsclonk, R. "Travelling Clinics Bring Care to Cardston," Canadian Hospital, 43 (1966), 39-41.

Kissick, W.L. "Planning Programming and Budgeting in Health," Medical Care, 5 (1967), 201-220.

Klarman, H.E. "Some Technical Problems in Areawide Planning for Hospital Care," Journal of Chronic Diseases, 17 (1964), 735-747.

Klarman, H.E. "Approaches to Moderating the Increases in Medical Care Costs," Medical Care, 7 (1969), 47-52.

Kuehn, A.A. and Hamburger, M. "A Heuristic Program for Locating Warehouses," Management Science, 9 (1963), 643-66.

Kuhn, H.W. and Kuenne, R.E. "An Efficient Algorithm for the Numerical Solution of the Generalized Weber Problem in Spatial Economics," Journal of Regional Science, 4 (1962), 21-34.

Lakshmanan, T.R. and Hansen, W. "A Retail Market Potential Model," Journal of the American Institute of Planners, 31 (1966), 134-143.

Lehmer, D.H. "The Machine Tools of Combinatorics," Chapter 1 in: Beckenbach, C.F. (ed). Applied Combinatorical Analysis. New York: Wiley, (1964).

Lembcke, P.A. "Regional Organization of Hospitals," Annals of The American Academy of Political and Social Sciences, 273 (1951), 53-61.

Letourneau, C. "Regional Hospital Planning," Hospital Management, 99 (1965), 41-44.

Lord Hewelyn-Davies. "Facilities and Equipment for Health Services: Needed Research," Millbank Memorial Fund Quarterly, 44 (1966), 249-272.

Lubin, J.W. et. al. "How Distance Affects Physician Activity," Modern Hospital, 107 (1966) 80-82.

Marrinson, R. "Hospital Service Areas," Hospitals, 38 (1964), 52-54.

McGibony, J.R. and Block, L. "Better Patient Care Through Co-ordination," Public Health Reports, 63 (1949), 1499-1527.

McKeown, T. "The Concept of the District General Hospital," The Lancet, (1966), 869-871.

McNerney, W. Regionalization and Rural Health Care, Ann Arbor: University of Michigan, (1962).

McLachlan, G. "Health Services: Public Policy Issues," Medical Care, 5 (1967), 55-61.

Metropolitan St. Paul Hospital Planning Council. Physician Admitting Patterns in Metropolitan St. Paul, St. Paul: The Council, 1965.

Ministry of Health of Great Britain. Interim Report on the Future Provision of Medical and Allied Services, London: H.M.S.O. Paper 693, 1920.

Morrill, R. and Earickson, R. "Hospital Variation and Patient Travel Distance," Inquiry, 5 (1969), 26-34.

Morrill, R. et. al. "Factors Influencing Distances Travelled to Hospitals," Economic Geography, 46 (1970), 161-171.

Morrill, R. and Kelley, M. "The Simulation of Hospital Use and the Estimation of Locational Efficiency," Geographical Analysis, 2 (1970), 283-300.

Mott, F.D. and Roemer, M.J. Rural Health and Medical Care, New York: McGraw-Hill, (1948), 50-73.

Myers, R.S. "Areawide Planning for Hospitals Leads to Good Medical Services," The Modern Hospital, 97 (1961), 113-116.

Odoroff, M.E. and Abbe, L.M. "Use of General Hospitals; Demographic and Ecologic Factors," Public Health Reports, 72 (1957), 397-403.

Peterson, G.W. "Unfortunate Trends Emerging as Government Insurance Plans Gain Experience," Hospital Administration in Canada, 9 (1967), 32-33.

Revelle, C.S. and Swain, R.W. "Central Facilities Location," Geographical Analysis, 2 (1970), 30-42.

Roemer, M.I. "Bed Supply and Hospital Utilization: A Natural Experiment," Hospitals, 35 (1961), 36-42.

Roemer, M.I. and Anzel, D.M. "Health Needs and Services of the Rural Poor," Medical Care Review, 25 (1968), 377-390.

Rosenstock, I.M. "Why People Use Health Services," Milbank Memorial Fund Quarterly, 44 (1966), 94-127.

Rosenthal, G.D. The Demand for General Hospital Facilities, Chicago: The American Hospital Association, Monograph No. 14, (1964).

Rutenberg, D.P. Plant Location Under Economies of Scale in Production and Transportation, Pittsburg: Graduate School of Industrial Administration, Carnegie-Mellon University, (1968).

Sanazaro, P.J. "The Evaluation of Medical Care Under Public Law 89-239," Medical Care, 5 (1967), 162-168.

Schneider, J.B. "Measuring the Locational Efficiency of the Urban Hospital," Health Services Research, 2 (1967), 154-169.

Schneider J.B. "Measuring, Evaluating and Redesigning Hospital-Physician Patient Spatial Relationships in Metropolitan Areas," Inquiry, 5 (1968), 24-43.

Scott, A.J. "On the Optimal Partitioning of Spatially Distributed Point Sets," in: Scott, A.J. (ed.): Studies in Regional Science, London, Pion, (1969), 57-72.

Scott, A.J. "Location-Allocation Systems: A Review," Geographical Analysis, 2 (1970), 95-119.

Scott, A.J. Dynamic Location-Allocation Systems: Some Basic Planning Strategies, Department of Geography Discussion Paper No. 3, Toronto, University of Toronto, (1969).

Sigmond, R.M. "Health Planning," Medical Care, 5 (1967), 117-128.

Singer, S. Multi-Centers and Multi-Medians of a Graph with an Application to Optimal Warehouse Location, Mimeographed Paper Presented at the Annual Meeting of the Management Science and Operations Research Association, San Francisco, June (1968).

Somers, H.M. and Somers, A.R. Doctors, Patients, and Health Insurance, New York: Doubleday, (1962).

Stewart, W.H. Report on Regional Medical Programs to the President and the Congress, Washington: U.S. Department of Health, Education, and Welfare, (1967).

Suchman, E.A. Social Patterns of Illness and Medical Care," Journal of Health and Human Behaviour, 6 (1965), 2-16.

Task Force Report. The Cost of Health Services in Canada, Ottawa: The Queens Printer, 1970.

Teitz, M.B. "Towards a Theory of the Urban Public Facility Location," Papers of the Regional Science Association, 21 (1968), 35-51.

Teitz, M.B. and Bart, P. "Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph," Operations Research, 16 (1968), 955-968.

Wagner, H.M. Principles of Operations Research, Englewood Cliffs: Prentice-Hall Inc., 1969.

Wahn, E.V. "Regional Hospital Planning: Voluntary or Compulsory?" Hospital Administration in Canada, 11 (1969), 58-60.

Ward, J.H. "Hierarchical Grouping to Optimize an Objective Function," American Statistical Association Journal, 58 (1963), 236-244.

Weinerman, E.R. (ed.). Readings in Medical Care, Chapel Hill: University of North Carolina Press, (1951).

Weinerman, "Group Practice Revisted," Medical Care, 7 (1969), 173-174.

White, C.H. "Production-Allocation with Set-Up Penalties and Concave Material Costs," Operations Research, 18 (1970), 959-972.

White, K.L. "Patterns of Medical Practice," Chapter 43 in: Clark, W. and MacMahon, B. (ed.). Preventive Medicine, Boston: Little, Brown and Co., (1967).

World Health Organization. Methodology of Planning an Integrated Health Programme for Rural Areas, Technical Report Series No. 83, Geneva: W.H.O. (1954).

Yerby, A.S. "Health Departments, Hospitals and Health Services," Medical Care, 5 (1967), 70-74.

Zangwill, W.I. "Minimum Concave Cost Flows in Certain Networks," Management Science, 14 (1968), 429-450.

NODE POPULATIONS AND DEMANDS BY MEDICAL CATEGORIES

NODE	POPULATION	MEDICAL	SURGICAL	MATERNITY	NODE	POPULATION	MEDICAL	SURGICAL	MATERNITY
1	177	9	8	5	85	4200	383	291	53
2	3467	192	161	107	86	291	22	17	4
3	870	46	40	27	87	750	45	50	12
4	476	27	22	14	88	1395	62	77	22
5	638	36	31	18	89	26777	1047	1390	444
6	68	4	3	1	90	1693	70	90	27
7	370	28	22	9	91	607	22	30	9
8	300	24	19	6	92	1067	41	59	18
9	734	60	51	17	93	903	38	47	15
10	1529	127	110	33	94	538	24	27	9
11	492	39	34	10	95	514	20	26	8
12	129	9	9	2	96	2660	109	132	45
13	547	41	41	10	97	1479	65	77	24
14	260	18	20	5	98	550	27	31	9
15	100	6	8	1	99	459	23	26	8
16	167	9	13	3	100	400	26	22	6
17	307	15	29	5	101	14795	594	774	248
18	7691	345	737	134	102	479	20	24	7
19	200	8	18	3	103	400	30	19	6
20	300	12	28	5	104	2626	203	142	44
21	270	15	17	6	105	476	39	25	7
22	417	25	22	9	106	2314	203	171	37
23	200	10	13	4	107	306	23	23	4
24	9205	541	522	227	108	435	38	36	6
25	798	45	43	20	109	300	23	20	5
26	161	9	8	3	110	816	67	53	16
27	287	22	18	5	111	150	10	12	2
28	412	31	27	7	112	63	4	3	1
29	356	26	22	7	113	6850	320	318	166
30	3749	290	270	76	114	785	35	35	19
31	114	9	8	2	115	148	6	6	3
32	1706	143	128	29	116	108	5	4	2
33	280	16	18	4	117	155	7	7	3
34	6539	384	400	118	118	191	9	9	4
35	412	24	24	7	119	9141	883	631	166
36	300	18	17	5	120	400	30	22	6
37	487	25	26	8	121	179	13	10	3
38	710	41	41	10	122	862	53	50	15
39	777	48	51	14	123	1194	66	58	25
40	1511	85	93	25	124	358	19	17	7
41	325	20	20	6	125	300	14	15	6
42	636	38	39	12	126	549	25	26	10
43	473	29	28	8	127	371	16	19	8
44	586	33	31	11	128	33938	945	1389	718
45	10927	605	618	210	129	472	22	28	11
46	745	41	41	14	130	442	22	25	11
47	500	36	35	11	131	300	12	16	6
48	471	24	27	8	132	675	23	33	14
49	1589	78	95	31	133	450	15	21	10
50	584	27	36	11	134	300	10	14	6
51	3501	151	275	57	135	1835	61	90	62
52	14487	536	1164	300	136	200	10	13	5
53	250	12	18	5	137	267	13	16	6
54	1005	44	75	21	138	441	23	29	11
55	3710	171	261	87	139	700	39	48	18
56	5900	285	465	101	140	4500	252	312	125
57	6104	294	421	151	141	500	31	35	12
58	251	19	19	7	142	500	31	36	11
59	483	34	35	10	143	179	10	12	4
60	453	36	35	12	144	178	10	12	4
61	450	38	37	8	145	153	10	9	2
62	1199	93	94	23	146	180	12	12	2
63	500	23	33	9	147	2151	195	160	35
64	638	33	42	12	148	240	17	17	3
65	2131	108	140	42	149	201	13	12	3
66	500	24	33	11	150	73	5	5	1
67	495	27	27	8	151	82	5	4	1
68	4805	264	244	83	152	200	11	10	4
69	1553	53	88	28	153	3688	277	200	89
70	455	14	25	7	154	608	43	35	13
71	410	18	28	6	155	600	42	33	12
72	321	12	22	4	156	800	48	40	17
73	2906	190	192	48	157	1114	79	77	31
74	674	44	44	11	158	200	14	13	5
75	3895	248	260	70	159	100	7	6	2
76	641	42	45	10	160	400	28	27	12
77	250	14	16	3	161	100	7	6	2
78	200	12	14	2	162	250	20	19	8
79	1257	112	118	25	163	1764	143	134	50
80	731	56	55	7	164	924	74	69	26
81	300	14	16	6	165	1434	104	94	41
82	16410	894	1064	299	166	63	3	4	2
83	1073	58	60	19	167	100	6	7	3
84	200	17	13	2	168	100	6	4	2

APPENDIX I

SHORT DISTANCE MATRIX FOR MEDICAL DEMAND NETWORKS

NODE	2	4	10	18	24	30	32	34	37	38	40	42	45	49	51	52	56	57	61	62	65	68	69	73	75	76
1	15	48	94	167	168	222	242	234	292	299	329	307	312	285	303	309	315	296	328	307	277	330	307	283	271	254
2	1	33	79	152	153	207	227	219	277	284	314	292	297	270	288	294	300	311	341	322	292	335	322	298	286	269
3	15	48	94	167	168	222	242	234	292	299	329	307	312	285	303	309	315	312	344	323	293	346	323	299	287	270
4	33	1	112	185	186	240	260	252	310	317	347	325	330	303	321	327	333	344	374	355	325	368	355	331	319	302
5	23	56	56	129	130	184	204	196	254	261	291	269	274	247	265	271	277	292	318	323	315	312	335	321	309	292
6	31	64	48	121	122	176	196	188	246	253	283	261	266	239	257	263	269	284	310	315	323	304	327	329	317	300
7	39	72	40	113	114	168	188	180	238	245	275	253	258	231	249	255	261	276	302	307	331	296	319	337	325	308
8	55	88	24	97	98	152	172	164	222	229	259	237	242	215	233	239	245	260	286	291	321	280	303	353	341	324
9	67	100	12	85	86	140	160	152	210	217	247	225	230	203	221	227	233	248	274	279	309	268	291	343	353	336
10	79	112	1	73	74	128	148	140	198	205	235	213	218	191	209	215	221	236	262	267	297	256	279	331	343	348
11	84	117	5	68	69	123	143	135	193	200	230	208	213	186	204	210	216	231	257	262	292	251	274	326	338	353
12	99	132	20	93	94	148	168	160	218	225	255	233	238	211	229	235	241	256	282	287	317	276	299	351	363	368
13	96	129	17	56	57	111	131	123	181	188	218	196	201	174	192	198	204	219	245	250	280	239	262	314	326	343
14	108	141	29	42	43	97	117	109	167	174	204	182	187	160	178	184	190	205	231	236	266	225	248	300	312	329
15	114	147	35	36	37	91	111	103	161	168	198	176	181	154	172	178	184	199	225	230	260	219	242	294	306	323
16	122	155	43	55	57	111	131	123	181	188	218	196	201	174	192	193	204	213	245	250	280	239	262	314	326	343
17	133	166	54	17	18	72	92	84	142	149	179	157	162	135	153	159	165	180	206	211	241	200	223	275	287	304
18	150	183	71	1	19	78	98	85	143	150	180	158	163	136	154	160	166	181	207	212	242	201	224	276	288	305
19	161	194	82	11	30	89	109	96	154	161	191	169	174	147	165	171	177	192	218	223	253	212	235	287	299	316
20	160	193	81	10	9	68	88	75	133	140	170	148	153	126	144	150	156	171	197	202	232	191	214	266	278	295
21	146	179	67	27	8	59	79	74	132	139	169	147	152	125	143	149	155	170	196	201	231	190	213	265	277	294
22	167	200	88	40	21	38	58	87	145	152	182	160	165	138	156	162	168	183	209	214	244	203	226	278	290	307
23	139	172	60	11	12	71	91	78	136	143	173	151	156	129	147	153	159	174	200	205	235	194	217	269	281	298
24	151	184	72	19	1	59	79	66	124	131	161	139	144	117	135	141	147	162	188	193	223	182	205	257	269	286
25	163	196	84	31	12	71	91	78	136	143	173	151	156	129	147	153	159	174	200	205	235	194	217	269	281	298
26	170	203	91	38	19	78	98	47	105	112	142	120	125	98	116	122	128	143	169	174	204	163	186	238	250	267
27	202	235	123	75	56	35	55	122	180	187	217	195	200	173	191	197	203	218	244	249	279	238	261	313	325	342
28	186	219	107	59	40	19	39	106	164	171	201	179	184	157	175	181	187	202	228	233	263	222	245	297	309	326
29	175	208	96	48	29	30	50	95	153	160	190	168	173	146	164	170	176	191	217	222	252	211	234	286	298	315
30	205	238	126	78	59	1	20	125	183	190	220	198	203	176	194	200	206	221	247	252	282	241	264	316	328	345
31	213	246	134	86	67	8	12	133	191	198	228	206	211	184	202	208	214	229	255	260	290	249	272	324	336	353
32	225	258	146	98	79	20	1	145	203	210	240	218	223	196	214	220	226	241	267	272	302	261	284	336	348	365
33	193	226	114	61	42	101	121	24	82	89	119	97	102	75	93	99	105	120	146	151	181	140	163	215	227	244
34	217	250	138	85	66	125	145	1	58	65	95	73	78	51	69	75	81	96	122	127	157	116	139	191	203	220
35	233	266	154	101	82	141	161	16	42	49	79	57	78	67	85	91	97	104	122	111	141	132	155	207	219	235
36	249	282	170	117	98	157	177	32	26	33	63	41	62	83	101	107	113	88	106	95	125	148	171	223	235	252
37	275	308	196	143	124	183	203	58	1	7	37	15	36	63	83	83	89	62	80	69	99	124	147	199	211	228
38	282	315	203	150	131	190	210	65	7	1	44	22	43	70	90	90	96	69	87	76	106	131	154	206	218	235
39	343	376	264	211	192	251	271	126	68	75	31	53	74	101	121	121	127	100	84	63	93	162	185	237	249	266
40	312	345	233	180	161	220	240	95	37	44	1	22	43	70	90	90	96	69	53	32	62	131	154	206	218	235
41	303	336	224	171	152	211	231	86	28	35	9	13	34	61	81	81	87	60	62	41	71	122	145	197	209	226
42	290	323	211	158	139	198	218	73	15	22	22	1	21	48	68	68	74	47	65	54	84	109	132	184	196	213
43	297	330	218	165	146	205	225	80	22	29	29	7	14	41	61	61	67	40	58	61	91	102	125	177	189	206
44	301	334	222	169	150	209	229	84	30	37	37	15	6	33	53	53	59	32	50	69	99	94	117	169	181	198
45	295	328	216	163	144	203	223	78	36	43	43	21	1	27	47	47	53	26	44	65	95	88	111	163	175	192
46	303	336	224	171	152	211	231	86	44	51	51	29	8	35	45	39	45	18	36	57	87	80	103	155	167	184
47	309	342	230	177	158	217	237	92	50	57	57	35	14	41	39	33	39	12	30	51	81	74	97	149	161	178
48	283	316	204	151	132	191	211	66	48	55	55	33	12	15	35	41	47	38	56	77	107	82	105	157	169	186
49	268	301	189	136	117	176	196	51	63	70	70	48	27	1	20	26	32	47	71	92	122	67	90	142	154	171
50	259	292	180	127	109	167	187	42	72	79	79	57	36	9	27	33	39	54	80	101	131	74	97	149	161	178
51	266	319	207	154	135	194	214	69	83	90	90	68	47	20	1	6	12	27	69	90	120	47	70	122	134	151
52	292	325	213	160	141	200	220	75	83	90	90	68	47	26	6	1	6	21	63	84	114	41	64	116	128	145
53	299	332	220	167	148	207	227	82	76	83	83	61	40	33	13	7	13	14	56	77	107	48	71	123	135	152
54	302	335	223	170	151	210	230	85	73	80	80	58	37	36	16	10	16	11	53	74	104	51	74	126	138	155
55	310	343	231	178	159	218	238	93	65	72	72	50	29	44	24	18	24	3	45	66	96	59	82	134	146	163
56	298	331	219	166	147	206	226	81	89	96	96	74	53	32	12	6	1	27	69	90	120	35	58	110	122	139
57	311	344	234	181	162	221	241	96	62	69	69	47	26	47	27	21	27	1	42	63	93	62	85	137	149	166
58	317	350	238	185	166	225	245	100	58	65	65	43	22	49	47	41	47	20	38	59	89	82	105	157	169	186
59	316	349	237	184	165	224	244	99	57	64	64	42	21	48	46	40	46	19	23	44	74	81	104	156	168	185
60	326	359	247	194	175	234	254	109	67	74	66	52	31	58	56	50	56	29	13	34	64	91	114	166	178	195
61	333	366	260	207	188	247	267	122	80	87	53	65	44	71	69	63	69	42	1	21	51	104	127	179	191	208
62	312	345	265	212	193	252	272	127	69	76	32	54	65	92	90	84	90	63	21	1	30	125	148	200	212	229
63	254	287	291	238	219	278	298	153	119	126	100	104	83	104	84	78	84	57	89	68	38	119	142	194	204	187
64	274	307	311	258	239	298	318	173	117	124	80	102	103	124	104	98	104	77	69	48	18	139	162	214	224	207
65	282	315	295	242	223	282	302	157	99	106	62	84	95	122	112	106	112	85	51	30	1	147	170	222	232	215
66	294	327	251	198	179	238	258	113	79	86	86	64	43	64	44	38	44	17	59	80	78	79	102	154	166	183
67	320	353	241	188	169	228	248	103	111	118	118	96	75	54	34	28	22	49	91	112	142	13	36	88	100	117
68	307	340	254	201	182	241	261	116	124	131	131	109	88	67	47	41	35	62	104	125	155	1	23	75	87	104
69	284	317	277	224	205	264	284	139	147	154	154	132	111	90	70	64	58	85	127	148	178	23	1	52	64	81
70	265	298	296	243	224	283	303	158	166	173	173	151	130	109	89	83	77	104	146	167	197	42	19	33	45	62
71	258	291	313	260	241	300	320	175	183	190	190	168	147	126	106	100	94	121	163	184	202	59	36	16	28	45
72	229	262	308	279	260	319	339	194	202	209	209	187	166	145	125	119	113	140	182	203	173	78	55	63	51	34
73	242	275	321	276	257	316	336	191	199	206	206	184	163	142	122	116	110	137	179	200	186	75	52	1	12	29
74	237	270	316	281	262	321	341	196	204	211	211	189	168	147	127	121	115	142	184	205	181	80	57	5	7	24
75	230	263	309	288	269	328	348	203	211	218	218	196	175	154	134	128	122	149	191	204	174	87	64	12	1	17
76	213	246	292	305	286	345	365	220	228	235	219	213	192	171	151	145	139	166	208	187	157	104	81	29	17	1
77	251	284	330	287	268	327	347	202	210	217	217	195	174	153	133	127	121	148	190	211	195	86	63	11	23	38
78	238	271	317	300	281	340	360	215	223	230	230	208	187	166	146	140	134	161	203	212	182	99	76	24	36	25
79	232	265	311	306	287	346	366	221	229	236	236	214	193	172	152	146	140	167	209	206	176	105	82	30	36	19
80	250	283	329	324	305	364	384	239	247	254	254	232	211	190	170	164	158	185	227	224	194	123	100	48	54	37
81	224	257	303	314	295	354	374	229	237	244	230	222	201	180	160	154	148	175	217	198	168	113	90	38	28	11
82	204	237	283	301	285	344	364	219	227	234	210	212	191	170	150	144	138	165	199	178	148	103	80	38	26	9
83	215	248	294	315	296	355	375	230	238	245	221	223	202	181	161	155	149	176	210	189	159	114	91	49	37	20
84	199	232	278	309	290	349	369	224	224	231	205	209	188	175	155	149	143	162	194	173	143	108	85	43	31	14

APPENDIX II

BODZ	79	82	83	85	87	89	101	104	106	110	113	119	123	128	135	140	147	153	156	157	163	164	165	169	170	171
1	273	245	256	234	221	212	159	145	137	112	68	129	161	196	235	247	282	306	325	299	251	246	271	293	404	305
2	288	260	271	249	236	227	174	160	152	127	83	144	176	211	250	262	298	381	340	314	266	261	286	308	419	320
3	289	261	272	250	237	228	175	161	153	128	84	145	177	212	251	263	299	382	341	315	267	262	287	309	420	321
4	321	293	304	282	269	260	207	193	185	160	116	177	209	244	283	295	331	414	373	347	299	294	319	341	452	353
5	311	283	294	272	259	250	197	183	175	150	106	167	199	234	273	285	321	404	363	337	289	284	309	331	442	343
6	319	291	302	280	267	258	205	191	183	158	114	175	207	242	281	293	329	412	371	345	297	292	317	339	450	351
7	327	299	310	288	275	266	213	199	191	166	122	183	215	250	289	301	337	420	379	353	305	300	325	347	458	359
8	343	315	326	304	291	282	229	215	207	182	138	199	231	266	305	317	353	436	395	369	321	316	341	363	474	375
9	355	327	338	316	303	294	241	227	219	194	150	211	243	278	317	329	365	448	407	381	333	328	353	375	486	387
10	361	339	350	328	315	306	253	239	231	206	162	223	255	290	329	341	377	460	419	393	345	340	365	387	498	399
11	356	344	355	333	320	311	258	244	236	211	167	228	260	295	334	346	382	465	424	398	350	345	370	392	503	404
12	381	359	370	348	335	326	273	259	251	226	182	243	275	310	349	361	397	480	439	413	365	360	385	407	518	419
13	344	342	353	345	332	323	270	256	248	223	179	240	272	307	346	358	386	469	436	410	362	357	382	404	507	416
14	330	328	339	339	344	335	282	268	260	235	191	252	284	319	358	370	372	455	448	422	374	369	394	416	493	428
15	324	322	333	333	346	337	288	274	266	241	197	258	290	325	364	376	366	449	454	428	380	375	400	422	487	434
16	344	342	353	353	358	349	296	282	274	249	205	266	298	333	372	384	386	469	462	436	388	383	408	430	507	442
17	305	303	314	314	327	318	307	293	285	260	216	277	309	344	363	393	347	430	471	447	399	394	419	441	468	453
18	306	304	315	315	328	319	318	310	302	277	233	294	326	361	400	394	348	431	472	457	416	411	436	458	469	470
19	317	315	326	326	339	330	329	321	313	288	244	305	337	372	411	405	359	442	483	468	427	422	447	469	480	481
20	296	294	305	305	318	309	308	320	312	287	243	304	336	371	410	384	338	421	462	447	426	421	446	468	459	460
21	295	293	304	304	317	308	307	306	298	273	229	290	322	357	396	383	337	420	461	446	412	407	432	454	458	466
22	308	306	317	317	330	321	320	327	319	294	250	311	343	378	417	396	350	433	474	459	433	428	453	475	471	487
23	299	297	308	308	321	312	311	299	291	266	222	283	315	350	389	387	341	424	465	450	405	400	425	447	462	459
24	287	285	296	296	309	300	299	311	303	278	234	295	327	362	401	375	329	412	453	438	417	412	437	459	450	471
25	299	297	308	308	321	312	311	323	315	290	246	307	339	374	413	387	341	424	465	450	429	424	449	471	462	483
26	268	266	277	277	290	281	280	294	302	297	253	314	326	349	388	356	310	393	434	419	404	399	424	446	431	458
27	343	341	352	352	365	356	355	362	354	329	285	346	378	413	452	431	385	468	509	494	468	463	488	510	506	522
28	327	325	336	336	349	340	339	346	338	313	269	330	362	397	436	415	369	452	493	478	452	447	472	494	490	506
29	316	314	325	325	338	329	328	335	327	302	258	319	351	386	425	404	358	441	482	467	441	436	461	483	479	495
30	346	344	355	355	368	359	358	365	357	332	288	349	381	416	455	434	388	471	512	497	471	466	491	513	509	525
31	354	352	363	363	376	367	366	373	365	340	296	357	389	424	463	442	396	479	520	505	479	474	499	521	517	533
32	366	364	375	375	388	379	378	385	377	352	308	369	401	436	475	454	408	491	532	517	491	486	511	533	529	545
33	245	243	254	254	267	258	257	271	279	304	276	293	303	326	365	333	287	370	411	396	381	376	401	423	408	435
34	221	219	230	230	243	234	233	247	255	280	300	269	279	302	341	309	263	346	387	372	357	352	377	399	384	411
35	237	235	246	246	259	250	241	255	263	288	316	277	287	310	349	325	279	362	403	388	365	360	385	407	400	419
36	253	251	262	262	275	266	225	239	247	272	316	261	271	293	333	335	295	378	419	397	349	344	369	391	416	403
37	229	227	238	238	251	242	199	213	221	246	290	235	245	268	307	309	271	354	395	371	323	318	343	365	392	377
38	236	234	245	245	258	249	206	220	228	253	297	242	252	275	314	316	278	361	402	378	330	325	350	372	399	384
39	267	265	276	276	275	266	201	215	223	248	292	237	247	270	309	311	309	392	399	373	325	320	345	367	430	379
40	236	234	245	245	244	235	170	184	192	217	261	206	216	239	278	280	278	361	368	342	294	289	314	336	399	348
41	227	225	236	236	249	240	179	193	201	226	270	215	225	248	287	289	269	352	377	351	303	298	323	345	390	357
42	214	212	223	223	236	227	184	198	206	231	275	220	230	253	292	294	256	339	380	356	308	303	328	350	377	362
43	207	205	216	216	229	220	177	191	199	224	268	213	223	246	285	287	249	332	373	349	301	296	321	343	370	355
44	199	197	208	208	221	212	169	183	191	216	260	205	215	238	277	279	241	324	365	341	293	288	313	335	362	347
45	193	191	202	202	215	206	163	177	185	210	254	199	209	232	271	273	235	318	359	335	287	282	307	329	356	341
46	185	183	194	194	207	198	155	169	177	202	246	191	201	224	263	265	227	310	351	327	279	274	299	321	348	333
47	179	177	188	188	201	192	149	163	171	196	240	185	195	218	257	259	221	304	345	321	273	268	293	315	342	327
48	187	185	196	196	209	200	175	189	197	222	266	211	221	244	283	275	229	312	353	338	299	294	319	341	350	353
49	172	170	181	181	194	185	184	198	206	231	275	220	230	253	292	260	214	297	338	323	308	303	328	350	335	362
50	179	177	188	188	201	192	191	205	213	238	282	227	237	260	299	267	221	304	345	330	315	310	335	357	342	369
51	152	150	161	161	174	165	164	178	186	211	255	200	210	233	272	240	194	277	318	303	288	283	308	330	315	342
52	146	144	155	155	168	159	158	172	180	205	249	194	204	227	266	234	188	271	312	297	282	277	302	324	309	336
53	153	151	162	162	175	166	151	165	173	198	242	187	197	220	259	241	195	278	319	304	275	270	295	317	316	329
54	156	154	165	165	178	169	148	162	170	195	239	184	194	217	256	244	198	281	322	307	272	267	292	314	319	326
55	164	162	173	173	186	177	140	154	162	187	231	176	186	209	248	250	206	289	330	312	264	259	284	306	327	318
56	140	138	149	149	162	153	164	178	186	211	255	200	210	233	272	228	182	265	306	291	281	283	308	330	303	342
57	167	165	176	176	189	180	137	151	159	184	228	173	183	206	245	247	209	292	333	309	261	256	281	303	330	315
58	187	185	196	196	209	200	157	171	179	204	248	193	203	226	265	267	229	312	353	329	281	276	301	323	350	335
59	186	184	195	195	208	199	156	170	178	203	247	192	202	225	264	266	228	311	352	328	280	275	300	322	349	334
60	196	194	205	205	218	209	166	180	188	213	257	202	212	235	274	276	238	321	362	338	290	285	310	332	359	344
61	209	207	218	218	231	222	159	173	181	206	250	195	205	228	267	269	251	334	357	331	283	278	303	325	372	337
62	230	228	239	225	212	203	138	152	160	185	229	174	184	207	246	249	272	355	336	310	262	257	282	304	393	316
63	206	178	189	167	154	145	80	94	102	127	171	116	126	149	188	190	223	306	278	252	204	199	224	246	344	258
64	226	198	209	187	174	165	100	114	122	147	191	136	146	169	208	210	243	326	298	272	224	219	244	266	364	278
65	234	206	217	195	182	173	108	122	130	155	199	144	154	177	216	218	251	334	306	280	232	227	252	274	372	286
66	184	182	193	193	194	185	120	134	142	167	211	156	166	189	228	230	226	309	318	292	244	239	264	286	347	298
67	118	116	127	127	140	131	146	160	168	193	237	182	192	215	254	206	160	243	284	269	259	264	289	311	281	323
68	105	103	114	114	127	118	133	147	155	180	224	169	179	202	241	193	147	230	271	256	246	251	276	298	268	310
69	82	80	91	91	104	95	110	124	132	157	201	146	156	179	218	170	124	207	248	233	223	228	253	275	245	287
70	63	61	72	72	85	76	91	105	113	136	182	127	137	160	199	151	105	188	229	214	204	209	234	256	226	268
71	46	54	65	65	78	75	84	98	106	131	175	120	130	153	192	134	88	171	212	197	187	192	217	239	209	251
72	53	25	36	36	49	40	55	69	77	102	146	91	101	124	163	138	92	175	216	201	179	174	199	221	213	233
73	30	38	49	49	62	59	68	82	90	115	159	104	114	137	176	118	72	155	196	181	171	176	201	223	193	235
74	35	33	44	44	57	54	63	77	85	110	154	99	109	132	171	123	77	160	201	186	176	181	206	228	198	240
75	36	26	37	37	50	47	56	70	78	103	147	92	102	125	164	124	78	161	202	187	177	175	200	222	199	234
76	19	9	20	20	33	30	39	53	61	86	130	75	85	108	147	107	61	144	185	170	160	158	183	205	182	217
77	19	47	58	58	71	68	77	91	99	124	168	113	123	146	185	107	61	144	185	170	160	165	190	212	182	224
78	6	34	45	45	58	55	64	78	86	111	155	100	110	133	172	94	48	131	172	157	147	152	177	199	169	211
79	1	28	39	39	52	49	58	72	80	105	149	94	104	127	166	88	42	125	166	151	141	146	171	193	163	205
80	18	46	57	57	70	67	76	90	98	123	167	112	122	122	161	70	24	107	148	133	123	128	153	175	145	187
81	8	20	31	31	44	41	50	64	72	97	141	86	96	119	158	96	50	133	174	159	149	154	179	201	171	213
82	28	1	11	11	24	21	30	44	52	77	121	66	76	99	138	113	67	150	191	176	154	149	174	196	188	208
83	39	11	1	22	35	32	41	55	63	88	132	77	87	110	149	124	78	161	202	187	165	160	185	207	199	219
84	33	5	16	6	19	16	25	39	47	72	116	61	71	94	133	108	62	145	186	171	149	144	169	191	183	203

NODE	2	4	10	18	24	30	32	34	37	38	40	42	45	49	51	52	55	57	61	62	65	68	69	73	75	76
85	205	238	284	315	296	355	375	230	230	237	211	215	194	181	161	155	149	168	200	179	149	114	91	49	37	20
86	217	250	296	327	308	367	387	242	242	249	223	227	206	193	173	167	161	180	212	191	161	126	103	61	49	32
87	218	251	297	328	309	368	388	243	243	250	224	228	207	194	174	168	162	181	213	192	162	127	104	62	50	33
88	211	244	290	327	308	367	387	242	236	243	217	221	200	193	173	167	161	174	206	185	155	126	103	69	57	40
89	203	236	282	319	300	359	379	234	228	235	209	213	192	185	165	159	153	166	198	177	147	118	95	77	65	48
90	210	243	289	312	293	352	372	227	235	242	216	220	199	178	158	152	146	173	205	184	154	111	88	84	72	55
91	206	239	285	322	303	362	382	237	231	238	212	216	195	188	168	162	156	169	201	180	150	121	98	74	62	45
92	193	226	272	329	310	369	389	244	218	225	199	203	182	195	175	169	163	156	188	167	137	128	105	87	75	58
93	205	238	284	341	322	381	401	256	242	249	223	227	206	207	187	181	175	180	212	191	161	140	117	93	81	64
94	187	220	266	331	312	371	391	246	212	219	193	197	176	197	177	171	177	150	182	161	131	196	173	149	137	120
95	180	213	259	324	305	364	384	239	205	212	186	190	169	190	170	164	170	143	175	154	124	189	166	142	130	113
96	179	212	258	313	294	353	373	228	194	201	175	179	158	179	159	153	159	132	164	143	113	188	165	141	129	112
97	190	223	269	302	283	342	362	217	183	190	164	168	147	168	148	142	148	121	153	132	102	183	176	152	140	123
98	204	237	283	288	269	328	348	203	169	176	150	154	133	154	134	128	134	107	139	118	88	169	190	166	154	137
99	214	247	293	298	279	338	358	213	179	186	160	164	143	164	144	138	144	117	149	128	98	179	200	176	164	147
100	208	241	287	320	301	360	380	235	201	208	182	186	165	186	166	160	166	139	171	150	120	201	194	170	158	141
101	174	207	253	318	299	358	378	233	199	206	180	184	163	184	164	158	164	137	169	148	118	183	160	136	124	107
102	185	218	264	337	322	381	401	256	222	229	203	207	186	207	187	181	187	160	192	171	141	160	137	113	101	84
103	168	201	247	320	305	364	384	239	205	212	186	190	169	190	170	164	170	143	175	154	124	177	154	130	118	101
104	160	193	239	312	313	367	387	247	213	220	194	198	177	198	178	172	178	151	183	162	132	185	162	138	126	109
105	155	188	234	307	308	362	382	252	218	225	199	203	182	203	183	177	183	156	188	167	137	190	167	143	131	114
106	152	185	231	304	305	359	379	255	221	228	202	206	185	206	186	180	186	159	191	170	140	193	170	146	134	117
107	167	200	246	319	320	374	394	270	236	243	217	221	200	221	201	195	201	174	206	185	155	208	185	161	149	132
108	145	178	224	297	298	352	372	262	228	235	209	213	192	213	193	187	193	166	198	177	147	200	177	153	141	124
109	137	170	216	289	290	344	364	270	236	243	217	221	200	221	201	195	201	174	206	185	155	208	185	161	149	132
110	127	160	206	279	280	334	354	280	246	253	227	231	210	231	211	205	211	184	216	195	165	218	195	171	159	142
111	111	144	190	263	264	318	338	296	262	269	243	247	226	247	227	221	227	200	232	211	181	234	211	187	175	158
112	98	131	177	250	251	305	325	309	275	282	256	260	239	260	240	234	240	213	245	224	194	247	224	200	188	171
113	83	116	162	235	236	290	310	302	290	297	271	275	254	275	255	249	255	228	260	239	209	262	239	215	203	186
114	108	141	187	260	261	315	335	327	315	322	296	300	279	300	280	274	280	253	285	264	234	287	264	240	228	211
115	64	97	143	216	217	271	291	283	309	316	290	294	273	294	274	268	274	247	279	258	228	281	258	234	222	205
116	44	77	123	196	197	251	271	263	321	328	310	314	293	314	294	288	294	267	299	278	248	301	278	254	242	225
117	111	144	190	263	264	318	338	330	318	325	299	303	282	303	283	277	283	256	288	267	237	290	267	243	231	214
118	121	154	200	273	274	328	348	340	328	335	309	313	292	313	293	287	293	266	298	277	247	300	277	253	241	224
119	144	177	223	296	297	351	371	269	235	242	216	220	199	220	200	194	200	173	205	184	154	207	184	160	148	131
120	180	213	259	332	327	386	406	261	227	234	208	212	191	212	192	186	192	165	197	176	146	199	176	152	140	123
121	166	199	245	318	313	372	392	247	213	220	194	198	177	198	178	172	178	151	183	162	132	185	162	138	126	109
122	170	203	249	322	323	377	397	285	251	258	232	236	215	236	216	210	216	189	221	200	170	223	200	176	164	147
123	176	209	255	328	329	383	403	279	245	252	226	230	209	230	210	204	210	183	215	194	164	217	194	170	158	141
124	187	220	266	339	334	393	413	268	234	241	215	219	198	219	199	193	199	172	204	183	153	206	183	159	147	130
125	193	226	272	345	340	399	419	274	240	247	221	225	204	225	205	199	205	178	210	189	159	212	189	165	153	136
126	194	227	273	346	347	401	421	285	251	258	232	236	215	236	216	210	216	189	221	200	170	223	200	176	164	147
127	226	259	305	378	379	433	453	317	283	290	264	268	247	268	248	242	248	221	253	232	202	230	207	155	161	144
128	211	244	290	363	364	418	438	302	268	275	249	253	232	253	233	227	233	206	238	217	187	240	217	169	175	158
129	238	271	317	390	391	445	465	329	295	302	276	280	259	280	260	254	260	233	265	244	214	245	222	170	176	159
130	245	278	324	397	398	452	472	336	302	309	283	287	266	287	267	261	267	240	272	251	221	263	240	188	194	177
131	229	262	308	381	382	436	456	320	286	293	267	271	250	271	251	245	251	224	256	235	205	254	231	179	185	168
132	220	253	299	372	373	427	447	311	277	284	258	262	241	262	242	236	242	215	247	226	196	249	226	178	184	167
133	229	262	308	381	382	436	456	320	286	293	267	271	250	271	251	245	251	224	256	235	205	258	235	187	193	176
134	239	272	318	391	392	446	466	330	296	303	277	281	260	281	261	255	261	234	266	245	215	268	245	197	203	186
135	250	283	329	402	403	457	477	341	307	314	288	292	271	292	272	266	272	245	277	256	226	279	256	208	214	197
136	247	280	326	399	380	439	459	314	294	301	275	279	258	265	245	239	233	232	264	243	213	198	175	123	129	112
137	241	274	320	393	394	448	468	331	298	305	279	283	262	282	262	256	250	236	268	247	217	215	192	140	146	129
138	246	279	325	398	395	453	473	329	303	310	284	288	267	280	260	254	248	241	273	252	222	213	190	138	144	127
139	278	311	357	410	391	450	470	325	325	332	306	310	289	276	256	250	244	263	295	274	244	209	186	134	140	123
140	262	295	341	394	375	434	454	309	309	316	290	294	273	260	240	234	228	247	279	258	228	193	170	118	124	107
141	277	310	356	399	380	439	459	314	322	329	305	307	286	265	245	239	233	260	294	273	243	198	175	123	129	112
142	283	316	362	393	374	433	453	308	316	323	311	301	280	259	239	233	227	254	296	279	249	192	169	117	123	106
143	268	301	347	378	359	418	438	293	301	308	296	286	265	244	224	218	212	239	281	264	234	177	154	102	108	91
144	262	295	341	384	365	424	444	299	307	314	290	292	271	250	230	224	218	245	279	258	228	183	160	108	114	97
145	278	311	357	368	349	408	428	283	291	298	287	276	255	234	214	208	202	229	271	255	225	167	144	92	98	81
146	271	304	350	358	339	398	418	273	281	288	277	266	245	224	204	198	192	219	261	245	215	157	134	82	88	71
147	261	294	340	348	329	388	408	263	271	278	267	256	235	214	194	188	182	209	251	235	205	147	124	72	78	61
148	248	281	327	358	339	398	418	273	273	280	254	258	237	224	204	198	192	211	243	222	192	157	134	85	80	63
149	235	268	314	345	326	385	405	260	260	267	241	245	224	211	191	185	179	198	230	209	179	144	121	79	67	50
150	281	314	360	368	349	408	428	283	291	298	287	276	255	234	214	208	202	229	271	255	225	167	144	92	98	81
151	321	354	400	408	389	448	468	323	331	338	327	316	295	274	254	248	242	269	311	295	265	207	184	132	138	121
152	369	402	448	456	437	496	516	371	379	386	375	364	343	322	302	296	290	317	359	343	313	255	232	180	186	169
153	344	377	423	431	412	471	491	346	354	361	350	339	318	297	277	271	265	292	334	318	288	230	207	155	161	144
154	359	392	438	446	427	486	506	361	369	376	365	354	333	312	292	286	280	307	349	333	303	245	222	170	176	159
155	353	386	432	459	440	499	519	374	382	389	378	367	346	325	305	299	293	320	362	346	316	258	235	183	189	172
156	340	373	419	472	453	512	532	387	395	402	378	380	359	338	318	312	306	333	367	346	316	271	248	196	202	185
157	314	347	393	457	438	497	517	372	371	378	352	356	335	323	303	297	291	309	341	320	290	256	233	181	187	170
158	339	372	418	482	463	522	542	397	396	403	377	381	360	348	328	322	316	334	366	345	315	281	258	206	212	195
159	304	337	383	447	428	487	507	362	361	368	342	346	325	313	293	287	281	299	331	310	280	246	223	171	177	160
160	291	324	370	434	415	474	494	349	348	355	329	333	312	300	280	274	268	286	318	297	267	233	210	158	164	147
161	288	321	367	420	401	460	480	335	335	342	316	320	299	286	266	260	254	273	305	284	254	219	196	144	150	133
162	276	309	355	428	429	483	503	364	333	340	314	318	297	315	295	289	283	271	303	282	252	248	225	173	179	162
163	266	299	345	418	419	473	493	357	323	330	304	308	287	308	288	282	281	261	293	272	242	246	223	171	177	160
164	261	294	340	413	414	468	488	352	318	325	299	303	282	303	283	277	283	256	288	267	237	251	228	176	182	165
165	286	319	365	438	439	493	513	377	343	350	324	328	307	328	308	302	308	281	313	292	262	276	253	201	207	190
166	316	349	395	468	469	523	543	407	373	380	354	358	337	358	338	332	338	311	343	322	292	306	283	231	237	220
167	294	327	373	446	447	501	521	385	351	358	332	336	315	336	316	310	316	289	321	300	270	284	261	209	215	198
168	306	339	385	458	459	513	533	397	363	370	344	348	327	348	328	322	328	301	333	312	282	296	273	221	227	210

NODE	79	82	83	85	87	89	101	104	106	110	113	119	123	128	135	140	147	153	156	157	163	164	165	169	170	171
85	39	11	22	1	13	22	31	45	53	78	122	67	77	100	139	102	56	139	180	165	155	150	175	197	177	209
86	51	23	34	12	25	34	43	57	65	90	134	79	89	112	151	90	44	127	168	153	143	148	173	195	165	207
87	52	24	35	13	1	15	44	58	66	91	135	80	90	113	152	115	69	152	193	178	168	163	188	210	190	222
88	59	31	42	20	7	8	37	51	59	84	128	73	83	106	145	122	76	159	200	185	161	156	181	203	197	215
89	67	39	50	28	15	1	29	43	51	76	120	65	75	98	137	130	84	167	208	193	153	148	173	195	205	207
90	74	46	57	35	22	7	36	50	58	83	127	72	82	105	144	137	91	174	215	200	160	155	180	202	212	214
91	64	36	47	25	12	3	32	46	54	79	123	68	78	101	140	127	81	164	205	190	156	151	176	198	202	210
92	77	49	60	38	25	10	19	33	41	66	110	55	65	88	127	129	94	177	217	191	143	138	163	185	215	197
93	83	55	66	44	31	22	43	45	53	78	122	67	77	100	139	141	100	183	224	203	155	150	175	197	221	209
94	139	111	122	100	87	78	13	27	35	60	104	49	59	82	121	123	156	239	211	185	137	132	157	179	277	191
95	132	104	115	93	80	71	6	20	28	53	97	42	52	75	114	116	149	232	204	178	130	125	150	172	270	184
96	131	103	114	92	79	70	5	19	27	52	96	41	51	74	113	115	148	231	203	177	129	124	149	171	269	183
97	142	114	125	103	90	81	16	30	38	63	107	52	62	85	124	126	159	242	214	188	140	135	160	182	280	194
98	156	128	139	117	104	95	30	44	52	77	121	66	76	99	138	140	173	256	228	202	154	149	174	196	294	208
99	166	138	149	127	114	105	40	54	62	87	131	76	86	109	148	150	183	266	238	212	164	159	184	206	304	218
100	160	132	143	121	108	99	34	48	56	81	125	70	80	103	142	144	177	260	232	206	158	153	178	200	298	212
101	126	98	109	87	74	65	1	14	22	47	91	36	46	69	108	110	143	226	198	172	124	119	144	166	264	178
102	103	75	86	64	51	42	23	25	33	58	102	47	57	80	119	121	120	203	209	183	135	130	155	177	241	189
103	120	92	103	81	68	59	6	8	16	41	85	30	40	63	102	104	137	220	192	166	118	113	138	160	258	172
104	128	100	111	89	76	67	14	1	8	33	77	24	48	71	110	112	145	228	200	174	126	121	146	168	266	180
105	133	105	116	94	81	72	19	5	3	28	72	19	50	73	112	114	150	233	202	176	128	123	148	170	271	182
106	136	108	119	97	84	75	22	8	1	25	69	16	47	70	109	111	147	230	199	173	125	120	145	167	268	179
107	151	123	134	112	99	90	37	23	15	40	84	31	62	85	124	126	162	245	214	188	140	135	160	182	283	194
108	143	115	126	104	91	82	29	15	7	18	62	9	41	76	115	118	154	237	205	179	131	126	151	173	275	185
109	151	123	134	112	99	90	37	23	15	10	54	17	49	84	123	126	162	245	213	187	139	134	159	181	283	193
110	161	133	144	122	109	100	47	33	25	1	44	17	49	84	123	135	171	254	213	187	139	134	159	181	292	193
111	177	149	160	138	125	116	63	49	41	16	28	33	65	100	139	151	187	270	229	203	155	150	175	197	308	209
112	190	162	173	151	138	129	76	62	54	29	15	46	78	113	152	164	200	283	242	216	168	163	188	210	321	222
113	205	177	188	166	153	144	91	77	69	44	1	61	93	128	167	179	215	298	257	231	183	178	203	225	336	237
114	230	202	213	191	178	169	116	102	94	69	25	86	118	153	192	204	240	323	282	256	208	203	228	250	361	262
115	224	196	207	185	172	163	110	96	88	63	19	80	112	147	186	198	234	317	276	250	202	197	222	244	355	256
116	244	216	227	205	192	183	130	116	108	83	39	100	132	167	206	218	254	337	296	270	222	217	242	264	375	276
117	233	205	216	194	181	172	119	105	97	72	28	89	121	156	195	207	243	326	285	259	211	206	231	253	364	265
118	243	215	226	204	191	182	129	115	107	82	38	99	131	166	205	217	253	336	295	269	221	216	241	263	374	275
119	150	122	133	111	98	89	36	24	16	17	61	1	32	67	106	118	154	237	196	170	122	117	142	164	275	176
120	142	114	125	103	90	81	28	30	29	53	97	36	42	41	80	82	118	201	170	144	96	91	116	138	239	150
121	128	100	111	89	76	67	14	16	15	39	83	22	32	55	94	96	132	215	184	158	110	105	130	152	253	164
122	166	138	149	127	114	105	52	50	42	43	87	26	6	41	80	92	138	211	170	144	96	91	116	138	249	150
123	160	132	143	121	108	99	46	48	47	49	93	32	1	35	74	85	132	205	164	138	90	85	110	132	243	144
124	149	121	132	110	97	88	35	37	36	60	104	43	35	34	73	75	111	194	163	137	89	84	109	131	232	143
125	155	127	138	116	103	94	41	43	42	66	110	49	29	28	67	79	117	198	157	131	83	78	103	125	236	137
126	156	136	149	127	114	105	52	54	53	67	111	50	18	17	56	68	114	187	146	120	72	67	92	114	225	126
127	125	150	161	139	146	137	84	86	85	99	143	82	50	15	54	47	83	166	136	110	70	65	90	112	204	124
128	139	155	166	144	131	122	69	71	70	84	128	67	35	1	39	51	97	170	129	103	55	50	75	97	208	109
129	140	165	176	154	158	149	96	98	97	111	155	94	62	27	66	52	98	143	102	76	28	23	48	70	181	82
130	158	183	194	172	165	156	103	105	104	118	162	101	69	34	73	70	116	153	112	86	38	33	58	80	191	92
131	149	173	184	162	149	140	87	89	88	102	146	85	53	18	57	61	107	152	111	85	37	32	57	79	190	91
132	148	164	175	153	140	131	78	80	79	93	137	76	44	9	30	60	106	179	138	112	64	59	84	106	217	118
133	157	173	184	162	149	140	87	89	88	102	146	85	53	18	21	69	115	188	147	121	73	68	93	115	226	127
134	167	183	194	172	159	150	97	99	98	112	156	95	63	28	11	79	125	198	157	131	83	78	103	125	236	137
135	178	194	205	183	170	161	108	110	109	123	167	106	74	39	1	90	136	209	168	142	94	89	114	136	247	148
136	93	118	129	107	120	129	95	97	96	120	164	103	82	47	86	15	51	134	104	78	68	73	98	120	172	132
137	110	135	146	124	137	146	99	101	100	114	158	97	65	30	69	32	68	151	121	95	85	80	105	127	189	139
138	108	133	144	122	135	144	104	106	105	119	163	102	70	35	74	20	66	148	107	81	33	38	63	85	186	97
139	104	129	140	118	131	140	126	128	127	151	195	134	106	71	110	16	62	114	73	47	49	54	79	101	152	113
140	88	113	124	102	115	124	110	112	111	135	179	118	90	55	94	1	46	129	89	63	53	58	83	105	167	117
141	93	118	129	107	120	129	125	127	126	150	194	133	107	72	111	17	51	134	106	80	70	75	100	122	172	134
142	87	112	123	101	114	123	131	133	132	156	200	139	115	80	119	25	45	128	114	88	78	83	108	130	166	142
143	72	97	108	86	99	108	116	118	117	141	185	124	103	68	107	16	30	113	105	79	69	74	99	121	151	133
144	78	103	114	92	105	114	110	112	111	135	179	118	97	62	101	10	36	119	99	73	63	68	93	115	157	127
145	62	87	98	76	89	98	107	121	127	151	195	134	113	78	117	26	20	103	115	89	79	84	109	131	141	143
146	52	77	88	66	79	88	97	111	119	144	188	133	123	88	127	36	10	93	125	99	89	94	119	141	131	153
147	42	67	78	56	69	78	87	101	109	134	178	123	133	98	137	46	1	83	124	109	99	104	129	151	121	163
148	55	54	65	43	56	65	74	88	96	121	165	110	120	111	150	59	13	96	137	122	112	117	142	164	134	176
149	68	41	52	30	43	52	61	75	83	108	152	97	107	124	163	72	26	109	150	135	125	130	155	177	147	189
150	62	87	98	76	89	98	107	121	129	154	198	143	153	118	157	66	20	63	104	129	119	124	149	171	101	183
151	102	127	138	116	129	138	147	161	169	194	238	183	193	158	197	106	60	23	64	90	138	143	168	151	61	202
152	150	175	186	164	177	186	195	209	217	242	286	231	230	195	234	154	108	25	66	92	140	145	170	153	63	204
153	125	150	161	139	152	161	170	184	192	217	261	206	205	170	209	129	83	1	41	67	115	120	145	128	38	179
154	140	165	176	154	167	176	185	199	207	232	276	221	190	155	194	115	98	15	26	52	100	105	130	113	53	164
155	153	178	189	167	180	189	198	212	212	226	270	209	177	142	181	102	111	28	13	39	87	92	117	100	66	151
156	166	191	202	180	193	202	198	200	199	213	257	196	164	129	168	89	124	41	1	26	74	79	104	87	79	138
157	151	176	187	165	178	187	172	174	173	187	231	170	138	103	142	63	109	67	26	1	48	53	78	61	105	112
158	176	201	212	190	203	212	197	199	198	212	256	195	163	128	167	88	134	92	51	25	73	78	81	36	130	115
159	141	166	177	155	168	177	162	164	163	177	221	160	128	93	132	53	99	77	36	10	38	43	68	71	115	102
160	128	153	164	142	155	164	149	151	150	164	206	147	115	80	119	40	86	90	49	23	25	30	55	77	128	89
161	114	139	150	128	141	150	136	138	137	161	205	144	116	81	120	26	72	104	63	37	39	44	69	91	142	103
162	143	168	179	157	170	179	134	136	135	149	193	132	100	65	104	55	101	105	64	38	10	15	40	62	143	74
163	141	166	177	155	168	177	124	126	125	139	183	122	90	55	94	53	99	115	74	48	1	5	30	52	153	64
164	146	171	182	160	173	172	119	121	120	134	178	117	85	50	89	58	104	120	79	53	5	1	25	47	158	59
165	171	196	207	185	198	197	144	146	145	159	203	142	110	75	114	83	129	145	104	78	30	25	1	45	183	34
166	201	226	237	215	228	227	174	176	175	189	233	172	140	105	144	113	159	175	134	108	60	55	30	75	213	48
167	179	204	215	193	206	205	152	154	153	167	211	150	118	83	122	91	137	153	112	86	38	33	8	53	191	26
168	191	216	227	205	218	217	164	166	165	179	223	162	130	95	134	103	149	165	124	98	50	45	20	65	203	14

APPENDIX III

A Computer Program for Calculating all or Some of the Shortest Distances Through a Network (the Dantzig Algorithm)

```
C       ***************************************************************
C  ***  THIS PROGRAMME COMPUTES SHORTEST PATHS THROUGH A NETWORK. FOUR
C  ***  OPTIONS ARE POSSIBLE
C  ***  A). ALL ORIGINS TO ALL DESTINATIONS
C  ***  B). ALL ORIGINS TO SOME DESTINATIONS
C  ***  C). SOME ORIGINS TO ALL DESTINATIONS
C  ***  D). SOME ORIGINS TO SOME DESTINATIONS
C  ***  N.B. SHORTEST PATHS TO ALL DESTINATIONS ARE IN FACT COMPUTED BUT
C  ***  PRINTING IS SUPPRESSED FOR THOSE DESTINATIONS NOT SPECIFIED
C  ***  PUNCHED OUTPUT IS OPTIONED (IN FORMAT 8F10.2)
C  ***  THE DANTZIG ALGORITHM IS USED
C       ***************************************************************
C  ***  N.B. NETWORK NODES SHOULD BE LABELLED IN SEQUENCE 1,...,N
C       *****DECK SET UP AS FOLLOWS*****
C  ***  1), PROBLEM NAME CARD MAXIMUM OF 72 COLUMNS
C  ***  2), PROBLEM PARAMETER CARD
C  ***    A), COLS 1-3  NUMBER OF NODES IN NETWORK (N)
C  ***    B). COLS 4-6  NUMBER OF ORIGINS FROM WHICH SHORTEST PATHS ARE TO
C  ***                  BE COMPUTED (NN)
C  ***    C). COLS 7-9  NUMBER OF DESTINATIONS TO WHICH SHORTEST PATHS ARE
C  ***                  TO BE COMPUTED (MM)
C  ***    D), COL 10    PUNCH 1 FOR (A) SHORTEST PATH NETWORK SUMMARY TABLE
C  ***                  AND (B) FOR SHORTEST PATH ROUTINGS. PUNCH 0 FOR
C  ***                  OPTION (A) ONLY
C  ***    E), COL 11    PUNCH 1 IF PUNCHED OUTPUT IS OPTIONED
C  ***    F). COL 15    SINCE AS MANY JOBS AS ARE REQUIRED MAY BE RUN
C  ***                  CONSECUTIVELY PUNCH IN 1 IF THIS IS THE LAST (OR
C  ***                  ONLY) JOB IN THE RUN
C  ***  3). READ IN CARD(S) CONTAINING THE LIST OF NODES FROM WHICH SHORT
C  ***       DISTANCES ARE REQUIRED. 4 COLS PER NODE. IF THE TOTAL NUMBER
C  ***       OF SUCH ORIGINS IS EQUAL TO THE TOTAL NUMBER OF NODES IN THE
C  ***       NETWORK (I.E.IF N=NN) THEN IGNORE THIS INSTRUCTION
C  ***  4). READ IN CARD(S) CONTAINING THE LIST OF DESTINATIONS TO WHICH
C  ***       SHORTEST DISTANCES ARE REQUIRED. 4 COLS PER NODE. IF THE TOTAL
C  ***       NUMBER OF SUCH DESTINATIONS IS EQUAL TO THE TOTAL NUMBER OF
C  ***       NODES IN THE NETWORK (I.E.IF N=MM) THEN IGNORE THIS
C  ***       INSTRUCTION
C  ***  5). READ IN CARD(S) CONTAINING THE DISTANCE EACH NODE (FROM WHICH
C  ***       THE SHORTEST PATHS TO OTHER NODES ARE REQUIRED) IS FROM
C  ***       ITSELF.4 COLS PER NODE THE LAST COLUMN OF WHICH IS A DECIMAL
C  ***       FRACTION GREATER THAN OR EQUAL TO ZERO. E.G. IF THE DISTANCE
C  ***       IS ZERO THEN PUNCH 0000 IF THE DISTANCE IS 1 THEN PUNCH 0010
C  ***       OR IF THE DISTANCE IS 1.5 THEN PUNCH 0015
C  ***  6). VARIABLE FORMAT CARD FOR READING IN DISTANCES FROM EACH NODE
C  ***       TO ALL ADJOINING NODES. E.G.(5F5.0) MEANS A MAXIMUM OF 5 SUCH
C  ***       VALUES (OR LESS IF THERE ARE FEWER CONNECTED NODES) PER CARD
C  ***       OF 5 DIGITS EACH
C  ***  7). READ IN A VECTOR CONTAINING THE NUMBER OF NODES EACH
C  ***       INDIVIDUAL NODE IN THE NETWORK IS CONNECTED TO. N.B.A NODE IS
C  ***       NOT CONSIDERED TO BE CONNECTED TO ITSELF.4 COLUMNS FOR EACH
C  ***       VALUE. READ IN ALL VALUES (20 PER CARD) IN SEQUENCE I =1,N
```

```
C *** 8). READ IN DATA CARDS AS FOLLOWS;
C ***     FOR EACH NODE I IN THE NETWORK IN REGULAR SEQUENCE BEGINNING
C ***     WITH NODE NUMBER 1
C ***     A), READ IN CARD(S) CONTAINING A LIST OF ALL THE NODES TO
C ***     WHICH NODE I IS CONNECTED. 4 COLUMNS PER NODE. N.B. A NODE IS
C ***     NOT CONNECTED TO ITSELF
C ***     B). READ IN CARD(S) UNDER THE SPECIFIED VARIABLE FORMAT CARD
C ***     WHICH CONTAIN THE DISTANCES FROM NODE I TO ALL NODES SPECIFIED
C ***     IN 8(A)
C ***     REPEAT INSTRUCTIONS 8(A) AND 8(B) IN SEQUENCE FOR ALL NODES
C ***     I =1,...,N
C       ***************************************************************
C ***
C *** PROGRAMMER - R.A.WHITAKER - DEPARTMENT OF GEOGRAPHY,UNIVERSITY OF
C *** BRITISH COLUMBIA, JANUARY 1970
C ***
C       ***************************************************************
      DIMENSION STORE(1000),LABEL(1000),NORIG(1000),MDEST(1000)
      DIMENSION KS(1000),NS(1000),JS(1000),LS(1000),ID(1000),MS(1000)
      DIMENSION MD(1000),MIN(4000),IS(1000),IJ(1000),DS(1000),PROB(18)
      DIMENSION FMT(20),NVEC(1000),DIST(1000)
      EQUIVALENCE (STORE(1),MIN(1))
      REAL LABEL
C ******** READ IN PROBLEM TITLE AND CONTROL CARD
2     FORMAT (3I3,2I1,I2,I2)
3     FORMAT (20I4)
11    FORMAT (18A4)
13    FORMAT (1H1,16X,17HPROBLEM NAME.....,18A4//)
14    FORMAT (7X,22HNO.OF NODES IN NETWORK,I5/6X,23HNO.OF ORIGINS SPECIF
     1IED,I5/28HNO.OF DESTINATIONS SPECIFIED,I5/)
15    FORMAT (20A4)
17    FORMAT (20F4.1)
90    FORMAT (10X,31HDISCONNECTION IN NETWORK SYSTEM/10X,22HCURRENT ORIG
     1IN IS NODE,I5/10X,29HNODES LINKED BEFORE ERROR ARE,20I4/(46X,20I4)
     2)
135   FORMAT (1X,6HORIGIN,3X,11HDESTINATION,3X,8HDISTANCE,7X,14HROUTED T
     1HROUGH/)
140   FORMAT (3X,I4,6X,I4,7X,F8.2,4X,22I4/(40X,22I4))
147   FORMAT (1H )
155   FORMAT (/3X,4HNODE,9X,27HSHORTEST PATH NETWORK TABLE/)
160   FORMAT (7X,12I10)
170   FORMAT (I7,12F10.2/(7X,12F10.2))
175   FORMAT (15F6.0)
1     READ (5,11) (PROB(I),I =1,18)
      READ (5,2) N,NN,MM,KSW,KJV,NT,LFIN
      WRITE (6,13) (PROB(I),I =1,18)
      WRITE (6,14) N,NN,MM
      IF (NN.EQ.N) GO TO 12
C ******** READ IN VECTORS OF ORIGINS AND DESTINATIONS IF OPTIONED
      READ (5,3) (NORIG(I),I =1,NN)
      GO TO 5
```

```
12      DO 4 I =1,N
4       NORIG(I) =I
5       IF (MM.EQ.N) GO TO 8
        DO 6 I =1,N
6       MDEST(I) =0
        READ (5,3) (MD(I),I =1,MM)
        DO 7 I =1,MM
        J =MD(I)
7       MDEST(J) =J
        GO TO 10
8       DO 9 I =1,N
        MD(I) =I
9       MDEST(I) =I
10      JJ =0
        READ (5,17) (DIST(I),I =1,NN)
        READ (5,15) (FMT(I),I =1,20)
        READ (5,3) (NVEC(I),I =1,N)
C ******** READ IN ADJOINING NODES AND LINK DISTANCES
        DO 40 I =1,N
        L =JJ
        NT =NVEC(I)
        READ (5,3) (KS(J),J =1,NT)
        READ (5,FMT) (STORE(J),J =1,NT)
        DO 20 J =1,NT
        IF (KS(J).EQ.0) GO TO 25
        JJ =JJ+1
        IS(JJ) =I
        IJ(JJ) =KS(J)
20      DS(JJ) =STORE(J)
C ******** ARRANGE EACH NODES LINKED NEIGHBOURS BY ASCENDING DISTANCE
C ******** ORDER
25      L =L+1
        IF (L.EQ.JJ) GO TO 40
        LJ =JJ-1
27      IND =1
        DO 30 J =L,LJ
        IF (DS(J+1).GE.DS(J)) GO TO 30
        TEMP =DS(J+1)
        NR =IJ(J+1)
        DS(J+1) =DS(J)
        IJ(J+1) =IJ(J)
        DS(J) =TEMP
        IJ(J) =NR
        IND =J
30      CONTINUE
        IF (IND-1)27,40,27
40      MS(I) =JJ
        LV =0
        DO 200 II =1,NN
C ******** INITIALIZE ARRAYS
        K =0
```

```
      DO 45 I =1,N
      LS(I) =MS(I)-K
      K =MS(I)
      ID(I) =0
45    LABEL(I) =10.0**10
      R =10.0**10
      KK =NORIG(II)
      KR =KK
      ID(KK) =0
      LABEL(KK) =0.0
      DO 50 I =1,JJ
50    MIN(I) =0
      NK =N-1
C ******** COMPUTE SHORTEST PATH FOR SOME NODE
      DO 100 IN =1,NK
      DO 65 I =1,JJ
      IF (IJ(I).NE.KK) GO TO 65
      MIN(I) =-1
      LJ =IS(I)
      LS(LJ) =LS(LJ)-1
65    CONTINUE
      TMIN =10.0**10
      KV =0
      LL =0
      DO 80 IK =1,N
      L =LL+1
      LL =MS(IK)
      IF (LABEL(IK).EQ.R) GO TO 80
      IF (LS(IK).EQ.0) GO TO 80
      DO 70 I =L,LL
      MJ =IS(I)
      NJ =IJ(I)
      IF (MIN(I).NE.0) GO TO 70
      RMIN =LABEL(MJ)+DS(I)
      IF (RMIN.GE.TMIN) GO TO 70
      JR =I
      TMIN =RMIN
      KK =NJ
      KV =MJ
      GO TO 80
70    CONTINUE
80    CONTINUE
      IF (KV.NE.0) GO TO 95
C ******** CHECK FOR DISCONNECTION
      KX =0
      DO 85 IV =1,N
      IF (ID(IV).EQ.0) GO TO 85
      KX =KX+1
      MIN(KX) =ID(IV)
85    CONTINUE
      IF (KX.NE.0) GO TO 87
```

```
      KX =1
      MIN(KX) =KR
87    WRITE (6,90) KR,(MIN(IV),IV =1,KX)
      GO TO 205
95    LABEL(KK) =TMIN
      NS(IN) =KV
      JS(IN) =KK
      ID(KK) =IN
      MIN(JR) =1
100   CONTINUE
      KV =0
      IF (KSW.EQ.0) GO TO 150
C ******** TRACE SHORTEST PATH ROUTING
      DO 145 JX =1,N
      IF (MDEST(JX).EQ.0) GO TO 145
      K =ID(JX)
      IF (K.EQ.0) GO TO 145
      KS(1) =JS(K)
      KS(2) =NS(K)
      KK =2
      LL =K-1
      IF (LL.EQ.0) GO TO 125
      DO 120 I =1,LL
      K =K-1
      IF (KS(KK).NE.JS(K)) GO TO 120
      KK =KK+1
      KS(KK) =NS(K)
120   CONTINUE
C ******** REARRANGE ROUTINGS IN INCREASING DISTANCE FROM ORIGIN
125   J =KK+1
      DO 130 I =1,KK
      J =J-1
130   MIN(I) =KS(J)
      IF (KV.EQ.1) GO TO 133
      KV =1
C ******** PRINT OUT HEADINGS
      WRITE (6,147)
      WRITE (6,135)
133   WRITE (6,140) KR,JX,LABEL(JX),(MIN(J),J =1,KK)
145   CONTINUE
150   IF (KSW.NE.0) GO TO 153
      IF (LV.NE.0) GO TO 162
      LV =1
153   WRITE (6,155)
      WRITE (6,160) (MD(I),I =1,MM)
      WRITE (6,147)
162   J =0
C ******** PRINT SHORTEST PATHS FROM ORIGINS TO SPECIFIED DESTINATIONS
      DO 165 I =1,N
      IF (MDEST(I).EQ.0) GO TO 165
      J =J+1
```

```
      STORE(J) =LABEL(I)
      IF (MDEST(I).EQ.NORIG(II)) STORE(J) =DIST(II)
165   CONTINUE
      WRITE (6,170) KR,(STORE(I),I =1,J)
      IF (KJV.EQ.0) GO TO 200
      WRITE (7,175) (STORE(I),I =1,J)
200   CONTINUE
      IF (LFIN)205,1,205
205   STOP
      END
```

APPENDIX IV

A Computer Program for Ward's Hierarchical Grouping Algorithm with Dendogram

```
C       ***********************************************************************
C   *** PROGRAM HIERARCHICAL GROUPING
C   *** THIS PROGRAM PERFORMS A HIERARCHICAL GROUPING ANALYSIS ACCORDING
C   *** TO A CRITERION (MINIMIZATION OF INCREMENT TO WITHIN GROUPS
C   *** VARIATION AT EACH STEP) GIVEN BY J.H.WARD. SEE AMER STAT ASSOC J.
C   *** PAGES 236-244. IT IS ADAPTED FROM A BASIC CODING BY D.J.VELDMAN IN
C   *** "FORTRAN PROGRAMMING FOR THE BEHAVIOURAL SCIENCES" CHANGES ARE AS
C   *** FOLLOWS
C   *** 1), SINGLE DIMENSIONED ARRAYS ARE USED TO DOUBLE THE SIZE OF
C   *** PROBLEM WHICH CAN BE HANDLED ON ANY MACHINE. ACCESSING THE I,J
C   *** VALUES IN MATRIX A MAY BE FRACTIONALLY INCREASED AS A CONSEQUENCE.
C   *** 2), COMPUTING TIME FOR LARGE PROBLEMS IS APPROXIMATELY HALVED.
C   *** THIS IS ACHEIVED BY A), SHRINKING AT EACH STEP THE DOUBLE LOOP
C   *** SEARCH IN WHICH THE NEXT GROUP IS JOINED AND B), BY ELIMINATING
C   *** THE DOUBLE LOOP IN WHICH THE CURRENT MEMBERS OF EACH GROUP ARE
C   *** FOUND, STORED, AND PRINTED.
C   *** 3), A CONTIGUITY CONSTRAINT IS INCLUDED.TWO GROUPS ARE CONSIDERED
C   *** CONTIGUOUS IF ANY MEMBER OF ONE GROUP IS CONTIGUOUS TO ANY MEMBER
C   *** OF ANOTHER GROUP
C   *** 4), A DENDROGRAM IS PRINTED ON THE LINE PRINTER IF OPTIONED
C       ***********************************************************************
C       *****DECK SET UP AS FOLLOWS*****
C   *** 1), PROBLEM OR TITLE CARD.MAXIMUM OF 80 COLUMNS ALLOWED
C   *** 2), PROBLEM PARAMETER CARD
C   ***   COLS 1-3 NUMBER OF SUBJECTS
C   ***   COLS 4-6 NUMBER OF VARIABLES
C   ***   COLS 7-9 LEVEL OF GROUPING TO BEGIN GROUP MEMBERSHIP PRINTING.
C   ***   IF LEFT BLANK A COMPLETE STEPWISE GROUPING IS PRINTED FOR
C   ***   ALL N-1 STEPS - N.B. SINGLE MEMBER GROUPS ARE NOT PRINTED
C   ***   COL 10 PUNCH 1 IF STANDARDIZATION OF VARIABLES IS OPTIONED
C   ***   COL 11 PUNCH 1 IF TREE GRAPH IS OPTIONED.MAX SIZE IS ARBITRARILY
C   ***   SET EQUAL TO 200
C   ***   COL 12 PUNCH 1 IF CONTIGUITY CONSTRAINT IS IMPOSED
C   ***   COLS 13-15 MAXIMUM NUMBER OF CONTIGUITIES POSSESSED BY ANY
C   ***   SUBJECT.MAY BE LEFT BLANK IF 20 OR UNDER
C   ***   COL 16 PUNCH 1 IF IDENTIFICATION NUMBERS ARE TO BE READ IN.
C   ***   THIS WILL BE NECESSARY IF THE LABELS OF THE N SUBJECTS DO NOT
C   ***   RUN CONSECUTIVELY FROM 1,...,N
C   ***   COL 20 SINCE AS MANY DIFFERENT JOBS AS ARE REQUIRED MAY BE RUN
C   ***   CONSECUTIVELY PUNCH 1 IN COLUMN 20 FOR THE LAST (OR ONLY) JOB IN
C   ***   THE RUN
C   *** 3), IDENTIFICATION NUMBERS OF ALL N SUBJECTS IF OPTIONED - 4 COLS
C   *** PER SUBJECT
C   *** 4), VARIABLE FORMAT CARD FOR DATA E.G. (2F6.2,3X,4F10.5)
C   *** 5), DATA CARDS--THESE MUST BE READ IN IN VARIABLE ORDER I.E. READ
C   *** IN AS AN N BY M MATRIX WHERE THE N ROWS ARE VARIABLES AND THE M
C   *** COLUMNS ARE SUBJECTS                          SUBJECTS
C   ***                                              ---------------
C   ***                                              I I I I I I I I
C   ***                                   VARIABLES  I I I I I I I I
C   ***                                              I I I I I I I I
```

```
C ***                                                  ---------------
C *** 6), CONTIGUITY MATRIX. READ IN ONE ROW AT A TIME
C *** PUNCH IN THE NUMBERS (LABELS) OF THOSE SUBJECTS TO WHICH EACH
C *** SUBJECT IS CONTIGUOUS. 4 COLS PER NUMBER. E.G IF SUBJECT NO 1
C *** IS CONTIGUOUS TO SUBJECTS 4 AND 7 THEN PUNCH IN 4 AND 7. IF THE
C *** MAXIMUM NUMBER OF SUBJECTS ANY SUBJECT IS CONTIGUOUS TO EXCEEDS 2
C *** THEN TWO OR MORE CARDS WILL BE REQUIRED FOR EACH SUBJECT
C     ******************* DIMENSIONING INFORMATION *******************
C *** IF N EQUALS MAXIMUM SIZE OF PROBLEM WHICH CAN BE HANDLED ON ANY
C *** PROCESSOR (199 FOR 32K STORAGE) THEN DIMENSION SIZE OF
C *** ARRAY A MUST BE SET EQUAL TO AT LEAST ((N*N)+N)/2
C *** DIMENSION SIZE OF ARRAY KHOLD MUST BE AT LEAST N*2
C *** DIMENSION SIZE OF ALL OTHER ARRAYS MUST BE EQUAL TO N, EXCEPT FOR
C *** ARRAYS FMT AND TITL WHICH REMAIN AS DEFINED
C     ******************************************************************
C ***
C *** PROGRAMMER - R.A.WHITAKER - DEPARTMENT OF GEOGRAPHY,UNIVERSITY OF
C *** BRITISH COLUMBIA, APRIL 1969
C ***
C     ******************************************************************
      DIMENSION D(200),KG(200),W(200),LC(200),CONT(200)
      DIMENSION LAM(200),NC(200),MC(200),A(20000)
      DIMENSION KHOLD(400),KSTORE(200),FMT(20),TITL(20),ID(200)
      EQUIVALENCE (CONT(1),D(1))
1     FORMAT (3I3,3I1,I3,I1,I4)
6     FORMAT (20A4)
9     FORMAT (1X,19HPROBLEM NAME ***** ,20A4/14HNO.OF SUBJECTS,I5/15HNO.
     1OF VARIABLES,I4)
42    FORMAT (1X,17HWITH CONTIGUITIES////)
43    FORMAT (1X,20HWITHOUT CONTIGUITIES////)
52    FORMAT (20I4)
80    FORMAT (/5H STEP,I4,2X,I4,1X,28HGROUPS AFTER COMBINING GROUP,I4,10
     1H AND GROUP,I4,20X,6HERROR=,F13.4,5X,10HCUMULATED=,F14.4/)
107   FORMAT (//1X,46HCONTIGUITY CRITERION PREVENTS FURTHER GROUPING)
110   FORMAT (8X,5HGROUP,I5,1X,1H(,I3,1X,6HITEMS),12I4/(30X,12I4))
600   FORMAT (1H1)
C ******** READ TITLE CARD AND PROBLEM PARAMETER CARD
7     READ (5,6) (TITL(I),I =1,20)
      READ (5,1) NS,NV,KP,KS,KT,KCONT,ML,IJ,KX
      NP =((NS*NS)+NS)/2
      NH =NS*2
      WRITE (6,600)
      WRITE (6,9) (TITL(I),I =1,20),NS,NV
      IF (KCONT.EQ.0) GO TO 41
      WRITE (6,42)
      GO TO 45
41    WRITE (6,43)
C ******** READ IN IDENTIFICATION NUMBERS IF OPTIONED
45    IF (IJ)44,47,44
47    DO 48 I =1,NS
48    ID(I) =I
```

```
      GO TO 46
44    READ (5,52) (ID(I),I =1,NS)
46    READ (5,6) (FMT(I),I =1,20)
      IF (ML.EQ.0) ML =20
      IF (KP.EQ.0) KP =NS-1
      LX =0
      LL =((NS*NS)+NS)/2
      DO 3 I =1,LL
3     A(I) =0.0
      L =0
      DO 24 JJ =1,NV
C ******** READ IN SUBJECTS IN VARIABLE ORDER - STANDARDIZE VARIABLES
C ******** IF OPTIONED
      READ (5,FMT) (D(J),J =1,NS)
      IF (KS.EQ.0) GO TO 12
      BSUM =0.0
      SUM =0.0
      DO 4 I =1,NS
      SUM =SUM+D(I)
4     BSUM =BSUM+(D(I)*D(I))
      FM =NS
      XBAR =SUM/FM
      SBAR =XBAR*XBAR
      STD =SQRT((BSUM/FM)-SBAR)
      X =1.0/STD
      DO 8 I =1,NS
8     D(I) =(D(I)-XBAR)*X
12    NG =NS-1
      DO 17 I =1,NG
      K =I+1
      DO 17 J =K,NS
      X =D(J)-D(I)
      L =I+(J*J-J)/2
      A(L) =A(L)+(X*X)
17    CONTINUE
24    CONTINUE
      DO 32 I =1,NG
      K =I+1
      DO 32 J =K,NS
      L =I+(J*J-J)/2
      A(L) =A(L)*0.5
      IF (A(L).EQ.0.0) A(L) =0.000001
32    CONTINUE
C ******** READ IN CONTIGUITY MATRIX STORE AS MINUS VALUES IN A
      IF (KCONT.EQ.0) GO TO 55
      DO 53 I =1,NS
      READ (5,52) (KG(J),J =1,ML)
      DO 51 JJ =1,ML
      IF (KG(JJ).LE.I) GO TO 51
      J =KG(JJ)
      LL =I+(J*J-J)/2
```

```
      A(LL) =A(LL)*(-1.0)
51    CONTINUE
53    CONTINUE
55    NG =NS
      KN =NS
C *** INITIALIZE ARRAYS
      DO 60 I = 1,NS
      KG(I) = I
      KSTORE(I) =1
      MC(I) =I
      LAM(I) =I
60    W(I) = 1.0
C ******** OBTAIN NEXT GROUP TO BE JOINED
      KZ =0
      CUM=0.0
65    NG = NG - 1
      IF (NG.EQ.0) GO TO 120
      X = 10.0**10
      KW =KN-1
      DO 75 II =1,KW
      I =MC(II)
      MJ =II+1
      DO 70 JJ =MJ,KN
      J =MC(JJ)
      LL =I+(J*J-J)/2
      IF (KCONT.EQ.0) GO TO 67
      IF (A(LL).GE.0.0) GO TO 70
67    LJ =I*(I+1)/2
      LS =J*(J+1)/2
      DX =ABS(A(LL))-A(LJ)-A(LS)
      IF (DX .GE. X) GO TO 70
      X = DX
      L = I
      M = J
70    CONTINUE
75    CONTINUE
C ******** CHECK WHETHER CONTIGUITY CRITERION (IF OPTIONED) PREVENTS
C ******** FURTHER GROUPING
      IF (KCONT.EQ.0.OR.X.NE.10.0**10) GO TO 76
      WRITE (6,107)
      GO TO 120
76    KZ =KZ+1
      CUM=CUM+X
      IF(KT.NE.1) GO TO 77
      KJ =NS+KZ
      KHOLD(KZ) =L
      KHOLD(KJ) =M
      CONT(KZ) =X
77    WRITE (6,80) KZ,NG,L,M,X,CUM
C ******** UPDATE GROUP MEMBERSHIP AND SET ARRAY FOR DENDOGRAM
      LJK =W(L)
```

```
      K =LAM(M)
      DO 85 I =1,NS
      IF (KG(I).NE.M) GO TO 81
      KG(I) =L
      LAM(I) =LAM(L)
      KSTORE(I) =KSTORE(I)+LJK
      GO TO 85
81    IF (LAM(I).GT.K) LAM(I) =LAM(I)-1
85    CONTINUE
      IF (KCONT.EQ.0) GO TO 89
C ******** ENSURE NEW GROUP HAS CONTIGUITIES OF ELIMINATED GROUP
      DO 87 I =1,NS
      IF (M.GT.I) GO TO 82
      LL =M+(I*I-I)/2
      GO TO 83
82    LL =I+(M*M-M)/2
83    IF (A(LL).GE.0.0.OR.L.EQ.I) GO TO 87
      IF (L.GT.I) GO TO 84
      LL =L+(I*I-I)/2
      GO TO 86
84    LL =I+(L*L-L)/2
86    IF (A(LL).LT.0.0) GO TO 87
      A(LL) =A(LL)*(-1.0)
87    CONTINUE
C ******** UPDATE ERROR POTENTIALS - NEW GROUP TO REMAINING GROUPS
89    WS =W(L)+W(M)
      LL =L+(M*M-M)/2
      X =ABS(A(LL))*WS
      LJ =L*(L+1)/2
      LS =M*(M+1)/2
      Y =ABS(A(LJ))*W(L)+ABS(A(LS))*W(M)
      A(LJ) =ABS(A(LL))
      DO 95 I = 1,NS
      IF (I .EQ. L .OR. KG(I) .NE. I) GO TO 95
      LL =I*(I+1)/2
      XY =ABS(A(LL))*W(I)
      IF (I.GT.L) GO TO 88
      LL =I+(L*L-L)/2
      LS =I+(M*M-M)/2
      GO TO 91
88    IF (I.LT.M) GO TO 90
      LL =L+(I*I-I)/2
      LS =M+(I*I-I)/2
      GO TO 91
90    LL =L+(I*I-I)/2
      LS =I+(M*M-M)/2
91    ATEMP =(ABS(A(LL))*(W(I)+W(L))+ABS(A(LS))*(W(I)+W(M))+X-Y-XY)/(W(I
     1)+WS)
      IF (A(LL).LT.0.0) GO TO 93
      A(LL) =ATEMP
      GO TO 95
```

```
93      A(LL) =ATEMP*(-1.0)
95      CONTINUE
        W(L) = WS
C *** REWRITE ARRAY OF REMAINING GROUPS SANS THE ONE JUST ELIMINATED
        JJ =0
        DO 92 I =1,KN
        IF (MC(I).EQ.M) GO TO 92
        JJ =JJ+1
        MC(JJ) =MC(I)
        J =MC(I)
        NC(JJ) =W(J)
92      CONTINUE
        KN =KN-1
        IF (NG.GT.KP) GO TO 65
C ******** ORDER MEMBERS OF EACH GROUP READY FOR PRINTING
        JJ =0
        DO 96 I =1,KN
        LL =NC(I)
        NC(I) =JJ+1
96      JJ =JJ+LL
        DO 100 I =1,NS
        J =LAM(I)
        K =NC(J)
        LC(K) =I
100     NC(J) =K+1
C ******** PRINT GROUP MEMBERSHIP
        JJ =0
        DO 115 I =1,KN
        J =JJ+1
        JJ =NC(I)-1
        L =(JJ-J)+1
        IF (L.LE.1) GO TO 115
        LL =LC(J)
        IF (IJ.EQ.0) GO TO 114
        DO 112 K =J,JJ
        LJ =LC(K)
112     LC(K) =ID(LJ)
114     WRITE (6,110) LL,L,(LC(LJ),LJ =J,JJ)
115     CONTINUE
        GO TO 65
120     IF (KT.NE.1) GO TO 135
        NY =NS-NG
        IF (NG.NE.0) GO TO 145
C *** SET UP READY FOR DENDOGRAM - GROUPING COMPLETE
        DO 130 I =1,NY
        LL =KSTORE(I)
        KG(LL) =I
  130 CONTINUE
        GO TO 138
C *** SET UP READY FOR DENDOGRAM - GROUPING INCOMPLETE DUE TO CONTIGUITY
C *** CRITERION
```

```
145   LJ =0
      DO 160 I =1,NS
      IF (KG(I).NE.I) GO TO 160
      L =0
      DO 150 J =1,NS
      IF (KG(J).NE.I) GO TO 150
      L =L+1
      NJ =KSTORE(J)
      LC (NJ) =J
150   CONTINUE
      DO 155 J =1,L
      LJ =LJ+1
      LAM(LJ) =LC(J)
155   CONTINUE
160   CONTINUE
      CALL TREE (A,NY,LJ,NP,NH,NS,MC,LAM,CONT,KHOLD)
      GO TO 135
138   CALL TREE (A,NY,NS,NP,NH,NS,MC,KG,CONT,KHOLD)
135   IF (KX.NE.1) GO TO 7
      STOP
      END
      SUBROUTINE TREE (DATA,NY,N,NP,NH,NS,JOKE,KG,CONT,KHOLD)
C ******** THIS SUBROUTINE PRINTS A DENDOGRAM ON THE LINE PRINTER
C ******** DIMENSION SIZE OF DATA MUST BE SET EQUAL TO AT LEAST (N*2)+1
C ******** FOR A MAXIMUM OUTPUT OF 120 CHARACTERS PER RECORD CHANGE THE
C ******** VALUES OF LMJ AND LJR IN THE DATA STATEMENT TO 84 AND 116
      DIMENSION KG(NS),JOKE(NS),CONT(NS),KHOLD(NH)
      DIMENSION DATA(NP),IMT(29),FSPEC(8)
      INTEGER DATA,FSPEC
      DATA JV,JX,JY,JW,LMJ,LJR/1H ,1H*,1H_,1H|,96,128/
      DATA FSPEC/1H7,1H6,1H5,1H4,1H3,1H2,1H1,1H0/
      DATA IMT/1H(,1H3,1HI,1H5,1H,,1H2,1HX,1H,,1HF,1H1,1H0,1H.,1H8,1H,,1
     1H2,1HX,1H,,1HA,1H1,1H,,1H1,1HX,1H,,1H1,1H0,1H1,1HA,1H1,1H)/
1     FORMAT (I4,15(4X,I4))
2     FORMAT (5X,13HITEMS GROUPED,6X,13(4X,I4))
3     FORMAT (26X,13(4X,I4))
16    FORMAT (1X,4HSTEP,4X,1HI,4X,1HJ,4X,5HERROR,4X,13(4X,I4))
17    FORMAT (30X,13(4X,I4))
20    FORMAT (3X,129A1)
25    FORMAT (2X,I4,15(4X,I4))
26    FORMAT (16(4X,I4))
27    FORMAT (2X,16(4X,I4))
32    FORMAT (29X,A1)
34    FORMAT (29X,A1,1X,101A1)
55    FORMAT (1H1)
C *** INITIALIZE VARIABLES
      NJ =NY-1
      L =(N*2)+1
      NM =0
      LX =LMJ
      MRJ =1
```

```
      KJ =LMJ/2
      KL =LJR/2
      FORM =10.0
      NL =KJ
      IF (N.LE.KJ) NL =N
4     WRITE (6,55)
      IF (LX.GE.L-2) GO TO 19
      MJ =LX
      GO TO 13
19    MJ =L
13    IF (MRJ.NE.1) GO TO 21
      KW =LMJ
C ******** PRINT OUT HEADINGS
      WRITE (6,2) (KG(I),I =1,NL,4)
      WRITE (6,3) (KG(I),I =2,NL,4)
      WRITE (6,16) (KG(I),I =3,NL,4)
      WRITE (6,17) (KG(I),I =4,NL,4)
      GO TO 22
21    NL =KJ+1
      KJ =KJ+KL
      IF (KJ.GE.N) KJ =N
      KW =LJR
      WRITE (6,1) (KG(I),I =NL,KJ,4)
      JJ =NL+1
      WRITE (6,25) (KG(I),I =JJ,KJ,4)
      JJ =JJ+1
      WRITE (6,26) (KG(I),I =JJ,KJ,4)
      JJ =JJ+1
      WRITE (6,27) (KG(I),I =JJ,KJ,4)
C ******** SET UP ARRAY ORDER FOR PRINTING
22    JS =0
      DATA(L) =JX
      GO TO 40
9     JS =JS+1
      K =0
      L =0
      DO 5 I =1,N
      J =KG(I)
      L =L+1
      JOKE(J) =L
      DATA(L) =JW
      L =L+1
      DATA(L) =JV
    5 CONTINUE
      L =L+1
      DATA(L) =JX
C *** OBTAIN NEXT LINE FOR PRINTING
   14 K =K+1
      IF (K.GT.NJ) GO TO 40
      KP =NS+K
      KM =KHOLD(K)
```

```
      KK =KHOLD(KP)
      LL =(JOKE(KK)-JOKE(KM))-1
      LS =JOKE(KM)+1
      LJ =(LS+LL)-1
      DO 10 I =LS,LJ
10    DATA(I) =JY
C ******** OBTAIN APPROPRIATE FORMAT AND PRINT LINE
      IF (MRJ.NE.1) GO TO 28
      IF (CONT(K).LT.FORM) GO TO 80
      NM =NM+1
      IMT(13) =FSPEC(NM)
      FORM =FORM*10.0
80    WRITE (6,IMT) K,KM,KK,CONT(K),JX,(DATA(I),I =1,MJ)
      GO TO 11
28    WRITE (6,20) (DATA(I),I =MRJ,MJ)
C *** CHANGE INDEXES MODIFY LINE
11    LS =LS-1
      LJ =LJ+1
      DO 12 I =LS,LJ
   12 DATA(I) =JV
      LS =(LL/2)+1
      JOKE(KM) =JOKE(KM)+LS
      LS =JOKE(KM)
      DATA(LS) =JW
      GO TO 14
C ******** CHECK LINE STATUS WITH REPECT TO BORDERS AND MODIFY ARRAYS
40    JJ =MJ-1
      LJ =MRJ+1
      IF (MRJ.NE.1) GO TO 37
      IF (NL.GT.KJ) GO TO 56
      DO 43 I =1,JJ
43    DATA(I) =JV
      WRITE (6,34) JX,(DATA(I),I =1,MJ)
      GO TO 65
56    WRITE (6,32) JX
65    IF(JS)85,9,85
37    IF(LX.GE.L-2)GO TO 42
      WRITE(6,20)JV
      GO TO 800
42    DO 31 I=MRJ,JJ
31    DATA(I)=JV
      WRITE (6,20) (DATA(I),I =MRJ,MJ)
800   IF(JS.EQ.0)GO TO 9
85    DO 33 I=MRJ,MJ,2
33    DATA(I)=JX
      DO 70 I =LJ,MJ,2
70    DATA(I) =JV
      IF(MRJ.NE.1)GO TO 39
      WRITE (6,34) JX,(DATA(I),I =1,MJ)
      GO TO 38
39    WRITE(6,20)(DATA(I),I=MRJ,MJ)
```

```
38      IF(LX.GE.L-2)GO TO 41
C ******** SET UP INDICES FOR NEXT SECTION OF ARRAY
        MRJ =MRJ+KW
        LX =LX+LJR
        GO TO 4
41      RETURN
        END
```

APPENDIX V

An Algorithm for the Graphical Generation of a Dendogram From any Stepwise Hierarchical Grouping Method

Graphical display and even interpretation of the results of any type of stepwise hierarchical linkage analysis can frequently be enhanced by the presentation of the completed grouping in the form of the familiar tree graph or dendogram. An algorithm for achieving this goal is presented in this appendix.

The basic objective is to determine a unique positioning of the ordered groups such that in the descendent tree projected from this ordering no horizontal span (line), which bridges the vertical branches representing the junction of two groups at some step in the linking process, will intersect the span of some other pair of groups joined at any prior or subsequent stage in the generation of the dendogram. This goal will be assisted by the convention that on the completion of the grouping of any pair of groups i and j into a single member group i, all current members of the jth group will be given the label i. For a complete stepwise linkage of n subjects

some n-1 steps, or pairs of groups, will be generated.

To clarify the algorithmic procedure the following definitions will be made at the outset.

For any step in the grouping procedure

a) Let C be some vector in which is stored the file of the number of members each group currently contains. Any element $c_k, k=1,\cdots,n$ will therefore be the number of subjects in group k.

b) Let A be a singly dimensioned array which labels the group to which each of the n subjects belongs. Thus the value contained in the kth cell of array A, $k=1,\cdots n$, is the group to which the kth subject is currently assigned.

c) Let B be some array in which each of the k elements, $k=1,\cdots,n$, contains an index referencing the order in which that subject (the kth subject) joined the group wherein it is currently located.

The algorithm for obtaining the required unique ordering of all n-1 paired groupings may be described in the following steps.

1. Initialize arrays A, B, and C as follows: Set the kth element in A equal to k. That is $a_k=k$ for all $k-1,\cdots,n$. Set all n elements of arrays B and C equal to 1.

2. Let i and j be the two groups joined at some (beginning with the first) step in the grouping analysis. Note that because of the aforementioned convention the label denoting group i is always < j.

3. Let $\ell = c_i$, the number of groups currently assigned to group i.

4. Scan all n elements in A, $k=1,\cdots,n$, and for any element in A which satisfies the condition $a_k = j$

(a) Set $a_k=i$, which updates membership in group i and

(b) adjust the kth element of B as follows: Let $m = b_k$ and reset b_k such that $b_k = \ell+m = m + c_k$.

5. Increment the number of members in group i by the number of members in group j so that $c_i = c_j + \ell$.

Repeat steps 2 to 5 for a total of n-1 iterations.

6. Redefine j to be some index, initially set equal to zero, and generate a new array D with n elements, which will contain the ordered positioning of the groups as sought.

7. Increment j by 1. Let i be the index stored in the jth cell of B, so that $i=b_j$. Now set $d_i=j$.

8. Repeat step 7 n times.

To illustrate this algorithm consider the problem of generating a dendogram from the stepwise hierarchical grouping of 6 subjects labelled consecutively 1 through 6. Suppose further that the following groups are paired in each of the n-1 steps: i) groups 1 and 6, ii) groups 3 and 4, iii) groups 2 and 5, iv) groups 1 and 2, and v) groups 1 and 3. The unique positioning of the ordered pairs for the generation of the dendogram from this stepwise grouping is illustrated in Figure 17. Figure 18 shows a 40 subject dendogram compiled from a set of data grouped by the well known Ward (1963) algorithm.

Step	Groups	ℓ	|A|	|B|	|C|
			1 2 3 4 5 6	1 1 1 1 1 1	1 1 1 1 1 1
1	1 and 6	1	1 2 3 4 5 1	1 1 1 1 1 2	2 1 1 1 1 0
2	3 and 4	1	1 2 3 3 5 1	1 1 1 2 1 2	2 1 2 0 1 0
3	2 and 5	1	1 2 3 3 2 1	1 1 1 2 2 2	2 2 2 0 0 0
4	1 and 2	2	1 1 3 3 1 1	1 3 1 2 4 2	4 0 2 0 0 0
5	1 and 3	4	1 1 1 1 1 1	1 3 5 6 4 2	6 0 0 0 0 0

|D| = 1 6 2 5 3 4

Dendogram for a 6 Subject Example

Step	Groups
1	1 6
2	3 4
3	2 5
4	1 2
5	1 3

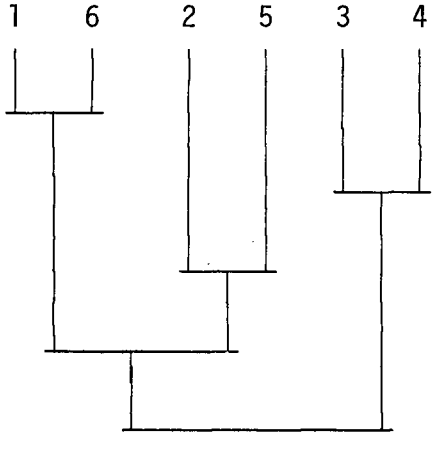
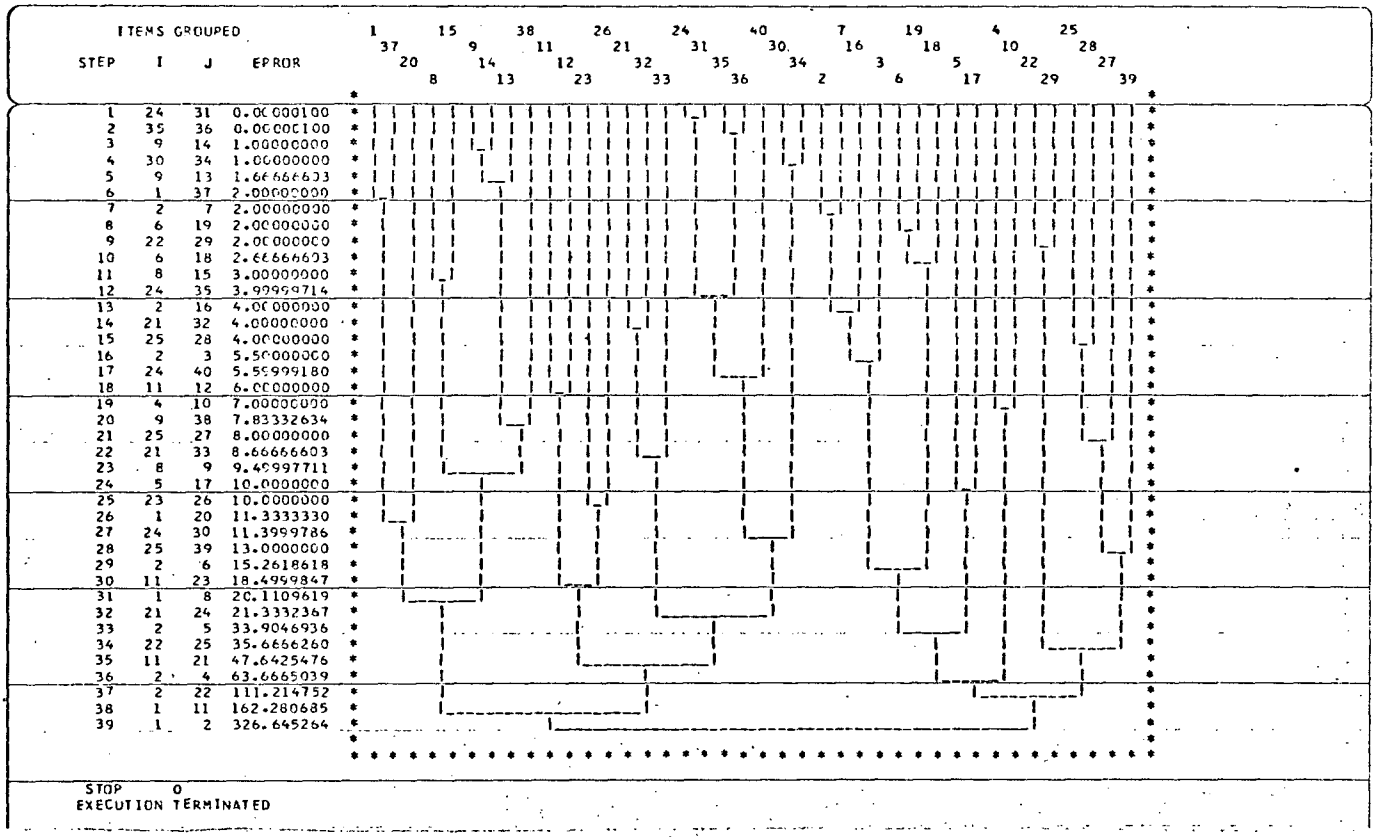


Figure 17: Illustration of Dendogram Algorithm for a 6 Subject Example.



STEP	I	J	EPROR
1	24	31	0.00000100
2	35	36	0.00000100
3	9	14	1.00000000
4	30	34	1.00000000
5	9	13	1.66666603
6	1	37	2.00000000
7	2	7	2.00000000
8	6	19	2.00000000
9	22	29	2.00000000
10	6	18	2.66666603
11	8	15	3.00000000
12	24	35	3.99999714
13	2	16	4.00000000
14	21	32	4.00000000
15	25	28	4.00000000
16	2	3	5.50000000
17	24	40	5.59999180
18	11	12	6.00000000
19	4	10	7.00000000
20	9	38	7.83332634
21	25	27	8.00000000
22	21	33	8.66666603
23	8	9	9.49997711
24	5	17	10.0000000
25	23	26	10.0000000
26	1	20	11.3333330
27	24	30	11.3999786
28	25	39	13.0000000
29	2	6	15.2618618
30	11	23	18.4999847
31	1	8	20.1109619
32	21	24	21.3332367
33	2	5	33.9046936
34	22	25	35.6666260
35	11	21	47.6425476
36	2	4	63.6665039
37	2	22	111.214752
38	1	11	162.280685
39	1	2	326.645264

Figure 17

Graphical Display of a 40 Subject Dendogram Grouped by the Ward Algorithm

APPENDIX VI

A Computer Program for Estimating the Medians of a Weighted Graph Subject, if Optioned, to Side Constraints*

*N.B. A much shorter and faster program of this algorithm which incorporates only the most effective of the various heuristic alternatives, and which does not evaluate combinatorial equalities if they occur, is nearing completion and will be available on request from the Department of Geography, University of British Columbia.

```
C       ***************************************************************
C  ***  THIS PROGRAM ESTIMATES THE P MEDIANS (P LOCATIONS) OF A WEIGHTED
C  ***  GRAPH SUBJECT,IF OPTIONED,TO CERTAIN SIDE CONSTRAINTS.SOLUTIONS
C  ***  WILL BE OPTIMAL OR NEAR OPTIMAL
C  ***  METHOD: SEE 'AN ALGORITHM FOR ESTIMATING THE MEDIANS OF A WEIGHTED
C  ***  GRAPH SUBJECT TO SIDE CONSTRAINTS' - R.A.WHITAKER PHD THESIS
C  ***  DEPARTMENT OF GEOGRAPHY UNIVERSITY OF BRITISH COLUMBIA 1971
C       ***************************************************************
C  ***  N.B. NETWORK NODES SHOULD BE LABELLED IN SEQUENCE 1,...,N
C       *****DECK SET UP AS FOLLOWS*****
C  ***  A). PROBLEM TITLE CARD MAXIMUM OF 80 COLUMNS ALLOWED
C  ***  B). CONTROL CARD
C  ***     1.. COLS 1-3 NUMBER OF NODES ON THE NETWORK (N)
C  ***     2.. COLS 4-6 NUMBER OF POTENTIAL LOCATION SITES (M)
C  ***     3.. COLS 7-9 NO. OF NODES TO BE FORCED INTO THE SOLUTION IF ANY
C  ***       SEE INSTRUCTION D
C  ***     4.. COL 10 THIS CONTROL SPECIFIES THE MAJOR PROGRAM OPTION;
C  ***       PUNCH 1 FOR THE CLASSICAL PLANT LOCATION PROBLEM. HERE THE
C  ***       VALUE OF P IS UNKNOWN AND BOTH PLANT AND TRANSPORT COSTS
C  ***       OBTAIN. BEGINNING AT A SPECIFIED LOWER BOUND (SEE INSTRUCTION
C  ***       K1) P IS INCREMENTED AT EACH STEP UNTIL THE P+1 SOLUTION HAS
C  ***       A TOTAL COST > THAN THAT OF THE P SOLUTION
C  ***       PUNCH 2 FOR STATIC LOCATION OPTION WHERE THE VALUE(S) OF P ARE
C  ***       PRE-DETERMINED BY THE USER. (SEE INSTRUCTION K2)
C  ***       PUNCH 3 FOR DYNAMIC LOCATION BACKWARD RECURSIVE OPTION
C  ***       PUNCH 4 FOR DYNAMIC LOCATION FORWARD RECURSIVE OPTION
C  ***     5.. COLS 11-12 THIS CONTROL SPECIFIES THE NUMBER OF MEDIAN
C  ***       INDICATORS THE PROGRAM EXPECTS TO BE READ IN. IT IS DEPENDENT
C  ***       UPON THE VALUE PUNCHED IN COLUMN 10 AS FOLLOWS;
C  ***       IF COLUMN 10 CONTAINS 1 THEN PUNCH IN 1
C  ***       IF COLUMN 10 CONTAINS 2 THEN PUNCH IN THE NUMBER OF MEDIANS
C  ***       TO BE COMPUTED E.G. A 3 WOULD INDICATE THAT THE PROGRAM
C  ***       EXPECTS TO READ IN & CALCULATE 3 SEPERATE MEDIAN VALUES
C  ***       IF COLUMN 10 CONTAINS 3 OR 4 THEN PUNCH IN 2
C  ***     6.. COL 13 PUNCH 1 IF PLANT SIZE UPPER BOUNDS ARE OPTIONED (MA)
C  ***     7.. COL 14 PUNCH 1 IF A MAXIMUM DISTANCE CONSTRAINT (FROM EACH
C  ***       NODE TO SOME SOURCE) IS OPTIONED (KY)
C  ***     8.. COL 15 PUNCH 1 IF PLANT COSTS ARE INCLUDED IN THIS RUN (MY)
C  ***     9.. 16 THIS CONTROL OPTIONS A USER SUPPLIED WEIGHTING SYSTEM
C  ***       FOR PARTITIONING THE GRAPH IF THE USER DOES NOT WISH TO RELY
C  ***       SOLELY UPON THOSE GENERATED BY THE ALGORITHM. THE MAXMIN
C  ***       PARTITIONING CRITERION IS USED. PUNCH 0 IF THIS OPTION IS NOT
C  ***       SPECIFIED-PUNCH 1 IF EXTENDED (ROW BY ROW) SEARCH IS REQUIRED
C  ***       PUNCH 2 FOR SHORT SEARCH PATTERN IN WHICH AN INITIAL SET IS
C  ***       NOT DEFINED FOR EACH ROW BUT OVER ALL ROWS.
C  ***    10.. COL 17 PUNCH 1 IF USER SUPPLIED DATA SUBROUTINE IS TO BE
C  ***       CALLED. SEE SECTION ON USER SUPPLIED SUBROUTINES (MW)
C  ***    11.. COL 18 PUNCH 1 IF USER SUPPLIED SUBROUTINE FOR REESTIMATING
C  ***       UNIT TRANSPORT COSTS IS CALLED. SEE USER SUPPLIED SUBROUTINES
C  ***    12.. COL 19 FIRST PARTITIONING HEURISTIC OPTION-THE SINGER
C  ***       CRITERION - PUNCH 0 IF EXTENDED (ROW BY ROW) SEARCH IS
```

```
C ***     REQUIRED - PUNCH 1 FOR SHORT SEARCH PATTERN - PUNCH 2 TO
C ***     EXCLUDE THIS HEURISTIC OPTION
C ***   13.. COL 20 SECOND PARTITIONING HEURISTIC OPTION PUNCH 0 IF
C ***     EXTENDED SEARCH (ROW BY ROW) SEACH IS REQUIRED. PUNCH 1 FOR
C ***     SHORT SEARCH PATTERN - PUNCH 2 TO EXCLUDE THIS HEURISTIC
C ***     OPTION
C ***   14.. COL 21 THIRD PARTITIONING HEURISTIC OPTION - WEIGHT SHIFT
C ***     ELIMINATION METHOD-PUNCH 0 TO INCLUDE THIS HEURISTIC PUNCH 1
C ***     TO EXCLUDE IT
C ***   15.. COL 22 FOURTH PARTITIONING HEURISTIC-WEIGHTED VERSION
C ***     OF COOPERS METHOD- PUNCH 0 TO INCLUDE THIS HEURISTIC PUNCH 1
C ***     TO EXCLUDE THIS HEURISTIC
C ***     SOME COMMENTS ON THE ABOVE CHOICES. FROM ONE TO ALL OF THE
C ***     SPECIFIED METHODS MAY BE USED TO  GIVE THE INITIAL START.
C ***     THE THIRD HEURISTIC GIVES IN GENERAL THE BEST STARTING
C ***     SOLUTION FOLLOWED BY THE SINGER CRITERION. COMPUTATION TIMES
C ***     FOR THE PARTITIONING PROCEDURES ARE SIGNIFICANTLY AFFECTED
C ***     BY BOTH THE NUMBER OF POTENTIAL PLANT SITES AND BY THE
C ***     HEURISTICS CHOSEN.THE SHORT VERSIONS OF THE FIRST TWO
C ***     HEURISTICS ARE THE FASTEST BUT THE EXTENDED VERSIONS OF THESE
C ***     METHODS ARE COMPUTATIONALLY THE MOST EXACTING.FOR VERY LARGE
C ***     PROBLEMS (SEVERAL HUNDRED POTENTIAL PLANT SITES) THE
C ***     SHORT VERSIONS OF THE HEURISTICS SHOULD BE USED IF COMPUTING
C ***     TIME IS SIGNIFICANT FOR THE USER.IN SUCH CASES THE FOURTH
C ***     HEURISTIC (AND PERHAPS THE THIRD FOR THE BIGGEST NETWORKS)
C ***     MAY ALSO BE IGNORED.FOR EXAMPLE TO SOLVE A 10 MEDIAN NETWORK
C ***     PROBLEM WITH 100 POTENTIAL PLANT SITES WILL TAKE ABOUT 2 TO
C ***     2.5 MINUTES (DEPENDING ON TOTAL NUMBER OF NODES) TO EXECUTE
C ***     ON AN IBM 360/67 IF ALL FOUR HEURISTICS ARE USED.USING THE
C ***     SHORT SINGER CRITERION SEARCH AND THE THIRD PARTITIONING
C ***     HEURISTIC WILL REDUCE THIS TO LESS THAN 1 MINUTE AS A RULE
C ***     SUMMARIZING--EXCEPT FOR SMALL PROBLEMS USE EXTENDED SINGER AND
C ***     THE THIRD HEURISTIC.FOR LARGE TO VERY LARGE PROBLEMS USE THE
C ***     SHORT VERSIONS OF THE FIRST TWO HEURISTICS AND THE THIRD
C ***     HEURISTIC.FOR EXCEPTIONALLY LARGE PROBLEMS THE THIRD HEURISTIC
C ***     MAY BE DROPPED IF A COMPUTATIONALLY FAST RESULT IS REQUIRED
C ***   16.. COL 23 PUNCH 0 IF ALL COMBINATIONS OF EQUIDISTANT NODES
C ***     (NODES EQUIDISTANT FROM MORE THAN ONE CURRENT MEDIAN) ARE TO
C ***     BE EVALUATED DURING COMPUTATION.MAXIMUM NUMBER CONSIDERED IS
C ***     RESTRICTED. PUNCH 1 TO EXCLUDE ALL COMBINATIONS SEARCH
C ***   17.. COL 24 IF USER SUPPLIED GUESS AT OPTIMAL SOLUTION IS
C ***     OPTIONED PUNCH AS FOLLOWS: 1 IF THIS IS IN ADDITION TO PROGRAM
C ***     GENERATED BEGINNING SETS OR 2 WITHOUT THE PROGRAM GENERATED
C ***     SETS. N.B. THIS OPTION IS ONLY ALLOWABLE IS COLUMN 10 IS
C ***     PUNCHED WITH A 2. USER MUST ENSURE PLANT SIZE CONSTRAINT IS
C ***     NOT VIOLATED
C ***   18.. COL 25 LEAVE BLANK IF PLANT COSTS ARE NOT OPTIONED OR IF
C ***     THEY ARE AND BOTH SETS OF PLANT COST HEURISTICS ARE TO BE
C ***     ACTIVATED. PUNCH 1 IF THE SECOND SET OF PLANT COST HEURISTICS
C ***     IS TO BE IGNORED.
C ***   19.. COL 26 LEAVE BLANK IF THE PLANT SIZE CONSTRAINT OPTION
```

```
C ***      IS NOT SPECIFIED. IF IT IS OPTIONED THEN PUNCH 0 TO ACTIVATE
C ***      BOTH SETS OF PLANT SIZE CONSTRAINT HEURISTICS. PUNCH 1 TO
C ***      ACTIVATE THE FIRST SET ONLY; PUNCH 2 TO ACTIVATE THE SECOND
C ***      SET ONLY.--N.B. THE SECOND SET IS SLIGHTLY TO BE PREFERRED TO
C ***      THE FIRST.
C ***   20.. COL 27 LEAVE BLANK IF DYNAMIC LOCATION OPTION IS NOT
C ***      SPECIFIED.PUNCH 1 IF A DIFFERENT SET OF MARKET DEMANDS (NODE
C ***      WEIGHTS) IS TO BE READ IN FOR EACH TIME PERIOD -IF NOT PUNCH (
C ***   21.. COL 28 LEAVE BLANK IF DYNAMIC LOCATION OPTION IS NOT
C ***      SPECIFIED. PUNCH 1 IF A DIFFERENT SET OF TRANSPORT COSTS
C ***      (DISTANCES) IS TO BE READ IN FOR EACH TIME PERIOD - OTHERWISE
C ***      LEAVE BLANK
C ***   22 COL 29 LEAVE BLANK IF DYNAMIC LOCATION OPTION IS NOT
C ***      SPECIFIED. PUNCH 1 IF A DIFFERENT SET OF PLANT COSTS IS TO BE
C ***      READ IN VIA SUBROUTINE DATA (SEE SECTION ON USER SUPPLIED
C ***      SUBROUTINE) FOR EACH TIME PERIOD - OTHERWISE LEAVE BLANK.
C ***   23.. COLS 30-39 MAXIMUM COST BUDGET CONSTRAINT ON THE TOTAL
C ***      SYSTEM IF OPTIONED. ONLY SOLUTIONS FOUND SATISFYING THIS ARE
C ***      PRINTED
C ***   24.. COL 40 PUNCH 1 IF THIS IS THE LAST (OR ONLY) JOB IN THE RUN
C *** C). READ IN THE NODE DESIGNATIONS (LABELS) OF ALL THE POTENTIAL
C ***      LOCATION SITES - 4 COLS PER NODE. IGNORE THIS INSTRUCTION IF
C ***      ALL NETWORK NODES ARE POTENTIAL PLANT SITES
C *** D). READ IN THE LIST OF NODES TO BE FORCED INTO SOLUTION IF ANY
C ***      N.B. THESE NODES MIGHT LABEL AN EXISTING SYSTEM OF LOCATIONS
C ***      TO WHICH NEW SOURCES ARE TO BE ADDED. ALTERNATIVELY THEY WILL
C ***      REFLECT SOME A PRIORI REASONS WHICH REQUIRE A CERTAIN PLANT(S)
C ***      TO BE IN SOLUTION
C *** E). READ IN VARIABLE FORMAT CARD FOR PLANT SIZE UPPER BOUNDS IF
C ***      OPTIONED E.G. (10F5.1)
C *** F). READ IN PLANT SIZE UPPER BOUNDS UNDER THIS FORMAT IF OPTIONED
C *** G). READ IN VARIABLE FORMAT CARD FOR MAXIMUM DISTANCE CONSTRAINT
C ***      IF OPTIONED
C *** H). READ IN MAXIMUM DISTANCE VALUES IF OPTIONED UNDER THIS FORMAT
C *** I). READ IN VARIABLE FORMAT CARD FOR USER SUPPLIED WEIGHTING
C ***      SYSTEM FOR INITIAL PARTITIONING IF OPTIONED
C *** J). READ IN THE USER SUPPLIED WEIGHTS UNDER THE SPECIFIED FORMAT
C *** K). READ IN A VECTOR (4 COLS PER NODE) WHICH CONTAINS THE VALUES
C ***      OF ALL LOCATION(S) TO BE COMPUTED. THIS WILL BE DEPENDENT UPON
C ***      THE VALUES PREVIOUSLY PUNCHED UNDER INSTRUCTIONS B4(COL 10)
C ***      AND B5(COLS 11-12)
C ***   1.. IF COL 10 IS PUNCHED 1 THEN PUNCH THE LOWER BOUND STARTING
C ***       POINT FOR INCREMENTATION OF THE VALUES OF P. E.G. A 2 WOULD
C ***       INSRUCT THE PROGRAM TO BEGIN WITH THE 2 MEDIAN SOLUTION AND
C ***       INCREMENT THE VALUE OF P SUCCESSIVELY UNTIL THE P+1 SOLUTION
C ***       EXCEEDS THE P SOLUTION AT WHICH TIME EXECUTION TERMINATES
C ***   2.. IF COL 10 IS PUNCHED WITH A 2 THEN THE PROGRAM EXPECTS TO
C ***       COMPUTE AS MANY MEDIANS AS ARE SPECIFIED IN COLS 11-12. E.G.
C ***       IF THIS IS 3 THEN 3 MEDIAN VALUES WILL BE READ IN. SUPPOSE
C ***       THESE TO BE 4,7 AND 9 THEN THE 4,7 AND 9 MEDIAN LOCATIONS OF
C ***       THE NETWORK WILL BE COMPUTED
```

```
C ***    3.. IF COL 10 IS PUNCHED 3 THEN THE PROGRAM EXPECTS TO READ IN
C ***        2 MEDIAN VALUES; AN UPPER STARTING AND LOWER FINISHING VALUE
C ***        RESPECTIVELY - IN THAT ORDER. E.G. IF THESE VALUES WERE 7
C ***        AND 2 THEN THE PROGRAM WOULD COMPUTE THE 7 MEDIAN SOLUTION
C ***        FIRST FOLLOWED BY THE 6 MEDIAN SOLUTION - (CHOSEN FROM AMONG
C ***        THE PREVIOUS 7), AND SO ON DOWN TO THE 2 MEDIAN SOLUTION
C ***    4.. IF COL 10 IS PUNCHED 4 THEN THE PROGRAM EXPECTS TO READ IN
C ***        2 MEDIAN VALUES; A LOWER STARTING AND UPPER FINISHING VALUE
C ***        RESPECTIVELY - IN THAT ORDER. E.G. IF THESE VALUES WERE 2
C ***        AND 7 THEN THE PROGRAM WOULD COMPUTE THE 2 MEDIAN SOLUTION
C ***        FIRST FOLLOWED BY THE 3 MEDIAN SOLUTION - (CONTAINING THE 2
C ***        MEDIANS FOUND IN THE PREVIOUS STEP AS FORCED NODES) AND SO
C ***        ON UP TO THE 7 MEDIAN SOLUTION
C *** L). READ IN VARIABLE FORMAT CARD FOR READING IN UNIT TRANSPORT
C ***        COSTS (SHORT PATHS) FROM EACH NODE TO ALL POTENTIAL MEDIANS
C *** M). READ IN THE SHORTEST PATH MATRIX UNDER THE SPECIFIED FORMAT
C ***        N.B. ***WARNING*** THERE WILL BE N ROWS, ONE FOR EACH NETWORK
C ***        NODE AND M COLUMNS, ONE FOR EACH POTENTIAL PLANT SITE. THE JTH
C ***        COLUMN IN THE A(I,J) SHORT PATH MATRIX MUST REFERENCE
C ***        THE DISTANCE FROM NODE I TO THE JTH NODE SPECIFIED IN THE LIST
C ***        OF POTENTIAL PLANT SITES (SEE C). E.G. ELEMENT A(1,5) MUST BE
C ***        THE DISTANCE FROM THE FIRST NODE TO THE FIFTH MEMBER OF THE
C ***        SUBSET OF POTENTIAL PLANT SITES. FURTHERMORE TO SAVE STORAGE
C ***        IT IS ASSUMED THAT THE SHORT PATH MATRIX IS SYMETRIC I.E. THE
C ***        DISTANCE FROM NODE I TO NODE J MUST BE THE SAME AS FROM NODE
C ***        J TO NODE I FOR EACH SUCH PAIR OF ELEMENTS IN THE MATRIX.
C ***        IF THIS IS NOT SO THE PROGRAM WILL REQUIRE SOME MODIFICATION
C ***        WITH MATRIX A REDEFINED AS A DOUBLY DIMENSIONED ARRAY.
C ***        N.B. IF DYNAMIC LOCATION IS OPTIONED SEE INSTRUCTION R FIRST.
C *** N). READ IN VARIABLE FORMAT FOR NODE (MARKET) DEMANDS (WEIGHTS)
C *** P). READ IN MARKET DEMANDS (NODE WEIGHTS) UNDER FORMAT SPECIFIED
C ***        N.B. IF DYNAMIC LOCATION IS OPTIONED SEE INSTRUCTION R FIRST.
C *** Q). READ IN PLANT COST FUNCTIONS VIA SUBROUTINE DATA IF OPTIONED
C ***        N.B. IF DYNAMIC LOCATION IS OPTIONED SEE INSTRUCTION R FIRST.
C *** R). IGNORE THIS INSTRUCTION IF 1), THE DYNAMIC LOCATION OPTION IS
C ***        NOT SPECIFIED OR 2), IF IT IS BUT ONLY ONE SET OF TRANSPORT
C ***        COSTS (DISTANCES), NODE WEIGHTS, AND PLANT COSTS (IF OPTIONED)
C ***        IS TO BE READ IN. OTHERWISE REPEAT INSTRUCTIONS L-Q FOR EACH
C ***        TIME PERIOD WHERE APPLICABLE. E.G. A 3 TIME PERIOD SOLUTION
C ***        WITH DIFFERENT TRANSPORT COST (I.E. SHORT PATH) MATRICES AND
C ***        NODE WEIGHTS (ONE SET FOR EACH PERIOD) WILL REQUIRE REPITITION
C ***        OF INSTRUCTIONS L-P THREE TIMES. FOR A SINGLE SET OF TRANSPORT
C ***        COSTS,BUT DIFFERENT NODE WEIGHTS FOR ALL THREE PERIODS CARRY
C ***        OUT INSTRUCTIONS L AND M ONCE BUT N AND P THREE TIMES.
C ***        N.B. ***WARNING*** THE ORDERING BY WHICH THE DIFFERENT SETS
C ***        ARE READ IN IS DEPENDENT UPON WHETHER THE FORWARDS OR
C ***        BACKWARDS RECURSION PROCEDURE IS OPTIONED. FOR A FORWARDS
C ***        RECURSION THE FIRST SET(S) READ IN MUST REFER TO TIME PERIOD
C ***        1, THE SECOND SET TO TIME PERIOD 2, AND THE LAST SET TO THE
C ***        LAST TIME PERIOD. FOR THE BACKWARDS RECURSION THIS ORDERING
C ***        IS REVERSED. THE FIRST SET(S) READ IN MUST REFER TO THE LAST
C ***        TIME PERIOD, THE SECOND SET TO THE SECOND TO LAST TIME PERIOD,
C ***        AND THE LAST SET TO THE FIRST TIME PERIOD.
C *** S). READ IN USER SUPPLIED GUESS AT OPTIMAL SOLUTION IF OPTIONED
```

```
C ***      ALLOW 4 COLS PER NODE. READ IN AS MANY SETS AS ARE SPECIFIED
C ***      BY THE MEDIAN VALUE GIVEN IN COL 11-12. N.B. THIS OPTION IS
C ***      ALLOWABLE ONLY IF COL 10 IS PUNCHED WITH A 2
C      ******************* USER SUPPLIED SUBROUTINES *******************
C *** THE FOLLOWING 2 SUBROUTINES MUST BE DEFINED BY THE USER IF PLANT
C *** COSTS ARE INVOKED.
C *** 1). SUBROUTINE DATA ---
C ***     THIS SUBROUTINE IS USED TO READ IN PLANT COST FUNCTIONS FOR
C ***     EACH POTENTIAL PLANT SITE WHERE THE FUNCTIONS VARY AMONG PLANTS
C ***     CONSIDER THE FOLLOWING EXAMPLE: SUPPOSE EACH POTENTIAL SITE HAS
C ***     A COST FUNCTION OF THE FORM Y= A+BX WHERE A AND B ARE CONSTANTS
C ***     X VARIES ACCORDING TO PLANT SIZE AND THERE ARE 10 SUCH SITES
C ***     SUBROUTINE DATA
C ***     COMMON DAT (2,10)
C ***     DO 5 I =1,10
C ***     READ (5,1) A,B
C *** 1   FORMAT (2F5.2)
C ***     DAT(1,I) =A
C ***     DAT(2,I) =B
C *** 5   CONTINUE
C ***     RETURN
C ***     END
C ***     IN THE FIRST ROW OF MATRIX DAT ALL THE A CONSTANTS ARE STORED
C ***     IN THE SECOND ROW OF MATRIX DAT ALL THE B CONSTANTS ARE STORED
C ***     THE CONTENTS OF DAT ARE TRANSFERRED TO SUBROUTINE COST BY THE
C ***     COMMON STATEMENT
C ***     N.B. THIS SUBROUTINE CAN BE SUPPLIED AS A DUMMY PROVIDING THE
C ***     PLANT COST FUNCTIONS ARE IDENTICAL FOR ALL PLANTS. IN SUCH A
C ***     CASE THE COST FUNCTION CAN BE DEFINED DIRECTLY IN SUBROUTINE
C ***     COST
C *** 2). SUBROUTINE COST ---
C ***     THIS SUBROUTINE CALCULATES THE COST FUNCTION SUPPLIED BY THE
C ***     USER. ITS ARGUMENT LIST MUST CONTAIN THE FOLLOWING; SUM THE
C ***     CURRENT PLANT SIZE WHICH IS CALCULATED IN THE MAIN PROGRAM, LR
C ***     THE INDEX REFERRING TO SOME PLANT, AND Y THE COST OF PRODUCING
C ***     AMOUNT SUM AT SOURCE LR. - E.G. SUPPOSE THE COST FUNCTION IS OF
C ***     THE FORM Y =A+BX, AND THAT THE CONSTANTS A AND B ARE NOT
C ***     IDENTICAL FOR EACH PLANT. THEN, ASSUMING THIS INFORMATION TO
C ***     HAVE BEEN READ IN VIA SUBROUTINE DATA AS JUST DEMONSTRATED WE
C ***     HAVE
C ***     SUBROUTINE COST (SUM,Y,LR)
C ***     DOUBLE PRECISION Y
C ***     COMMON DAT (2,10)
C ***     Y =DAT(1,LV) + (DAT(2,LV)*SUM)
C ***     RETURN
C ***     END
C ***     IF PLANT COSTS ARE THE SAME OVER ALL PLANTS THEN THE INDEX LV
C ***     WILL NOT BE SPECIFICALLY ACTIVATED BUT IS NEEDED IN THE
C ***     THE SUBROUTINE ARGUMENT LIST. N.B. VARIABLE Y MUST ALWAYS BE
C ***     DOUBLE PRECISIONED.
C *** THE FOLLOWING SUBROUTINE MAY BE SUPPLIED BY THE USER AT HIS
C *** DISCRETION IF UNIT TRANSPORT COSTS REFLECT NON-LINEARITIES AND
C *** PLANT SIZE UPPER BOUNDS OBTAIN
```

```
C *** SUBROUTINE TCOST ---
C *** THIS SUBROUTINE MIGHT BE WRITTEN IN THE FOLLOWING CIRCUMSTANCES.
C *** IF THE PLANT SIZE CONSTRAINT IMPOSITION RESULTS IN SOME MARKET
C *** BEING SUPPLIED FROM MORE THAN ONE SOURCE (PLANT) THEN THE UNIT
C *** TRANSPORT COST OVER SOME LINK (WHICH IS AN AVERAGE VALUE BASED ON
C *** COST OF SHIPPING ALL DEMAND OVER THAT LINK),MAY BE MADE REDUNDANT
C *** IF THE COST FUNCTION IS NON LINEAR. THE USER CAN SPECIFY A
C *** FUNCTION IN TCOST TO RECALCULATE NEW UNIT COSTS BASED ON THE
C *** CURRENT AMOUNT ACTUALLY ALLOCATED OVER THAT LINK. THE ARGUMENT
C *** LIST IS IDENTICAL TO THAT OF SUBROUTINE COST I.E. SUBROUTINE
C *** TCOST (SUM,Y,LR)
C     ***********************************************************
C *** A VERSION OF THE BASIC ALGORITHM INCLUSIVE ONLY OF FORCED NODE
C *** CAPABILITY AND RESTRICTIONS ON POTENTIAL SITES IS AVAILABLE.
C *** THIS WILL PRINT ALL COMBINATIONS OF MEDIANS IF THE OPTIMAL SET IS
C *** NOT UNIQUE BUT WILL NOT EVALUATE, DURING COMPUTATION, COMBINATION
C *** OF NODES WHICH MIGHT BE EQUIDISTANT TO MORE THAN ONE MEMBER OF TH
C *** CURRENT MEDIAN LIST. THIS VERSION IS CONSIDERABLY FASTER FOR THE
C *** BASIC ALGORITHM THAN IS THIS PROGRAM
C     ******************* DIMENSIONING INFORMATION *******************
C *** DIMENSION THE FOLLOWING ARRAYS TO AT LEAST N (THE NUMBER OF
C *** NETWORK NODES) STORAGE SPACES --- MC,HN,IN,ID
C *** DIMENSION THE FOLLOWING ARRAYS TO AT LEAST M (THE NUMBER OF
C *** POTENTIAL SITES) SPACES---UP,JN,C,QS,PL,IB,KC,NZ,MZ,MXR,IX,JC,
C *** NST,MT,IZ,VR,Q,MKB,V,F,W,NP,MP
C *** THE FOLLOWING ARRAYS MUST BE DIMENSIONED AT LEAST EQUAL TO M7
C *** WHERE M7 IS SOME ADDITIONAL STORAGE SET ASIDE FOR SHARED NODES OR
C *** ALLOCATIONS. M7 IS DEFINED AT THE BEGINNING OF THE PROGRAM AS 50
C *** BUT MAY BE INCREASED FOR LARGE PROBLEMS --- RS,JG,JA
C *** DIMENSION THE FOLLOWING ARRAYS TO AT LEAST N+M7 --- LW,NC,B,D,JD,
C *** NB,KD,DN,MD,CJ
C *** DIMENSION THE FOLLOWING ARRAYS TO AT LEAST M+M7 --- E,P,MB,KB,LC
C *** DIMENSION ARRAY JQ EQUAL TO AT LEAST M+M8 WHERE M8 IS FOR
C *** ADDITIONAL STORAGE AND IS DEFINED AT THE BEGINNING OF THE PROGRAM
C *** AS 100 BUT MAY BE INCREASED FOR LARGER PROBLEMS WHERE THE PLANT
C *** SIZE CONSTRAINTS ARE VERY RESTRICTIVE
C *** ARRAY A MUST BE DIMENSIONED TO AT LEAST ((M*M)+M)/(M*(N-M)) SPACES
C *** DIMENSION ARRAY KG TO KR*9 STORAGE SPACES WHERE KR IS THE MAXIMUM
C *** NUMBER OF P MEDIANS SPECIFIED DURING ANY ONE RUN
C ***
C *** OBJECT CODE = APPROXIMATELY 75000 BYTES.
C ***
C     ***********************************************************
C ***
C *** PROGRAMMER - R.A.WHITAKER - DEPARTMENT OF GEOGRAPHY,UNIVERSITY OF
C *** BRITISH CLUMBIA,SEPTEMBER 1970
C ***
C     ***********************************************************
      DOUBLE PRECISION E,S,S2,S3,S4,S6,SSS,ASUM,P
      DIMENSION PROB(20),FMT1(20),FMT2(20),FMT3(20),FMT4(20),FMT5(20)
      DIMENSION LW(230),NC(230),B(230),D(230),MC(180),HN(180),PL(55)
```

```
      DIMENSION C(55),IB(55),KC(55),NZ(55),MB(120),V(55)
      DIMENSION JN(55),UP(55),MZ(55),MXR(55),JD(220),IN(180)
      DIMENSION IX(55),E(120),LC(120),KB(120),JC(55),NB(220)
      DIMENSION NST(55),MT(55),ID(180),IZ(55),KG(300),MKB(55)
      DIMENSION KD(230),DN(230),RS(50),JQ(150),VR(55)
      DIMENSION A(5000),JG(50),JA(50),JUD(10),P(180)
      DIMENSION QS(55),Q(55),MD(220),CJ(220)
      DIMENSION F(20),NP(20),MP(20),W(20)
      COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
      COMMON/COM2/AM,AP,DS,BD,SX,PM,SY
      EQUIVALENCE (KC(1),IX(1))
3     FORMAT (20A4)
5     FORMAT (3I3,I1,I2,17I1,F10.3,I1)
22    FORMAT (5X,85HNUMBER OF LOCATIONS SPECIFIED IS LESS THAN NUMBER OF
     1 NODES TO BE FORCED INTO SOLUTION)
23    FORMAT (5X,74HNUMBER OF LOCATIONS SPECIFIED IS GREATER THAN NUMBER
     1 OF POSSIBLE LOCATIONS)
138   FORMAT (40HCAPACITY CONSTRAINT CRITERION PREVENTS A,I4,1X,17HLOCAT
     1ION SOLUTION/)
134   FORMAT (5X,110HEXECUTION TERMINATING - SYSTEM HIGHLY CONSTRAINED -
     1 INCREASE PLANT UPPER BOUNDS BY JUST 1 UNIT EACH AND RE-RUN)
1     READ (5,3) (PROB(I),I =1,20)
C ******** READ IN CONTROL AND TITLE CARDS
      READ (5,5) N,M,MM,NM,KA,MA,KY,MY,MQ,MW,J2,J5,J6,J9,J7,LJS,KZP,J1,I
     1KS,I2,IKK,JKK,BD,LFIN
      M7 =50
      M8 =100
      WRITE (6,2) (PROB(I),I =1,20)
2     FORMAT (1H1,2X,19HPROBLEM NAME ..... ,20A4//)
      LH =M+M7
      IH =N+M7
      LI =((M*M)+M)/2+(M*(N-M))
      IKP =((M*M)-M)/2
      SY =10.0**15
      PM =SY
      K3 =0
      LE =2
      LZ =M*9
      IF (BD.EQ.0.0) BD =10.0**15
4     FORMAT (20A4)
      IF (N.EQ.M) GO TO 8
6     FORMAT (20I4)
C ******** READ IN POTENTIAL LOCATION SITES
      READ (5,6) (JN(I),I =1,M)
      GO TO 9
 8    DO 61 I =1,N
 61   JN(I) =I
9     IF (N.EQ.M) GO TO 60
C ******** SET UP INDEXING SYSTEM IN ARRAY ID FOR REFERENCING SHORT
C ******** PATHS STORED IN ARRAY A
C ******** SET UP ARRAY MC AS FOLLOWS
```

```
C ******** ZERO VALUES INDICATE THIS NODE NOT A POTENTIAL PLANT SITE
C ******** LESS THAN ZERO INDICATES THIS NODE TO BE FORCED INTO SOLUTION
C ******** GREATER THAN ZERO INDICATES NODE TO BE A POTENTIAL LOCATION
C ******** ACTUAL VALUES OF ELEMENTS REFERENCE THE COLUMN INDICES OF THE
C ******** SHORT PATH MATRIX A
      DO 45 I =1,N
      ID(I) =0
45    MC(I) =0
      DO 50 I =1,M
      L =JN(I)
      ID(L) =I
50    MC(L) =I
      J =M
      DO 55 I =1,N
      IF (ID(I).NE.0) GO TO 55
      J =J+1
      ID(I) =J
55    CONTINUE
      GO TO 70
60    DO 65 I =1,N
      ID(I) =I
65    MC(I) =I
70    IF (MM.EQ.0) GO TO 90
C ******** READ IN NODES (IF ANY) WHICH MUST BE IN FINAL SOLUTION
      READ (5,6) (NC(I),I =1,MM)
      DO 80 I =1,MM
      J =NC(I)
      NP(I) =MC(J)
      MP(I) =MC(J)
80    MC(J) =MC(J)*(-1)
90    IF (MA.EQ.0) GO TO 10
C ******** READ IN PLANT SIZE UPPER BOUNDS - IF OPTIONED
      READ (5,4) (FMT3(I),I =1,20)
      READ (5,FMT3) (UP(I),I =1,M)
      GO TO 12
10    DO 11 I =1,M
11    UP(I) =SY
12    IF (KY.EQ.0) GO TO 13
C ******** READ IN MAXIMUM DISTANCE ANY DEMAND POINT IS TO BE FROM SOME
C ******** PLANT - IF OPTIONED
      READ (5,4) (FMT4(I),I =1,20)
      READ (5,FMT4) (HN(I),I =1,N)
      GO TO 15
13    DO 14 I =1,N
14    HN(I) =SY
C ******** READ IN USER SUPPLIED WEIGHTING SYSTEM - IF OPTIONED
15    IF (MQ.EQ.0) GO TO 18
      READ (5,4) (FMT5(I),I =1,20)
      READ (5,FMT5) (C(I),I =1,M)
C ******** READ IN VECTOR WHICH CONTAINS THE VALUES OF THE LOCATIONS
C ******** THAT ARE TO BE COMPUTED E.G. 2 FOR A 2 MEDIAN SOLUTION
```

```
18      READ (5,6) (JUD(I),I =1,KA)
        IVKX =0
64      IF (IKK.EQ.2) GO TO 63
        READ (5,4) (FMT1(I),I =1,20)
C ******** READ IN TRANSPORT COST,TIME,DR,SHORTEST ROUTE MATRIX
        DO 35 I =1,N
        K =ID(I)
        READ (5,FMT1) (D(J),J =1,M)
        DO 35 J =1,M
        A(ICHEC(K,J,M)) =D(J)
35      CONTINUE
        IF (IKK.EQ.0) IKK =2
63      IF (I2.EQ.2) GO TO 67
C ******** READ IN NODE DEMAND VALUES
        READ (5,4) (FMT2(I),I =1,20)
        READ (5,FMT2) (D(I),I =1,N)
        IF (I2.EQ.0) I2 =2
67      IF (MW.EQ.0) GO TO 118
        IF (JKK.EQ.2) GO TO 118
        CALL DATA
        IF (JKK.EQ.0) JKK =2
118     SX =100.0
        IF (IVKX.EQ.1) GO TO 21
        IVKX =1
        DS =0.0
        DO 17 I =1,N
17      DS =DS+D(I)
        IJKM =0
        ASUM =SY
        KRB =0
20      KRB =KRB+1
        IKS =IRS
        IF (KRB.GT.KA) GO TO 515
C ******** GET THE NEXT MEDIAN VALUE TO BE COMPUTED
        KP =JUD(KRB)
21      IF (KP.GE.MM) GO TO 25
        WRITE (6,22)
        GO TO 515
25      IF (KP.LE.M) GO TO 24
        WRITE (6,23)
        GO TO 515
24      JJ =0
        LS =0
        K9 =0
        TSUM =0.0
        KG(1) =1
C ******** STORE FORCED NODES FOR PARTITIONING PURPOSES AND INITIALIZE
C ******** ARRAYS FOR FEASIBILITY CHECK
        DO 130 I =1,M
        J =JN(I)
        IF (MC(J))132,130,133
```

```
132     JJ =JJ+1
        B(JJ) =0.0
        JC(JJ) =I
        LS =LS+1
        LC(LS) =I
        TSUM =TSUM+UP(I)
        GO TO 130
133     JJ =JJ+1
        JC(JJ) =I
        B(JJ) =UP(I)
130     CONTINUE
        SUM =TSUM
127     IF (KP.EQ.1.AND.MA.EQ.0) GO TO 275
        IF (KZP.EQ.2) GO TO 275
        IF (MA.EQ.0) GO TO 140
C ******** CHECK THAT TOTAL NETWORK DEMAND CAN BE SATISFIED UNDER PLAN
C ******** SIZE UPPER BOUND OPTION
        IF (KP.EQ.MM) GO TO 136
        CALL ASCEND (N,N,M,1,B,JJ,LW,JC,-1)
        KJ =(JJ-(KP-MM))+1
        DO 135 I =KJ,JJ
135     SUM =SUM+B(I)
136     IF (SUM-DS)137,139,139
137     WRITE (6,138) KP
        K9 =1
        GO TO 503
139     IF (KP.EQ.MM) GO TO 140
        KJ =MM+1
        KK =KP-MM
        K =KJ-1
141     K =K+1
        J =K-1
        SUM =TSUM
        DO 142 I =1,KK
        J =J+1
142     SUM =SUM+B(J)
        IF (SUM-DS)141,143,143
143     PM =B(K)
C ******** ANY COMBINATION OF NODES WHOSE PLANT SIZES EXCEED VALUE OF
C ******** WILL SATISFY THIS DEMAND
        L =K-1
        AM =KP
        IF (B(L).LT.PM.OR.L.EQ.MM) GO TO 140
        IF ((PM*AM).GE.DS) GO TO 140
        PM =PM+0.1
        K =K-1
145     K =K+1
        J =K-1
        SUM =TSUM
        DO 144 I =1,KK
        J =J+1
```

```
      IF (J.GT.KK) GO TO 146
      IF (B(J).LT.PM) GO TO 145
144   SUM =SUM+B(I)
146   IF (SUM.GE.DS) GO TO 140
      WRITE (6,134)
      GO TO 503
140   MGB =0
      IF (KP.EQ.1.OR.KZP.EQ.2) GO TO 275
      IF (KP.EQ.MM) GO TO 174
      DO 73 I =1,M
      L =JN(I)
      PL(I) =D(L)
73    V(I) =D(L)
      IF (N.EQ.M.AND.J7.EQ.1.AND.J9.EQ.1) GO TO 159
C ******** ASSIGN NON POTENTIAL PLANT SITES TO NEAREST POTENTIAL SITE
      JX =0
      DO 158 I =1,N
      AR =SY
      JJ =ID(I)
      DO 262 J =1,M
      IF (I.EQ.JN(J)) GO TO 262
      LL =ICHECK(JJ,J,M)
      IF (A(LL).GE.AR) GO TO 262
      AR =A(LL)
      IJ =J
262   CONTINUE
      IF (MC(I).NE.0) GO TO 263
      V(IJ) =V(IJ)+D(I)
      IF (V(IJ).GT.UP(IJ)) V(IJ) =UP(IJ)
      PL(IJ) =V(IJ)
      GO TO 158
263   JX =JX+1
      JD(JX) =I
      MZ(JX) =I
      NB(JX) =IJ
      JC(JX) =IJ
      Q(JX) =AR
      DN(JX) =AR
      NZ(JX) =IABS(MC(I))
      IZ(JX) =NZ(JX)
158   CONTINUE
159   LX =0
      IP =0
160   KT =0
      LJ =0
      IF (J5.EQ.2) GO TO 250
      IF (J5.EQ.1) LJ =1
C ******** CALCULATE A MATRIX FOR THE FIRST AND THEN SECOND PARTITIONI
C ******** HEURISTICS, AND ON COMPLETION REPLACE SHORT PATH MATRIX.
      CALL ARK (LI,M,0,A,V,UP,VR,LJ,KV,KR,LE,0)
      GO TO 180
```

```
171     IF (J5.EQ.2) GO TO 178
        CALL ARK (LI,M,1,A,V,UP,VR,LJ,KV,KR,LE,0)
178     IF (MQ.EQ.0) GO TO 250
        LJ =0
        IF (MQ.EQ.2) LJ =1
        CALL ARK (LI,M,0,A,C,UP,VR,LJ,KV,KR,LE,0)
        GO TO 180
172     IF (MQ.EQ.0) GO TO 173
        CALL ARK (LI,M,1,A,C,UP,VR,LJ,KV,KR,LE,0)
173     IF (J6.EQ.2) GO TO 250
        LJ =0
        IF (J6.EQ.1) LJ =1
        KT =1
        CALL ARK (LI,M,0,A,C,UP,V,LJ,KV,KR,LE,1)
180     AM =0.0
        IZP =0
        JS =KP
        AR =0.0
        GO TO 205
174     DO 202 I =1,MM
202     KG(I) =LC(I)
        IP =1
        GO TO 275
205     KVP =0
        SUMP =0.0
        IF (IKS.EQ.2) KVP =1
        IF (IKS.EQ.2) SUMP =PM
C ******** CALCULATE BEGINNING SETS BASED ON FIRST OR SECOND HEURISTIC
        DO 240 I =1,M
        II =I
210     IF (LJ.EQ.1) II =KV
        IF (LJ.EQ.1.AND.MM.NE.0) GO TO 237
        IF (UP(II).LT.SUMP) GO TO 235
        JS =MM
        J =JN(II)
        IF (MC(J).LT.0) GO TO 235
        JS =JS+1
        LC (JS) =II
237     CALL SEARCH (N,M,A,B,IP,LC,NC,KC,MC,KG,UP,LI,JN,LX,SUMP,LZ,KT,V)
        IF (KVP) 235,216,235
216     KVP =1
        SUMP =PM
        IF (LJ.EQ.1) KV =KR
        GO TO 210
235     IF (IKS.EQ.2) GO TO 236
        SUMP =0.0
236     IF (LJ.EQ.1) GO TO 250
240     CONTINUE
250     LX =IP
```

```
      MGB =MGB+1
      GO TO (171,172,259),MGB
259   IF (IJKM.EQ.1) GO TO 269
      IF (J7.EQ.1.AND.J9.EQ.1) GO TO 275
C ******** SET UP FOR THIRD AND FOURTH PARTITIONING HEURISTICS
      IF (J7.EQ.1) GO TO 266
      LJ =0
C ******** PARTITION NETWORK BASED ON THIRD AND FOURTH HEURISTICS
      CALL PART (N,M,A,LI,JD,NB,DN,NZ,LJ,V,D,MC,ID,UP,MT,IB,KB,VR,CJ,NP
266   IF (J9.EQ.1) GO TO 273
      LJ =1
      CALL PART (N,M,A,LI,MZ,JC,Q,IZ,LJ,PL,D,MC,ID,UP,MT,IB,KB,VR,CJ,MP
273   IF (MA.EQ.0) IJKM =1
269   JV =0
      IZP =0
      JS =KP
      IF (J7.EQ.1) GO TO 267
      IF (IP.EQ.0) JV =1
      DO 271 I =1,JS
271   JC(I) =NP(I)
      CALL ASCEND (N,M,N,1,B,JS,JC,JD,1)
      CALL COMP (IP,JV,JC,KG,M,LZ)
267   IF (J9.EQ.1) GO TO 275
      IF (IP.EQ.0) JV =1
      DO 268 I =1,JS
268   KC(I) =MP(I)
      CALL ASCEND (N,M,N,1,B,JS,KC,JD,1)
      CALL COMP (IP,JV,KC,KG,M,LZ)
275   IF (KP.EQ.1) IP=1
      JS =KP
      IF (NM.NE.2.OR.KZP.EQ.0.OR.KP.EQ.1) GO TO 182
C ******** READ IN USER SUPPLIED GUESS AT OPTIMUM SOLUTION - IF OPTIONED
      READ (5,6) (KD(I),I =1,JS)
      DO 278 I =1,JS
      J =KD(I)
278   KC(I) =IABS(MC(J))
      IF (KZP.EQ.1) GO TO 177
      IP =0
      JV =1
      GO TO 181
177   JV =0
181   CALL COMP (IP,JV,KC,KG,M,LZ)
182   MRJ =0
      NST(1) =0
      XW =0.0001
      IF (IKS.EQ.2.AND.IRS.NE.2) GO TO 184
      S3 =SY
      SSS =SY
C ******** THIS LOOP CONSIDERS EACH BEGINNING SET IN TURN
184   DO 500 II =1,IP
```

```
      MG =0
      IF (IKS.EQ.-1) IKS=2
183   S1 =SY
      S2 =SY
      KXJ =0
      MIJ =MRJ
      IXX =0
      KGT =0
C ******** GET THE NEXT BEGINNING SET
      DO 335 I =1,JS
      MRJ =MRJ+1
      IB(I) =KG(MRJ)
335   MZ(I) =I
C ******** FIND AND STORE NEXT COMBINATION OF CURRENT MEDIANS
443   NT =1
      NA =1
      JP =1
445   IX(NT) =NA
      IF (NT-JS)447,450,447
447   NA =NA+1
      IF (NA-MZ(JP))447,447,448
448   JP =JP+1
      NT =NT+1
      GO TO 445
450   IF (MA.EQ.0) GO TO 312
      AR =0.0
      DO 418 I =1,JS
      J =IB(I)
418   AR =AR+UP(J)
      IF (AR.LT.DS) GO TO 497
312   KPK =0
      IF (IXX.EQ.0) GO TO 439
C ******** THE FOLLOWING SECTION TESTS THE LOWER BOUND OBTAINED FROM
C ******** SOME BEGINNING SET
      KPK =1
      IF (IXX.GT.1) GO TO 304
      DO 419 I =1,N
      IN(I) =I
419   B(I) =0.0
      DO 417 I =1,JS
      J =IX(I)
      LC(I) =IB(J)
      L2 =LC(I)
      JJ =JN(L2)
417   MT(I) =MC(JJ)
      AR =0.0
      L6 =0
      L8 =0
      BMIN =SY
      SMIN =SY
C ******** FIND MEDIANS TO BE DROPPED
```

```
      DO 282 I =1,JS
      L2 =LC(I)
      JJ =JN(L2)
      IN(JJ) =0
      AR =AR+UP(L2)
      AV =DN(I)
      IF (MC(JJ).LT.0) GO TO 282
      DO 280 J =1,JS
      IF (I.EQ.J) GO TO 280
      L3 =LC(J)
      AW =A(ICHECK(L2,L3,M))*D(JJ)
      IF (AW.GE.SMIN) GO TO 280
      SMIN =AW
      L6 =IX(I)
      AX =UP(L2)
280   CONTINUE
      IF (AV.GE.BMIN) GO TO 282
      BMIN =AV
      L8 =IX(I)
      AXX =UP(L2)
282   CONTINUE
C ******** L6 IS THE MEDIAN TO BE DROPPED WHOSE WEIGHTED DISTANCE TO
C ******** SOME OTHER MEDIAN IS A MINIMUM
C ******** L8 IS THE MEDIAN WITH THE MINIMUM SUBGRAPH DEMAND
      IF (L6.EQ.0) GO TO 521
      AG =AR
      ARR =AR
      AR =AR-AX
      ARR =ARR-AXX
      IF (AR.GE.ARR) AR =ARR
      L4 =0
      L7 =0
      BM =0.0
      SM =0.0
      LJ =0
C ******** FIND SUBSET OF CANDIDATE REPLACEMENT MEDIANS
      DO 287 I =1,LR
      IF (LW(I).EQ.0) GO TO 287
      J =LW(I)
      IF (IN(J).EQ.0.OR.MC(J).EQ.0) GO TO 287
      L5 =MC(J)
      IF (IKS.EQ.2.AND.UP(L5).LT.PM) GO TO 287
      AS =UP(L5)+AR
      IF (AS.LT.DS) GO TO 287
      LJ =LJ+1
      KD(LJ) =L5
287   CONTINUE
      IF (LJ.EQ.0) GO TO 521
      SM =0.0
      AM =SY
C ******** FIND THE CRITERION REPLACEMENT MEDIAN (L4)
```

```
      DO 289 I =1,LJ
      NJ =KD(I)
      DO 286 IJ =1,JS
286   B(IJ) =0.0
      CM =0.0
      DO 288 IJ =1,LR
      IF (LW(IJ).EQ.0) GO TO 288
      J =LW(IJ)
      IF (IN(J).EQ.0) GO TO 288
      JJ =NC(IJ)
      JV =LC(JJ)
      JX =ID(J)
      LL =ICHECK(JX,JV,M)
      JR =ICHECK(JX,NJ,M)
      IF (A(LL).LE.A(JR)) GO TO 288
      BM =(A(LL)-A(JR))*CJ(IJ)
      B(JJ) =B(JJ)+BM
      CM =CM+BM
288   CONTINUE
      IF (CM.LE.SM) GO TO 289
      SM =CM
      L4 =NJ
C ******** FIND THE THIRD DROPPED MEDIAN
      DO 291 IJ =1,JS
      IF ((DN(IJ)-B(IJ)).GE.AM.OR.MT(IJ).LT.0) GO TO 291
      AM =DN(IJ)-B(IJ)
      L7 =IX(IJ)
291   CONTINUE
289   CONTINUE
C ******** ELIMINATE EQUALITIES AMONG THE DROPPED AND THE ADDED MEDIANS
C ******** AND REPLACE DROPPED MEDIAN BY APPROPRIATE ADDED NODE
      IF (L8.EQ.L6) L8 =0
      IF (L7.EQ.L6) L7 =0
      IF (L7.EQ.L8) L7 =0
304   GO TO (305,306,307),IXX
305   IB(L6) =L4
      GO TO 439
306   IF (L8.EQ.0) GO TO 307
      IB(L8) =L4
      GO TO 439
307   IF (L7.EQ.0) GO TO 521
      AG =AG-UP(L7)
      AG =AG+UP(L4)
      IF (AG.LT.DS) GO TO 521
      IB(L7) =L4
439   JM =0
      JL =0
      IF (MG.EQ.1) GO TO 446
C ******** CALL SUBROUTINE FOR ASSIGNING EACH NODE TO SOME SUBGRAPH
      CALL SORT (N,M,B,PL,A,D,IX,IB,NC,LW,UP,QS,JM,JL,JA,JG,JQ,RS,LI,ID
     1E,DN,MT,KD,MC,LJS,JD,M7,M8)
      IF (JM.EQ.0) GO TO 405
```

```
C ******** FIND NEXT COMBINATION (IF ANY) OF NODES EQUIDISTANT TO MEDIA
      LT =1
      LA =1
      LP =1
390   JC(LT) =LA
      IF (LT-JM)395,405,395
395   LA =LA+1
      IF (LA-JA(LP))395,395,400
400   LP =LP+1
      LT =LT+1
      GO TO 390
405   DO 410 I =1,JS
410   V(I) =PL(I)
      DO 411 I =1,N
      KD(I) =I
      IF (NC(I).EQ.0) KD(I) =0
411   CONTINUE
      LR =N
      IF (JM.EQ.0) GO TO 438
C ******** UPDATE ARRAYS ACCORDING TO THIS COMBINATION. ARRAY KD WILL
C ******** CONTAIN THE NODES, ARRAY NC THEIR CORRESPONDING SUBGRAPHS ANC
C ******** D THE AMOUNT OF DEMAND ALLOCATED. INFORMATION ON MARKETS
C ******** SUPPLIED BY MORE THAN ONE SOURCE IS STORED IN ELEMENTS N+1-LR
      DO 440 I =1,JM
      JJ =JG(I)
      KK =JC(I)
      KJ =JQ(KK)
      V(KJ) =V(KJ)+RS(I)
      IF (RS(I).EQ.D(JJ)) GO TO 442
      LR =LR+1
      KD(LR) =JJ
      NC(LR) =KJ
      D(LR) =RS(I)
      JD(LR) =0
      GO TO 440
442   NC(JJ) =KJ
      KD(JJ) =JJ
440   CONTINUE
      IF (MA.EQ.0.OR.IKS.EQ.2) GO TO 438
C ******** CHECK THAT EACH MEDIAN CAN SATISFY ITS SUBGRAPH DEMAND
      DO 413 I =1,JS
      LJ =IX(I)
      LL =IB(LJ)
      IF (V(I).LE.UP(LL)) GO TO 413
      GO TO 490
413   CONTINUE
C ******** STORE ARRAY VALUES IN ORDER READY FOR MEDIAN CALCULATION
438   CALL COUNT (M,JS,LR,IH,NC,LW,KD,B,D,IN,MKB)
      IF (MA.EQ.0.OR.JM.EQ.0.OR.IKS.EQ.2) GO TO 441
C ******** THE FOLLOWING SECTION ATTEMPTS TO RE-ASSIGN NODES MORE
C ******** OPTIMALLY UNDER PLANT SIZE CONSTRAINT OPTION
```

```
      AV =0.0
      MF =0
      X =0.00001
C ******** TEST IF SOME NODE (VR) IS NOT ASSIGNED TO CLOSEST MEDIAN (MK
      DO 329 NS =1,JS
      JW =MKB(NS)-1
      JR =NS-1
      JZ =1
      IF (NS.NE.1) JZ =MKB(JR)
      DO 329 I =JZ,JW
      IR =LW(I)
      IF (JD(IR).EQ.NS) GO TO 329
      NJ =ID(IR)
      JJ =IX(NS)
      MK =IB(JJ)
      LV =ICHECK(NJ,MK,M)
C ******** IF TRUE EXAMINE SUBGRAPH SETS OF ALL CLOSER MEDIANS. TEST
C ******** WHETHER SOME NODE IN 1 OF THESE SETS IF NEARER MK THAN IS VR
      DO 328 J =1,JS
      IF (NS.EQ.J) GO TO 328
      JJ =IX(J)
      KJ =IB(JJ)
      LL =ICHECK(NJ,KJ,M)
      IF (A(LL).GE.A(LV)) GO TO 328
      AM =A(LV)-A(LL)
      KN =MKB(J)-1
      JR =J-1
      KM =1
      IF (J.NE.1) KM =MKB(JR)
C ******** IF TRUE THEN SET UP TO SWITCH NODES. DETERMINE FRACTIONAL
C ******** AMOUNTS TO RE-ASSIGN OR REMAIN
      DO 318 IJ =KM,KN
      IF (LW(IJ).EQ.LW(I)) GO TO 318
      IT =LW(IJ)
      IV =ID(IT)
      LJ =ICHECK(IV,MK,M)
      LL =ICHECK(IV,KJ,M)
      AR =AM-(A(LJ)-A(LL))
      IF (AR.LE.AV) GO TO 318
      AV =AR
      MF =1
      IK =IR
      IL =IT
      KV =IJ
      MV =I
      KR =NS
      JX =J
318   CONTINUE
328   CONTINUE
329   CONTINUE
      IF (MF.EQ.0) GO TO 441
```

```
C ******** REASSIGN NODES AND FRACTIONAL DEMANDS AMONG MEDIANS
      AY =B(KV)
      AX =B(MV)
      IF (AX.GT.(AY-X).AND.AX.LT.(AY+X)) GO TO 320
      IF (AX.LT.AY) GO TO 322
      AX2 =AX-AY
      AX1 =AX-AX2
      AY1 =AY
      AY2 =0.0
      GO TO 323
320   AX1 =AX
      AX2 =0.0
      AY1 =AY
      AY2 =0.0
      GO TO 323
322   AX1 =AX
      AX2 =0.0
      AY2 =AY-AX
      AY1 =AY-AY2
323   CALL INCRE (LR,LR,KD,MD,NC,D,IK,AX1,JX,2,IH,IH)
      CALL INCRE (LR,LR,KD,MD,NC,D,IL,AY1,KR,2,IH,IH)
      IF (AX2.EQ.0.0) GO TO 324
      CALL INCRE (LR,LR,KD,MD,NC,D,IK,AX2,KR,2,IH,IH)
324   IF (AY2.EQ.0.0) GO TO 325
      CALL INCRE (LR,LR,KD,MD,NC,D,IL,AY2,JX,2,IH,IH)
325   IJ =IN(MV)
      JJ =IN(KV)
      KD(IJ) =0
      KD(JJ) =0
      GO TO 438
441   IF (LR.EQ.N) GO TO 446
C ******** AGGREGATE ANY FRACTIONAL ALLOCATIONS ACCORDING TO NODE AND TO
C ******** SUBGRAPH
      MJ =N+1
      CALL ASCEND (IH,IH,IH,MJ,D,LR,KD,NC,3)
      J =N
414   J =J+1
      IF (J.EQ.LR) GO TO 449
      JJ =J+1
      DO 415 K =JJ,LR
      IF (KD(K).GT.KD(J)) GO TO 414
      IF (KD(K).EQ.0.OR.NC(K).NE.NC(J)) GO TO 415
      D(J) =D(J)+D(K)
      KD(K) =0
415   CONTINUE
      GO TO 414
449   IF (LR.EQ.N) GO TO 446
      J =N
      CALL SHRINK (MJ,LR,J,KD,NC,D,IH)
      LR =J
446   IF (MG.LE.0) GO TO 327
```

```
      NR =0
      CALL TEST (N,M,IH,LI,A,D,KD,MC,NC,ID,V,Q,IX,IB,LR,NR)
      IF (NR.EQ.0) GO TO 489
      CALL COUNT (M,JS,LR,IH,NC,LW,KD,B,D,IN,MKB)
327   DO 491 I =1,JS
491   W(I) =V(I)
492   LD =0
      LS =0
C ******** CALL SUBROUTINE TO CALCULATE MEDIANS
      CALL MED (N,M,S,LS,P,NC,NZ,D,MB,LW,MC,A,W,E,LC,HN,UP,LR,LD,LI,MT,
     1D,JN,IB,IX,MKB,IN,MXR,LH,B,IH,DN)
      IF (S.GE.S1) GO TO 483
      S1 =S
      LRS =LS
      LPS =LR
      KXJ =0
C ******** STORE CURRENT BEST SOLUTION IN SUBROUTINE TRANS
      CALL TRANS (N,M,V,NZ,NC,MB,KD,D,QS,VR,IZ,NB,KB,MD,CJ,Q,JS,LS,LR,IH
     1,1,DN,F)
483   IF (MY.NE.0.AND.MG.EQ.0.AND.KP.NE.1.OR.IKS.EQ.2) GO TO 489
C ******** TEST FOR A LOWER BOUND SOLUTION PRINT BEST CURRENT SOLUTION
      IF (S.GE.(SSS-XW).OR.S.GE.(ASUM-XW)) GO TO 498
      IF (S.GT.(S3-XW).AND.S.LT.(S3+XW)) GO TO 452
      IF (S.GT.S3) GO TO 498
      NST(1) =0
      KGT =1
      S3 =S
452   IF (KGT.EQ.1) GO TO 489
      CALL PRINT (N,M,W,B,LW,IX,IB,P,E,JN,LD,NST,KD,MKB,K3,DN,IH,MXR,LH,
     1LC,MT)
      GO TO 489
498   IF (KXJ.EQ.0.OR.MY.EQ.0) GO TO 489
      CALL COUNT (M,JS,LPS,IH,NB,LW,MD,B,CJ,IN,MKB)
      DO 493 I =1,JS
493   W(I) =VR(I)
      GO TO 492
489   IF (JM.EQ.0) GO TO 497
C ******** TEST IF LAST COMBINATION OF EQUIDISTANT NODES IS EVALUATED
490   LA =JC(LT)+1
      IF (LA-JA(LP))390,390,495
495   LP =LP-1
      LT =LT-1
      IF (LT)490,497,490
497   IF (KPK.EQ.1) GO TO 485
C ******** TEST WHETHER LAST COMBINATION OF MEDIANS HAS BEEN EVALUATED
      NA =IX(NT)+1
      IF (NA-MZ(JP))445,445,484
484   JP =JP-1
      NT =NT-1
      IF (NT)497,485,497
485   KXJ =0
```

```
C ******** IF LOWER BOUND FOUND ON LAST ITERATION THEN RE-CYCLE
      IF (S1-S2)486,488,488
486   S2 =S1
      IXX =0
      CALL TRANS (N,M,VR,IZ,NB,KB,MD,CJ,Q,V,MZ,NC,IB,KD,D,QS,JS,LRS,LPS
     1IH,0,F,DN)
      IF (KGT.EQ.1) KXJ =1
      KGT =0
      GO TO 443
488   IF (S3.LT.SSS) SSS =S3
      IF (KP.EQ.1) GO TO 503
      IF (KP.EQ.MM) GO TO 521
C ******** SET UP FOR DROPPING AND ADDING NODES
      IF (IXX.EQ.3.OR.MG.EQ.1) GO TO 521
      CALL TRANS (N,M,VR,IZ,NB,KB,MD,CJ,Q,V,MZ,NC,IB,LW,D,QS,JS,LRS,LPS,
     1IH,1,F,DN)
      LR =LPS
      IXX =IXX+1
      GO TO 443
521   IF (MA.EQ.0.OR.IKS.NE.2) GO TO 527
      IKS =-1
      GO TO 528
C ******** RECYCLE CALCULATING PLANT COSTS ON SUBSEQUENT ITERATIONS
527   IF (MY.EQ.0) GO TO 500
      IF (KP.EQ.MM.AND.MG.EQ.-1) GO TO 503
      MG =MG+2
      GO TO (525,526,522,500),MG
525   MG =1
      GO TO 523
526   MG =-1
      GO TO 523
522   MG =2
523   IF (MG.NE.-1) GO TO 524
528   S1 =SY
      S2 =SY
524   LR =LPS
      CALL TRANS (N,M,VR,IZ,NB,KB,MD,CJ,Q,V,MZ,NC,IB,KD,D,QS,JS,LRS,LPS,
     1IH,1,F,DN)
      IXX =0
      IF (MG.NE.2) GO TO 443
      IF (J1.EQ.1) GO TO 500
      MRJ =MIJ
      GO TO 183
500   CONTINUE
C ******** CHECK FOR SECOND PLANT SIZE CONTRAINT HEURISTIC OPTION
      IF (MA.EQ.0.OR.IKS.EQ.-1.OR.IRS.NE.0) GO TO 503
      IKS =2
      DO 509 I =1,IZW
509   KG(I) =0
      GO TO 24
503   GO TO (501,20,502,506),NM
```

```
501   IF (SSS.GE.ASUM.AND.LD.EQ.0) GO TO 515
      ASUM =SSS
      KP =KP+1
      GO TO 64
502   KP =JS-1
      IF (LD.NE.0.AND.K.NE.0) GO TO 515
      IF (KP.LT.JUD(2)) GO TO 515
      IF (K9.EQ.1) GO TO 515
C ******** SET UP FOR NEXT LOCATION SEARCH ON DYNAMIC BACKWARD RECURSIOI
C ******** OPTION. ENSURE THAT SEARCH FOR THE P-1 SOLUTION WILL BE
C ******** RESTRICTED FROM AMONG THE BEST P MEDIANS FOUND ON THIS RUN
      DO 504 I =1,N
      IF (MC(I).LT.0) GO TO 504
      MC(I) =0
504   CONTINUE
      DO 505 I =1,JS
      J =NST(I)
      JJ =JN(J)
      IF (MC(JJ).LT.0) GO TO 505
      MC(JJ) =J
505   CONTINUE
      GO TO 64
506   KP =JS+1
      IF (LD.NE.0.AND.K.NE.0) GO TO 515
      IF (KP.GT.JUD(2)) GO TO 515
      IF (K9.EQ.1) GO TO 21
C ******** SET UP FOR DYNAMIC FORWARD RECURSION OPTION. ENSURE THAT THE
C ******** P MEDIAN SOLUTION FOUND ON THIS RUN WILL BE FORCED INTO THE
C ******** P+1 SOLUTION
      DO 507 I =1,N
      IF (MC(I).LT.0) MC(I) =MC(I)*(-1)
507   CONTINUE
      DO 508 I =1,JS
      NP(I) =NST(I)
      MP(I) =NST(I)
      J =NST(I)
      JJ =JN(J)
508   MC(JJ) =MC(JJ)*(-1)
      MM =JS
      GO TO 21
C ******** GET NEXT PROBLEM SET
515   IF (LFIN)520,1,520
520   STOP
      END
      SUBROUTINE SORT (N,M,B,V,A,D,KC,KB,K,L,U,Q,JM,JL,JA,JG,JQ,RS,LI,ID
     1,R,DN,MT,KD,MC,LJS,JD,M7,M8)
C ******** THIS SUBROUTINE ASSIGNS NODES TO THEIR APPROPRIATE CURRENT
C ******** MEDIAN
      DOUBLE PRECISION TC,RC,R
      DIMENSION B(N),V(M),A(LI),D(N),KC(M),KB(M),U(M),Q(M),L(N)
      DIMENSION K(N),JA(M7),JG(M7),RS(M7),JQ(M8),ID(N),R(M),DN(N)
```

```
      DIMENSION KD(N),MT(M),MC(N),JD(N)
      COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
      COMMON/COM2/AM,AR,DS,BD,SX,PM,SY
      IT =1
      KX =0
      LSK =0
      JV =N
      NR =1
C ******** INITIALIZE ARRAYS
      DO 45 I =1,N
      DN(I) =D(I)
      KD(I) =1
45    L(I) =I
      DO 7 I =1,JS
7     V(I) =0.0
      DO 29 I =1,JS
      LJ =KC(I)
      LL =KB(LJ)
29    Q(I) =U(LL)
      IF (MG.NE.2.AND.MA.EQ.0) GO TO 82
      IF (IKS.NE.2) GO TO 1
      DO 3 I =1,JS
3     Q(I) =10.0**15
      GO TO 82
1     JV =0
      NR =1
      DO 73 I =1,JS
73    MT(I) =0
C ******** ESTABLISH PENALTY VECTOR. FOR EACH NODE STORE THE DIFFERENCE
C ******** BETWEEN ITS WEIGHTED DISTANCES TO ITS CLOSEST TWO MEDIANS
C ******** ACTIVATED FOR PLANT SIZE CONSTRAINT ONLY
      DO 6 I =1,N
      IF (DN(I).EQ.0.0) GO TO 6
      JV =JV+1
      TEMP =SY
      TEMP1 =SY
      DO 2 J =1,JS
      IF (Q(J).EQ.0.0) GO TO 2
      JJ =KC(J)
      KJ =KB(JJ)
      II =ID(I)
      AX =A(ICHECK(II,KJ,M))
      IF (J2.EQ.0) GO TO 61
      SUM =DN(I)
      CALL TCOST (SUM,TC,KJ)
      AX =TC
61    IF (AX.GE.TEMP) GO TO 34
      TEMP1 =TEMP
      TEMP =AX
      MR =J
      GO TO 2
```

```
34      IF (AX.GE.TEMP1) GO TO 2
        TEMP1 =AX
2       CONTINUE
        KD(JV) =MR
        IF (LSK.EQ.0) JD(JV) =MR
        MT(MR) =MT(MR)+1
        B(JV) =TEMP-TEMP1
        IF (MC(I).LT.0) B(JV) =-SY
        L(JV) =I
6       CONTINUE
        LSK =1
        IF (JV.EQ.0) GO TO 12
C ******** RANK PENALTY VECTOR IN DESCENDING ORDER FOR NODE ASSIGNMENTS
        CALL ASCEND (N,N,N,1,B,JV,KD,L,0)
        IF (MG.NE.2) GO TO 82
C ******** SET UP NODE ORDER ASSIGNMENT FOR SECOND PLANT COST HEURISTICS
        LR =1
        JX =0
75      JY =0
76      JY =JY+1
        IF (JY.GT.JS) GO TO 75
        IF (MT(JY))77,76,77
77      LJ =LR
        DO 80 I =LJ,JV
        IF (KD(I).NE.JY) GO TO 80
        JX =JX+1
        JW =JX+1
        IF (JW.GT.JV) GO TO 83
        IF (JW.GT.I) GO TO 78
        JJ =KD(I)
        JR =L(I)
        II =I
        KK =II
        DO 86 J =JW,II
        KK =KK-1
        KD(KK+1) =KD(KK)
        L(KK+1) =L(KK)
86      CONTINUE
        KD(JX) =JJ
        L(JX) =JR
78      LR =LR+1
        MT(JY) =MT(JY)-1
        GO TO 76
80      CONTINUE
83      IF (KX.NE.0) GO TO 82
        DO 84 I =1,JS
        JJ =L(I)
        DN(JJ) =0.0
        V(I) =V(I)+D(JJ)
        Q(I) =Q(I)-D(JJ)
84      K(JJ) =I
```

```
      IF (KX.EQ.0) NR =JS+1
      KX =1
C ******** CONSIDER EACH NODE IN ORDER FOR ASSIGNMENT
82    DO 10 I =NR,JV
      JJ =L(I)
      SUMMA =DN(JJ)
      MR =KD(I)
      IJ =1
      MT(IJ) =MR
      JZ =MR
      IF (MA.NE.0.AND.LJS.NE.0.AND.MG.NE.2) GO TO 9
      JR =ID(JJ)
      RC =SY
      SUM =SUMMA+V(MR)
      SUMR =SUM
      CALL SET (MR,J2,JR,KC,KB,LI,TC,M,A,SUMMA,LV)
      IF (MG.NE.2) GO TO 22
      S =V(MR)
      CALL FCOST (S,SUM,LV,TC)
C ******** THIS LOOP ASSIGNS A NODE TO ITS APPROPRIATE MEDIAN
22    DO 8 J =1,JS
      IF (JZ.EQ.J.OR.Q(J).LE.0.0) GO TO 8
      SUM =SUMMA+V(J)
      CALL SET (J,J2,JR,KC,KB,LI,RC,M,A,SUMMA,LV)
      IF (MG.NE.2) GO TO 23
      S =V(J)
      CALL FCOST (S,SUM,LV,RC)
23    IF (RC-TC)26,28,8
28    IF (LJS)8,31,8
C ******** CHECK NUMBER OF EQUIDISTANT NODES FOR COMBINATION LIMIT
31    IF (IT-16)27,27,8
26    MR =J
      IJ =1
      MT(IJ) =MR
      TC =RC
      SUMR =SUM
      GO TO 8
27    IF (MG.LE.0) GO TO 43
C ******** IN SECOND PLANT COST HEURISTIC OPTION ASSIGN NODE TO MEDIAN
C ******** WITH LARGEST CURRENT PLANT SIZE IN EVENT OF EQUALITY
      IF (V(MR).GE.V(J)) GO TO 8
      MR =J
      MT(1) =MR
      SUMR =SUM
      GO TO 8
43    SUMR =SUMR-SUMMA
      IJ =IJ+1
      MT(IJ) =J
8     CONTINUE
9     IF (MA.EQ.0.OR.IKS.EQ.2) GO TO 63
      S =SUMMA
```

```
      LR =MT(1)
C ******** ARRAY Q CONTAINS AMOUNT OF AVAILABLE SPACE LEFT AT EACH
C ******** MEDIAN. REDUCE Q AT PLANT WITH MINIMUM AVAILABLE SPACE
C ******** IN EVENT OF EQUALITY
      DO 66 J =1,IJ
      KJ =MT(J)
      IF (Q(KJ).GE.SUMMA) GO TO 66
      LR =KJ
      SUMMA =Q(KJ)
66    CONTINUE
      MR =LR
63    Q(MR) =Q(MR)-SUMMA
      DN(JJ) =DN(JJ)-SUMMA
      IF (SUMMA.LT.D(JJ).OR.IJ.NE.1) GO TO 67
      K(JJ) =MR
      V(MR) =SUMR
      GO TO 69
67    K(JJ) =0
      IT =IT*IJ
      IF (IT.GT.16) IJ =1
      IF (IJ.EQ.1) IT =16
      JM =JM+1
C ******** ASSIGN FRACTIONAL NODE DEMANDS AND/OR EQUIDISTANT NODES TO
C ******** MEDIANS
      DO 64 J =1,IJ
      KJ =MT(J)
      CALL INCRE (JM,JL,JG,JA,JQ,RS,JJ,SUMMA,KJ,1,M7,M8)
64    CONTINUE
69    IF (Q(MR).LE.0.0) GO TO 1
10    CONTINUE
12    RETURN
      END
      SUBROUTINE MED (N,M,S,LS,P,NB,NZ,D,KB,LW,MC,A,V,E,LC,HN,UP,LR,LD,L
     1I,MT,ID,JN,IB,KC,MKB,KD,MXR,LH,C,IH,F)
C ******** THIS SUBROUTINE FINDS THE POTENTIAL PLANT SITES IN EACH
C ******** SUBGRAPH ARRAYS THEM AND CALCULATES MINIMUM AGGREGATE TRAVEL
C ******** COSTS FROM EACH ONE TO ALL OTHERS WITHIN THE SUBGRAPH
      DOUBLE PRECISION Y,R,P,E,S,B1,B2
      DIMENSION NZ(M),V(M),UP(M),D(IH),HN(N),IB(M),KC(M),F(M)
      DIMENSION A(LI),NB(IH),LW(IH),MC(N),ID(N),MT(M),JN(M),C(IH)
      DIMENSION LC(LH),P(LH),E(LH),KB(LH),KD(N),MKB(M),MXR(M)
      COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
      COMMON/COM2/AM,AR,DS,BD,SX,PM,SY
      L =0
      JW =0
C ******** CONSIDER EACH SUBGRAPH IN TURN
      DO 50 I =1,JS
      JB =KC(I)
      JQ =IB(JB)
      JV =JN(JQ)
      JZ =JW+1
```

```
      JW =MKB(I)-1
      LJ =L+1
      L =L+1
      LC(L) =JQ
C ******** STORE PREVIOUS BEST SOLUTION IN ARRAY MXR
      MXR(I) =L
      CALL ASCEND (IH,IH,IH,JZ,C,JW,LW,NB,2)
      IF (MC(JV).LT.0) GO TO 7
      L =L-1
C ******** FIND POTENTIAL SUBGRAPH PLANT SITES AND STORE IN ARRAY LC
      DO 85 J =JZ,JW
      K =LW(J)
      IF (MC(K).LE.0) GO TO 85
      L =L+1
      LC(L) =MC(K)
      IF (MC(K).EQ.JQ) MXR(I) =L
85    CONTINUE
C ******** ARRAY LW CONTAINS ALL THE NODES IN THE SUBGRAPH SET
7     DO 9 J =LJ,L
      KB(J) =1
9     P(J) =0.0
C ******** CALCULATE TRANSPORT (DISTANCE) COSTS FOR EACH POTENTIAL SITE
      DO 10 II =LJ,L
      JJ =LC(II)
      DO 10 J =JZ,JW
      LL =LW(J)
      LV =ID(LL)
      R =A(ICHECK(LV,JJ,M))
      IF (R.LE.HN(LL)) GO TO 3
      KB(II) =0
3     R =R*C(J)
      P(II) =P(II)+R
10    CONTINUE
      IF (KY.EQ.0.OR.LD.EQ.1) GO TO 15
C ******** CHECK MAXIMUM DISTANCE CONSTRAINT VIOLATION IF IMPOSED. STORE
C ******** MINUS VALUES IN ARRAY P FOR SITES NOT MEETING CONSTRAINT
      JX =0
      DO 25 J =LJ,L
      JJ =LC(J)
      IF (KB(J).NE.0) GO TO 25
      IJ =JN(JJ)
      IF (MC(IJ).LT.0) GO TO 12
      P(J) =P(J)*(-1.0)
12    JX =JX+1
25    CONTINUE
      JJ =(L-LJ)+1
      IF (JX.NE.JJ) GO TO 15
      LD =1
15    MT(I) =L
C ******** CALCULATE TOTAL COST FUNCTION FOR EACH POTENTIAL SOURCE
      DO 28 J =LJ,L
```

```
      K =LC(J)
      SUM =V(I)
      E(J) =0.0
      IF (MG.EQ.0.AND.KP.NE.1.OR.MY.EQ.0) GO TO 28
      CALL COST (SUM,Y,K)
      E(J) =Y
28    CONTINUE
50    CONTINUE
      IF (LD.NE.1.OR.KY.NE.1) GO TO 26
C ******** TEST WHETHER EACH SUBGRAPH HAS ONE POTENTIAL SITE NOT GREATEI
C ******** THAN MAXIMUM DISTANCE CONSTRAINT.IF TRUE SUBJECT ONLY PLUS
C ******** P VALUES TO THE MEDIAN SEARCH. IF FALSE NO SOLUTION ON THIS
C ******** ITERATION THEREFORE RESET P VALUES ALL POSITIVE AND CONTINUE
      DO 27 J =1,L
      IF (P(J).GE.0.0) GO TO 27
      P(J) =P(J)*(-1.0)
27    CONTINUE
26    XX =0.0001
      S =0.0
      LS =0
      L =0
      KK =1
      DO 2 I =1,N
2     KD(I) =I
C ******** FIND THE MEDIAN FOR EACH SUBGRAPH
      DO 45 I =1,JS
      LJ =L+1
      L =MT(I)
      K =0
      SM =V(I)
      IF (IKS.EQ.2) SM =PM
      TEMP =SY
      DO 30 J =LJ,L
      IF (KY-1)22,23,24
22    EX =P(J)+E(J)
      GO TO 29
23    EX =P(J)
      GO TO 29
24    EX =E(J)
29    JJ =LC(J)
      JL =JN(JJ)
      IF (MC(JL).LT.0) GO TO 31
      IF (P(J).LT.0.0.OR.UP(JJ).LT.SM) GO TO 30
31    IF (KD(JL).EQ.0) GO TO 30
      IF (EX.GT.(TEMP-XX).AND.EX.LT.(TEMP+XX)) GO TO 32
      IF (EX.GT.TEMP) GO TO 30
      TEMP =EX
      II =J
      K =1
      LS =KK
      GO TO 37
```

```
32      LS =LS+1
C ******** STORE NEW BEST SOLUTION IN ARRAY KB
37      KB(LS) =LC(J)
30      CONTINUE
        IF (K.NE.0) GO TO 52
        S =10.0**15
        GO TO 42
52      IF (MA.EQ.0) GO TO 55
C ******** CHECK TO ENSURE THAT THE SAME NODE IS NOT CHOSEN AS A MEDIAN
C ******** IN MORE THAN 1 SUBGRAPH. CAN OCCUR IF PLANT SIZE CONSTRAINED
        DO 71 IJ =KK,LS
        KV =KB(IJ)
        JL =JN(KV)
71      KD(JL) =0
55      NZ(I) =LS
        F(I) =P(II)
        S =S+TEMP
        KK =LS+1
45      CONTINUE
C ******** CHECK BUDGET CONSTRAINT VIOLATION IF IMPOSED
42      IF (S.LE.BD.OR.LD.EQ.1) GO TO 35
        LD =2
35      RETURN
        END
        SUBROUTINE PRINT (N,M,V,B,LW,KC,KB,SM,SB,JN,LD,NST,KD,MKB,K3,DN,IH
       1,MXR,LH,LC,MT)
C ******** THIS SUBROUTINE PRINTS SOLUTIONS FOR THE LOWER BOUNDS
C ******** N.B. THE WRITE,DATA,AND FORMAT STATEMENTS IN THIS SUBROUTINE
C ******** ARE DESIGNED FOR A 130 PLUS LINE PRINTER
        DOUBLE PRECISION TC,SM,SB,S,S1,S2
        DIMENSION LW(IH),KC(M),KB(M),SM(LH),SB(LH),IMT(15),IMT1(9),KD(IH)
        DIMENSION JN(M),B(IH),V(M),NH(65),NST(M),ISPEC(5),ISPEC1(5),MKB(M)
        DIMENSION  DN(IH),MXR(M),LC(M),MT(M)
        COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
        COMMON/COM2/AM,AR,DS,BD,SX,PM,SY
        DATA JV,JK/1H*,5/
        DATA ISPEC/1H1,1H2,1H3,1H4,1H5/
        DATA ISPEC1/1H4,1H8,2H12,2H16,2H20/
        DATA IMT/4H(I4,,3HI7,,4HF12.,3H2,F,4H14.2,4H,2F1,4H2.2,,3H3X,,1H5,
       13HI4,,1H4,2HX,,1H5,4HF8.1,1H)/
        DATA IMT1/3H(64,2HX,,1H5,3HI4,,1H4,2HX,,1H5,4HF8.1,1H)/
2       FORMAT (1X,65A1,65A1)
5       FORMAT (5X,6HMEDIAN,2X,10HPLANT SIZE,2X,12HFREIGHT COST,2X,10HPLAN
       1T COST,2X,10HTOTAL COST,6X,14HNODES SERVICED,12X,30HVOLUME SUPPLIE
       2D TO EACH MEDIAN/)
8       FORMAT (//14X,80HNO SOLUTION FOUND ON THIS ITERATION SATISFYING MA
       1XIMUM DISTANCE CONSTRAINT FOR A,I4,1X,17HLOCATION SOLUTION/)
11      FORMAT (5X,80HNO SOLUTION FOUND ON THIS ITERATION SATISFYING TOTAL
       1 PLANT COST CONSTRAINT FOR A,I4,1X,17HLOCATION SOLUTION)
12      FORMAT (/44X,25HPLANT LOCATION-ALLOCATION,I3,16H MEDIAN SOLUTION/)
70      FORMAT (4X,7HNETWORK,F12.2,F14.2,2F12.2/)
```

```
102   FORMAT (1X,65A1,65A1)
      IF (LD-1)3,7,9
7     WRITE(6,8) JS
      GO TO 75
9     WRITE (6,11) JS
      GO TO 75
3     IF (NM.LT.3.OR.NST(1).NE.0) GO TO 6
C ******** STORE LOWER BOUND MEDIANS IF DYNAMIC LOCATION IS OPTIONED
      DO 4 I =1,JS
      J =KC(I)
4     NST(I) =KB(J)
6     DO 1 I =1,65
1     NH(I) =JV
      IF (K3.NE.0) GO TO 14
      K3 =1
      WRITE (6,2) (NH(I),I =1,65),(NH(I),I =1,65)
14    WRITE (6,12) JS
      WRITE (6,5)
      S =0.0
      S1 =0.0
      S2 =0.0
C ******** ARRANGE SUBGRAPH MEDIANS IN ASCENDING ORDER FOR PRINTING
      DO 15 I =1,JS
      LL =KC(I)
      LC(I) =I
      MT(I) =KB(LL)
15    DN(I) =0.0
      CALL ASCEND (N,M,M,1,DN,JS,MT,LC,3)
C ******** CONSIDER EACH SUBGRAPH IN TURN SET UP INDICES FOR PRINTING
      DO 65 IJ =1,JS
      I =LC(IJ)
      JW =MKB(I)-1
      KR =I-1
      IF (KR.EQ.0) JZ =1
      IF (KR.NE.0) JZ =MKB(KR)
      L =MT(IJ)
      II =MXR(I)
      JJ =0
C ******** OBTAIN SUBGRAPH NODES FOR PRINT OUT
      DO 10 J =JZ,JW
      JJ =JJ+1
      KD(JJ) =LW(J)
10    DN(JJ) =B(J)
      TC =SM(II)+SB(II)
      JY =JK
      IF (JY.GT.JJ) JY=JJ
      JX =1
      JP =(JY-JX)+1
      LP =(JK-JP)+1
      IMT(9) =ISPEC(JP)
      IMT(11) =ISPEC1(LP)
```

```
      IMT(13) =ISPEC(JP)
      WRITE (6,IMT) IJ,JN(L),V(I),SM(II),SB(II),TC,(KD(J),J =JX,JY),(DN
     1J),J =JX,JY)
31    JX =JY+1
      JY =JY+JK
      IF (JX.GT.JJ) GO TO 33
      IF (JY.GT.JJ) JY =JJ
      JP =(JY-JX)+1
      LP =(JK-JP)+1
      IMT1(3) =ISPEC(JP)
      IMT1(5) =ISPEC1(LP)
      IMT1(7) =ISPEC(JP)
C ******** PRINT OUT ALLOCATIONS
      WRITE (6,IMT1) (KD(J),J =JX,JY),(DN(J),J =JX,JY)
      GO TO 31
C ******** CALCULATE AND PRINT OUT TOTALS
33    S =S+SM(II)
      S1 =S1+SB(II)
      S2 =S2+TC
65    CONTINUE
      WRITE (6,70) DS,S,S1,S2
      WRITE (6,102) (NH(I),I =1,65),(NH(I),I =1,65)
75    RETURN
      END
      SUBROUTINE PART (N,M,A,LI,JD,NB,B,NC,LJ,V,D,MC,ID,UP,MT,IB,KB,VR,G
     1,NP)
C ******** THIS SUBROUTINE OBTAINS THE INITIAL PARTITIONING SET OF P FOR
C ******** THE THIRD AND FOURTH HEURISTICS
      DIMENSION A(LI),JD(M),NB(M),B(M),NC(M),V(M),D(N),MC(N)
      DIMENSION ID(N),UP(M),MT(M),IB(M),KB(M),VR(M),G(M),NP(M)
      COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
      COMMON/COM2/AM,AR,DS,BD,SX,PM,SY
      LM =M
      IF (LM.EQ.MM) GO TO 30
      IF (MA.EQ.0) GO TO 2
      DO 1 I =1,M
      MT(I) =JD(I)
      IB(I) =NB(I)
      KB(I) =NC(I)
      VR(I) =V(I)
1     G(I) =B(I)
      IF (IKS.EQ.2) GO TO 23
2     AR =SY
C ******** THIS LOOP ELIMINATES ONE NODE. THE ONE WHOSE WEIGHTED
C ******** DISTANCE TO SOME OTHER POTENTIAL SITE IS A MINIMUM
      DO 15 I =1,LM
      JJ =JD(I)
      IF (MC(JJ).LT.0) GO TO 15
      MJ =NC(I)
      AM =B(I)*D(JJ)
      IF (LJ.EQ.1) AM =B(I)*V(MJ)
```

```
      IF (AM.GE.AR) GO TO 15
      AR =AM
      JR =NC(I)
      JV =NB(I)
      IJ =I
15    CONTINUE
      IF (LJ.EQ.1) V(JV) =V(JV)+V(JR)
      IF (V(JV).GT.UP(JV)) V(JV) =UP(JV)
      NP(LM) =JR
      JJ =IJ-1
      KJ =IJ+1
      IF (KJ.GT.LM) GO TO 17
C ******** REWRITE AND SHRINK ARRAYS SANS DROPPED NODE
      DO 16 I =KJ,LM
      JJ =JJ+1
      JD(JJ) =JD(I)
      NB(JJ) =NB(I)
      B(JJ) =B(I)
      NC(JJ) =NC(I)
16    CONTINUE
17    LM =LM-1
      IF (LM.EQ.MM) GO TO 21
      IF (LM.EQ.1) GO TO 2
C ******** CONSIDER ALL NODES WHOSE SHORTEST WEIGHTED DISTANCE WAS TO
C ******** THE ELIMINATED NODE. FIND A NEW SUCH DISTANCE TO ANOTHER NODE
      DO 20 I =1,LM
      IF (NB(I).NE.JR) GO TO 20
      JB =NC(I)
      AR =SY
      DO 18 J =1,LM
      JX =NC(J)
      IF (JB.EQ.JX) GO TO 18
      LL =ICHECK(JB,JX,M)
      IF (A(LL).GE.AR) GO TO 18
      AR =A(LL)
      IJ =JX
18    CONTINUE
      NB(I) =IJ
      B(I) =AR
20    CONTINUE
      GO TO 2
21    IF (MA.EQ.0) GO TO 30
C ******** DOES THIS SET OF P SATISFY TOTAL NETWORK DEMAND
C ******** IF NOT RE-CYCLE CONSIDERING ONLY NODES WITH PLANT SIZES > PM
      AR =0.0
      DO 22 I =1,JS
      J =NP(I)
22    AR =AR+UP(J)
      IF (AR.GE.DS) GO TO 30
23    JX =0
      DO 28 I =1,M
```

```
      J =MT(I)
      IF (MC(J))28,28,24
24    JJ =MC(J)
      IF (UP(JJ).GE.PM) GO TO 28
      J =IB(JJ)
      VR(J) =VR(J)+VR(JJ)
      IF (VR(J).GT.UP(J)) VR(J) =UP(J)
28    CONTINUE
      DO 25 I =1,M
      J =MT(I)
      IF (MC(J))27,25,26
26    JJ =MC(J)
      IF (UP(JJ).LT.PM) GO TO 25
27    JX =JX+1
      NB(JX) =IB(I)
      JD(JX) =MT(I)
      NC(JX) =KB(I)
      V(JX) =VR(I)
      B(JX) =G(I)
25    CONTINUE
      LM =JX
      GO TO 2
30    RETURN
      END
      SUBROUTINE SEARCH (N,M,A,B,IP,LC,NC,KC,MC,KG,UP,LI,JN,LX,SP,LZ,KT,
     1V)
C ******** THIS SUBROUTINE FINDS THE SET OF P NODES FOR THE FIRST AND
C ******** SECOND PARTITIONING HEURISTICS
      DIMENSION A(LI),B(N),NC(N),KC(M),MC(N),LC(M),KG(LZ),UP(M),JN(M)
      DIMENSION V(M)
      COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
      COMMON/COM2/AM,AR,DS,BD,SX,PM,SY
      IF (JS.EQ.KP) JS =JS-1
      K =1
C     *** INITIALIZE ARRAYS
      DO 4 I =1,M
      B(I) =SY
4     NC(I) =I
      RMAX =SY
      IF (JS.EQ.MM.OR.MM.EQ.0) GO TO 2
      JR =LC(JS)
      DO 3 I =1,MM
      II =LC(I)
      LL =ICHECK(II,JR,M)
      AP =A(LL)
      IF (KT.EQ.1) AP =A(LL)*V(II)
      IF (AP.LT.RMAX) RMAX =AP
3     CONTINUE
2     DO 6 I =1,JS
      L =LC(I)
6     NC(L) =0
```

```
      XX =0.0001
42    TMAX =0.0
C ******** THIS LOOP FINDS THE NEXT MEMBER OF THE P CANDIDATE SET FOR A
C ******** ROW
      DO 50 II =1,M
      IF (UP(II).LT.SP.OR.NC(II).EQ.0) GO TO 50
      DO 48 J =K,JS
      LS =LC(J)
      LL =ICHECK(II,LS,M)
      AP =A(LL)
      IF (KT.EQ.1) AP =A(LL)*V(II)
      IF (AP.GE.B(II)) GO TO 48
      B(II) =AP
48    CONTINUE
      IF (B(II)-TMAX)50,50,46
46    MR =II
      TMAX =B(II)
50    CONTINUE
      JS =JS+1
      LC(JS) =MR
      K =JS
      NC(MR) =0
      IF (JS-KP)42,10,10
10    IF (MA.EQ.0) GO TO 5
C ******** TEST WHETHER THIS SET OF P SATISFIES NETWORK DEMAND UNDER
C ******** PLANT CONSTRAINT IF NOT REPEAT CHOOSING FROM ONLY NODES > PM
      TSUM =0.0
      DO 1 I =1,JS
      JV =LC(I)
1     TSUM =TSUM+UP(JV)
      IF (TSUM-DS)75,5,5
5     KVP =1
      AR =TMAX
      IF (AR.GT.RMAX) AR =RMAX
      JK =0
      KJ =0
C ******** TEST WHETHER THIS P SET HAS > MAXIMUM BOUND THAN BEFORE
C ******** IF SO STORE THIS SOLUTION AND NEW BOUND
      IF (AR.GT.(AM-XX).AND.AR.LT.(AM+XX)) GO TO 7
      IF (AR-AM)75,7,9
7     JV =0
8     DO 11 I =1,JS
11    KC(I) =LC(I)
      CALL ASCEND (M,M,M,1,B,JS,KC,NC,1)
      CALL COMP (IP,JV,KC,KG,M,LZ)
      GO TO 75
9     AM =AR
      IP =LX
      IF (LX)7,72,7
72    JV =1
      GO TO 8
```

```
75      RETURN
        END
        SUBROUTINE TEST (N,M,IH,LI,A,B,KD,MC,NC,ID,V,Q,KC,KB,LR,NR)
C ******** THIS SUBROUTINE WHICH EXECUTES ONLY WHEN PLANT COSTS ARE
C ******** INCURRED ATTEMPTS NODE SHIFTING TO MAXIMIZE PLANT SCALE
C ******** ECONOMIES
        DOUBLE PRECISION TC,RC
        DIMENSION A(LI),KD(IH),NC(IH),B(IH),MC(N),ID(N),V(M),Q(M),KC(M)
        DIMENSION KB(M)
        COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
        COMMON/COM2/AM,AR,DS,BD,SX,PM,SY
        NR =0
5       MJ =0
        JX =LR
        DO 100 I =1,LR
        IF (KD(I).EQ.0) GO TO 100
C ******** GET FIRST CONTENDER FOR SWITCH
        JJ =KD(I)
        IF (MC(JJ).LT.0) GO TO 100
        JR =ID(JJ)
        MR =NC(I)
        AX =B(I)
        JZ =MR
C ******** EVALUATE COST REDUCTION OF REMOVING NODE FROM CURRENT MEDIAN
C ******** ENSURE AT LEAST 1 UNIT OF DEMAND IS LEFT AT MEDIAN
        IF ((V(MR)-AX).LE.1.0) AX =AX-1.0
        IF (AX.LE.0.0) GO TO 100
        AV =SY
C ******** TENTATIVELY ADD THIS NODE DEMAND TO SOME OTHER MEDIAN
        DO 80 J =1,JS
        IF (MR.EQ.J.OR.Q(J).LE.0.0) GO TO 80
        AY =AX
        IF (Q(J).LT.AY) AY =Q(J)
        SM =V(MR)-AY
        AM =V(MR)
        CALL SET (MR,J2,JR,KC,KB,LI,TC,M,A,AY,LV)
        CALL FCOST (AM,SM,LV,TC)
        AM =V(J)+AY
        SM =V(J)
        CALL SET (J,J2,JR,KC,KB,LI,RC,M,A,AY,LV)
        CALL FCOST (AM,SM,LV,RC)
C ******** COMPARE COST DIFFERENTIAL
        IF (RC-TC)50,80,80
50      BM =TC-RC
        IF (BM.LE.AV) GO TO 80
        MJ =1
        NR =1
        AV =BM
        JZ =J
        CM =AY
80      CONTINUE
```

```
C ******** IF COST REDUCTION OCCURS SWITCH NODES - MODIFY PLANT SIZES
      IF (JZ.EQ.MR) GO TO 100
      V(MR) =V(MR)-CM
      Q(MR) =Q(MR)+CM
      V(JZ) =V(JZ)+CM
      Q(JZ) =Q(JZ)-CM
      IF (AY.NE.B(I)) GO TO 60
      NC(I) =JZ
      GO TO 100
60    KD(I) =0
      IF (AX.NE.B(I)) AX =AX+1.0
      BM =AX-CM
      JX =JX+1
      KD(JX) =KD(JJ)
      B(JX) =BM
      NC(JX) =MR
      JX =JX+1
      KD(JX) =JJ
      B(JX) =CM
      NC(JX) =JZ
100   CONTINUE
      IF (MJ.EQ.0) GO TO 150
      LR =N
      J =N+1
      CALL SHRINK (J,JX,LR,KD,NC,B,IH)
      GO TO 5
150   RETURN
      END
      SUBROUTINE ASCEND (N,M,K,L,B,JS,LW,JD,KR)
C ******** A GENERAL PURPOSE ROUTINE FOR ORDERING DATA IN ASCENDING
C ******** ORDER. DATA IN ASSOCIATED ARRAYS MAY ALSO BE RE-ORDERED
      DIMENSION B(N),LW(M),JD(K)
      LJ =JS-1
      KJ =L
      IF (LJ.LT.L) GO TO 5
1     IND =1
      JND =1
      JK =0
      DO 4 I =KJ,LJ
      IF (KR.GE.1) GO TO 2
      IF (B(I+1).GE.B(I)) GO TO 7
      TEMP =B(I+1)
      NR =JD(I+1)
      B(I+1) =B(I)
      JD(I+1) =JD(I)
      B(I) =TEMP
      JD(I) =NR
      IF (KR)6,3,2
2     IF (LW(I+1).GE.LW(I)) GO TO 7
3     NS =LW(I+1)
      LW(I+1) =LW(I)
```

```
      LW(I) =NS
      IF (KR.LT.2) GO TO 6
      TEMP =B(I+1)
      B(I+1) =B(I)
      B(I) =TEMP
      IF (KR.LT.3) GO TO 6
      NR =JD(I+1)
      JD(I+1) =JD(I)
      JD(I) =NR
6     IND =I
      JK =1
      GO TO 4
7     IF (JK.EQ.1) GO TO 4
      JND =I
4     CONTINUE
      IF (IND.EQ.1) GO TO 5
      LJ =IND-1
      KJ =JND-1
      IF (KJ.LT.L) KJ =L
      GO TO 1
5     RETURN
      END
      SUBROUTINE ARK (LI,M,MGB,A,W,UP,V,LJ,KV,KR,LE,K)
C ******** THIS SUBROUTINE CALCULATES THE A , A1 MATRICES FOR THE  FIRST
C ******** TWO PARTITIONING HEURISTICS AND REPLACES SHORT PATH MATRIX IN
C ******** ARRAY A AT THE TERMINATION OF THE SEARCH
      DIMENSION A(LI),UP(M),W(M),V(M)
      COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
      COMMON/COM2/AM,AR,DS,BD,SX,PM,SY
      DOUBLE PRECISION S
      SR =1.0/SX
      IF (MGB.EQ.1) GO TO 55
      IF (K.EQ.1) GO TO 11
      DO 10 I =1,M
      DO 10 J =I,M
      LLL =J*(J-1)/2+I
 10   A(LLL) =A(LLL)*W(I)*W(J)/(W(I)+W(J))
11    IF (LJ.EQ.0) GO TO 30
      AX =0.0
      AV =0.0
      DO 15 I =1,M
      DO 15 J =I,M
      LL =J*(J-1)/2+I
      AW =A(LL)
      IF (K.EQ.1) AW =A(LL)*V(I)
      IF (AW.LE.AV) GO TO 12
      AV =AW
      KV =I
12    IF (MA.EQ.0) GO TO 15
      IF (UP(I).LT.PM.OR.UP(J).LT.PM) GO TO 15
      IF (AW.LE.AX) GO TO 15
```

```
      AX =AW
      KR =I
15    CONTINUE
      GO TO 30
55    DO 20 I =1,M
      DO 20 J =I,M
      LLL =J*(J-1)/2+I
      S =A(LLL)*(W(I)+W(J))/(W(I)*W(J))
      MAX =((S*SX)+0.5)
      S =MAX
 20   A(LLL) =S*SR
30    RETURN
      END
      SUBROUTINE COUNT (M,JS,LR,IH,NC,LW,KD,B,D,IN,MKB)
C ******** THIS SUBROUTINE ORDERS NODES BY SUBGRAPH AND SETS UP ARRAYS
C ******** FOR PENDING MEDIAN SEARCH
      DIMENSION MKB(M),NC(IH),KD(IH),LW(IH),B(IH),D(IH),IN(IH)
      DO 352 I =1,JS
352   MKB(I) =0
      DO 353 I =1,LR
      IF (KD(I).EQ.0) GO TO 353
      J =NC(I)
      MKB(J) =MKB(J)+1
353   CONTINUE
      L =0
      DO 354 I =1,JS
      LL =MKB(I)
      MKB(I) =L+1
354   L =L+LL
      DO 355 I =1,LR
      IF (KD(I).EQ.0) GO TO 355
      J =NC(I)
      K =MKB(J)
      LW(K) =KD(I)
      B(K) =D(I)
      IN(K) =I
      MKB(J) =K+1
355   CONTINUE
      RETURN
      END
      SUBROUTINE COMP (IP,JV,KC,KG,M,LZ)
C ******** THIS SUBROUTINE STORES THE SET OF P DERIVED FROM EACH
C ******** PARTITIONING HEURISTIC
      DIMENSION KG(LZ),KC(M)
      COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
      IF (JV.EQ.1) GO TO 20
      DO 16 LM =1,IP
      MJ =(LM*JS)-JS
      DO 15 LL =1,JS
      MJ =MJ+1
C ******** HAS THIS SET OF P BEEN PREVIOUSLY OBTAINED. IF SO RETURN
```

```
C ******** IF NOT STORE THE SET OF P
      IF (KC(LL)-KG(MJ))16,15,16
15    CONTINUE
      GO TO 30
16    CONTINUE
      IZP =IZP+1
      IF (IZP.GT.2) GO TO 30
20    MJ =IP*JS
      IP =IP+1
      DO 8 I =1,JS
      MJ =MJ+1
8     KG(MJ) =KC(I)
30    RETURN
      END
      SUBROUTINE TRANS (N,M,V,NZ,NC,MB,KD,D,QS,VR,IZ,NB,KB,MD,CJ,Q,JS,LS
     1,LR,IH,K,DN,F)
C ******** THIS SUBROUTINE STORES THE CURRENT LOWER BOUND SOLUTION
      DIMENSION NC(IH),NB(IH),MD(IH),KD(IH),D(IH),CJ(IH),Q(M),QS(M)
      DIMENSION V(M),VR(M),IZ(M),NZ(M),KB(M),MB(M),F(M),DN(M)
      DO 5 I =1,JS
      F(I) =DN(I)
      VR(I) =V(I)
      Q(I) =QS(I)
5     IZ(I) =NZ(I)
      DO 10 I =1,LS
10    KB(I) =MB(I)
      IF (K.EQ.0) GO TO 35
      DO 20 I =1,LR
      NB(I) =NC(I)
      MD(I) =KD(I)
20    CJ(I) =D(I)
35    RETURN
      END
      SUBROUTINE SET (MR,J2,JR,KC,KB,LI,TC,M,A,SUMMA,LV)
      DOUBLE PRECISION TC
      DIMENSION A(LI),KC(M),KB(M)
      LJ =KC(MR)
      LV =KB(LJ)
      IF (J2.EQ.0) GO TO 90
      CALL TCOST (SUMMA,TC,LV)
      GO TO 92
90    TC =A(ICHECK(JR,LV,M))
92    TC =TC*SUMMA
      RETURN
      END
      SUBROUTINE INCRE (JM,JL,JG,JA,JQ,RS,JJ,SUMMA,MR,K,M7,M8)
C ******** THIS SUBROUTINE INCREMENTS SPECIFIED ARRAYS WITH NON UNIQUELY
C ******** ASSIGNED NODES OR NODES WITH FRACTIONALLY ASSIGNED DEMANDS
C ******** ARRAY JG CONTAINS THE NODE LABEL,JQ THE RELEVENT SUBGRAPHS,JA
C ******** THE NO OF SUBGRAPHS COMMON TO EACH MEMBER OF JG, AND ARRAY RS
C ******** THE AMOUNT OF MARKET DEMAND ALLOCATED
```

```
      DIMENSION JG(M7),JA(M7),RS(M7),JQ(M8)
      IF (K.EQ.1) GO TO 5
      JM =JM+1
5     JL =JL+1
      JG(JM) =JJ
      RS(JM) =SUMMA
      JQ(JL) =MR
      IF (K.EQ.2) GO TO 10
      JA(JM) =JL
10    RETURN
      END
      FUNCTION ICHECK (I,J,M)
C ******** THIS FUNCTION SUBROUTINE FINDS APPROPRIATE I,J ELEMENTS IN
C ******** THE SINGLY DIMENSIONED SHORT PATH MATRIX A
      COMMON/COM1/ MM,KP,MA,NM,JS,KY,J2,MRJ,MY,KVP,IKP,IZP,MG,IKS
      IF (I.LE.M) GO TO 2
      ICHECK =IKP+((I-M)*M)+J
      RETURN
2     IF (I.GT.J) GO TO 5
      ICHECK =J*(J-1)/2+I
      RETURN
5     ICHECK =I*(I-1)/2+J
      RETURN
      END
      SUBROUTINE SHRINK (MJ,LR,JJ,KD,NC,D,IH)
C ******** THIS SUBROUTINE REWRITES ARRAYS AFTER NODE REASSIGNMENTS HAVE
C ******** MADE SOME ARRAY ELEMENT REDUNDANT
      DIMENSION KD(IH),NC(IH),D(IH)
      DO 10 I =MJ,LR
      IF (KD(I).EQ.0) GO TO 10
      JJ =JJ+1
      KD(JJ) =KD(I)
      NC(JJ) =NC(I)
      D(JJ) =D(I)
10    CONTINUE
      RETURN
      END
      SUBROUTINE COST (SUM,P,LR)
C ******** OPTIONAL USER SUPPLIED SUBROUTINE FOR CALCULATING PLANT
C ******** COSTS BASED UPON CURRENT SIZE OF THAT PLANT
      COMMON NF(10)
      DOUBLE PRECISION P
      IF (NF(LR).NE.1) GO TO 2
      P =7*SQRT(SUM)
      GO TO 5
2     P =10*SQRT(SUM)
5     RETURN
      END
      SUBROUTINE FCOST (S,SUM,LV,TC)
      DOUBLE PRECISION X,XX,TC
      CALL COST (S,XX,LV)
```

```
      CALL COST (SUM,X,LV)
      TC =TC+(XX-X)
      RETURN
      END
      SUBROUTINE DATA
C ******** THIS OPTIONAL USER SUPPLIED SUBROUTINE READS IN PLANT COST
C ******** DATA FOR EACH PLANT. THIS INFORMATION MAY BE SUPPLIED TO
C ******** SUBROUTINE COST VIA A SUITABLE COMMON STATEMENT
      COMMON NF(10)
      READ (5,1) (NF(I),I =1,10)
1     FORMAT (10I1)
      RETURN
      END
      SUBROUTINE TCOST (SUMR,R,JJ)
C ******** THIS OPTIONAL USER SUPPLIED SUBROUTINE RE-CALCULATES UNIT
C ******** TRANSPORT COSTS OVER SOME LINK IF PLANT SIZE CONSTRAINTS
C ******** MAKE THIS NECESSARY
      RETURN
      END
```

APPENDIX VII

A Computer Program for the Optimal Sequencing of Plant Locations Over Time by Dynamic Programming

```
C      ***********************************************************************
C  *** A DYNAMIC PROGRAM FOR THE OPTIMAL SEQUENCING OF PLANT LOCATIONS
C  *** OVER TIME
C  *** METHOD: SEE 'AN ALGORITHM FOR ESTIMATING THE MEDIANS OF A WEIGHTED
C  *** GRAPH SUBJECT TO SIDE CONSTRAINTS' - R.A.WHITAKER PHD THESIS
C  *** DEPARTMENT OF GEOGRAPHY UNIVERSITY OF BRITISH COLUMBIA 1971
C      ***********************************************************************
C  *** N.B. NETWORK NODES SHOULD BE LABELLED IN SEQUENCE 1,...,N
C      *****DECK SET UP AS FOLLOWS*****
C  *** A). PROBLEM TITLE CARD MAXIMUM OF 80 COLUMNS ALLOWED
C  *** B). CONTROL CARD
C  ***     1.. COLS 1-3 NUMBER OF NODES ON THE NETWORK (N)
C  ***     2.. COLS 4-6 NUMBER OF TIME PERIODS. (NP) N.B. THE ASSUMPTION
C  ***      IS MADE THAT ONE PLANT IS LOCATED IN EACH TIME PERIOD
C  ***     3.. COLS 7-9 NUMBER OF PLANTS CURRENTLY IN OPERATION OVER THE
C  ***      NETWORK, IF ANY. (NK) N.B. IT IS ASSUMED THAT THE NUMBER OF
C  ***      PLANTS AS SPECIFIED IN COLS 4-6 IS TO BE ADDED SEQUENTIALLY
C  ***      WITH RESPECT TO THOSE ALREADY LOCATED (I.E. IN OPERATION).
C  ***     4.. COL 10 LEAVE BLANK IF PLANT COSTS ARE NOT INCLUDED.PUNCH 1
C  ***      IF THE PLANT COST FUNCTIONS (WHICH ARE USER SUPPLIED) ARE THE
C  ***      SAME OVER ALL TIME PERIODS. PUNCH 2 IF A DIFFERENT SET OF
C  ***      PLANT COST FUNCTIONS IS TO BE READ IN FOR EACH TIME PERIOD
C  ***      SUBROUTINE DATA WILL BE NEEDED IF A 2 IS PUNCHED. (SEE SECTION
C  ***      BELOW ON USER SUPPLIED SUBROUTINES). N.B. THE COST
C  ***      FUNCTIONS SHOULD BE OF THE SAME GENERAL TYPE ALTHOUTH
C  ***      INDIVIDUAL PARAMETERS MAY CHANGE OVER TIME. IF THE USER WANTS
C  ***      TO SUPPLY DIFFERENT TYPES OF COST FUNCTIONS HE MUST DEFINE
C  ***      SEVERAL SETS OF DATA AND COST SUBROUTINES AND CALLING
C  ***      STATEMENTS IN THE MAIN PROGRAM.
C  ***     5.. COL 11 PUNCH 1 IF USER SUPPLIED SUBROUTINE FOR READING IN
C  ***      PLANT COST FUNCTIONS IS SUPPLIED: OTHERWISE LEAVE BLANK (MW).
C  ***      SEE SECTION ON USER SUPPLIED SUBROUTINES.
C  ***     6.. COL 12 PUNCH 1 IF A DIFFERENT SET OF MARKET DEMANDS (NODE
C  ***      WEIGHTS) IS TO BE READ IN FOR EACH TIME PERIOD: OTHERWISE
C  ***      LEAVE BLANK (NZ)
C  ***     7.. COL 13 LEAVE BLANK IF THE MATRIX OF UNIT TRANSPORT COSTS
C  ***      (DISTANCES) FROM EACH NODE TO EACH NETWORK MEDIAN IS TO REMAIN
C  ***      THE SAME OVER ALL TIME PERIODS. PUNCH 1 IF A DIFFERENT SET OF
C  ***      TRANSPORT COSTS (DISTANCES) IS TO BE READ IN FOR EACH TIME
C  ***      PERIOD (IKK)
C  ***     8.. COL 14 PUNCH 0 IF THE SHORT PRINT OUT IS OPTIONED. PUNCH 1
C  ***      FOR THE FULL PRINT OUT WHICH INCLUDES MEDIAN PLANT SIZES,
C  ***      TRANSPORT (DISTANCES) AND PLANT COSTS, AND NODE TO MEDIAN
C  ***      ALLOCATIONS. (KRZ)
C  ***     9.. COL 15 SINCE AS MANY JOBS AS ARE REQUIRED MAY BE RUN
C  ***      CONSECUTIVELY PUNCH 1 IF THIS IS THE LAST (OR ONLY) JOB IN THE
C  ***      RUN (LFIN)
C  *** C). READ IN THE NODE DESIGNATIONS (LABELS) OF ALL PLANTS INCLUDING
C  ***      BOTH THOSE TO BE FORCED INTO SOLUTION,IF ANY, (I.E. THE
C  ***      EXISTING LOCATIONS ON THE NETWORK) AND THOSE TO BE SEQUENCED
C  ***      OVER TIME. N.B. THE ORDER FOR THESE WILL NORMALLY BE IN
```

```
C ***      ASCENDING ORDER. (SEE INSTRUCTION F FOR FURTHER CLARIFICATION
C *** D).  READ IN SPECIFICATION INDICATORS FOR PLANT STATUS AS FOLLOWS:
C ***      (THERE WILL BE AS MANY INDICATORS AS THERE ARE NODES READ IN
C ***      UNDER INSTRUCTION C). PUNCH 1 IF THE CORRESPONDING NODE LISTEI
C ***      IN C IS A CANDIDATE MEDIAN FOR SEQUENTIAL LOCATION. PUNCH 2 II
C ***      THE CORRESPONDING NODE IN C HAS FORCED STATUS (I.E. IS AN
C ***      EXISTING LOCATION IN OPERATION) E.G. SUPPOSE THE NODES READ II
C ***      UNDER INSTRUCTION C ARE 4,15,21, AND 34 REPECTIVELY AND
C ***      FURTHER THAT NODE 21 IS FORCED. THEN MEDIANS 4,15, AND 34 ARE
C ***      TO BE SEQUENTIALLY LOCATED OVER 3 TIME PERIODS WITH RESPECT TC
C ***      NODE 21 WHICH HAS BEEN LOCATED IN SOME PREVIOUS TIME PERIOD.
C ***      THE LIST OF SPECIFICATION INDICATORS WILL THEREFORE BE 1,1,2,
C ***      AND 1. PUNCH THE REQUISITE 1 OR 2 IN EVERY FOURTH COLUMN.
C *** E).  READ IN VARIABLE FORMAT CARD FOR READING IN THE UNIT TRANSPORT
C ***      COST (SHORT DISTANCE) MATRIX FROM EACH NODE TO EACH MEMBER OF
C ***      THE LIST SPECIFIED IN INSTRUCTION C. E.G. (10F5.1)
C *** F).  READ IN THE SHORT PATH MATRIX UNDER THE SPECIFIED VARIABLE
C ***      FORMAT GIVEN IN (E).IF A DIFFERENT SET OF TRANSPORT COSTS
C ***      (DISTANCES) IS SPECIFIED FOR EACH TIME PERIOD THEN READ IN THE
C ***      SHORT PATH MATRICES CONSECUTIVELY I.E. THE FIRST ONE WILL
C ***      REFER TO TIME PERIOD ONE AND THE LAST SET TO THE LAST TIME
C ***      PERIOD. N.B. **WARNING** FOR EACH MATRIX THERE WILL BE N ROWS,
C ***      ONE FOR EACH NETWORK NODE AND M COLUMNS, ONE FOR EACH OF THE
C ***      NODES CONTAINED IN THE LIST READ IN UNDER INSTRUCTION C. THE
C ***      JTH OF THE M COLUMNS IN THE A(I,J) SHORT PATH MATRIX MUST
C ***      REFERENCE THE DISTANCE FROM NODE I TO THE JTH NODE SPECIFIED
C ***      IN THE LIST READ IN UNDER INSTRUCTION C. - E.G. ELEMENT A(1,5)
C ***      MUST BE THE DISTANCE FROM THE FIRST NODE TO WHICHEVER NODE IS
C ***      THE FIFTH MEMBER OF THE SUBSET DEFINED IN C.
C *** G).  READ IN VARIABLE FORMAT FOR READING IN MARKET DEMANDS (NODE
C ***      WEIGHTS)
C *** H).  READ IN MARKET DEMANDS (NODE WEIGHTS) UNDER THE VARIABLE
C ***      FORMAT SPECIFIED IN G. - N.B. IF COL 12 (INSTRUCTION B) IS
C ***      PUNCHED WITH A ONE THEN AS MANY SETS OF WEIGHTS AS THERE ARE
C ***      TIME PERIODS MUST BE READ IN. THE FIRST SET READ IN WILL REFER
C ***      TO TIME PERIOD ONE, THE SECOND SET TO TIME PERIOD 2 AND SO ON.
C ***      IF COL 12 (INSTRUCTION B) IS NOT PUNCHED WITH A ONE THEN ONLY
C ***      ONE SET OF WEIGHTS WILL BE READ IN.
C *** K).  READ IN PLANT COST FUNCTIONS VIA SUBROUTINE DATA IF OPTIONED.
C ***      IF A DIFFERENT SET OF FUNCTIONS IS SUPPLIED FOR EACH TIME
C ***      PERIOD THEN THE FIRST SET READ IN MUST REFER TO THE LAST TIME
C ***      PERIOD, THE SECOND SET TO THE NEXT TO LAST TIME PERIOD, ETC
C ***      DOWN TO THE LAST SET WHICH REFERS TO THE FIRST TIME PERIOD.
C *** L).  IF THE LONG PRINT OUT IS OPTIONED (I.E. B.7 COL 13 PUNCHED
C ***      WITH A 1) AND IF A DIFFERENT SET OF PLANT COST FUNCTIONS IS
C ***      SPECIFIED FOR EACH TIME PERIOD THEN READ IN THE PLANT
C ***      COST FUNCTIONS VIA SUBROUTINE DATA AGAIN, BUT THIS TIME IN
C ***      REVERSE ORDER. I.E THE FIRST SET OF FUNCTIONS READ IN WILL
C ***      REFER TO THE FIRST TIME PERIOD, THE SECOND TO THE SECOND TIME
C ***      PERIOD AND SO ON DOWN TO THE LAST SET WHICH WILL REFERENCE
C ***      THE LAST TIME PERIOD.
```

```
C       ******************* USER SUPPLIED SUBROUTINES *******************
C  ***  THE FOLLOWING 2 SUBROUTINES MUST BE DEFINED BY THE USER IF PLANT
C  ***  COSTS ARE INVOKED.
C  ***  1). SUBROUTINE DATA ---
C  ***     THIS SUBROUTINE IS USED TO READ IN PLANT COST FUNCTIONS FOR
C  ***     EACH POTENTIAL PLANT SITE WHERE THE FUNCTIONS VARY AMONG PLANT
C  ***     CONSIDER THE FOLLOWING EXAMPLE: SUPPOSE EACH POTENTIAL SITE HA
C  ***     A COST FUNCTION OF THE FORM Y= A+BX WHERE A AND B ARE CONSTANT
C  ***     X VARIES ACCORDING TO PLANT SIZE AND THERE ARE 10 SUCH SITES
C  ***     SUBROUTINE DATA
C  ***     COMMON DAT (2,10)
C  ***     READ (5,1) A,B
C  *** 1   FORMAT (2F5.0)
C  ***     DO 5 I =1,10
C  ***     DAT(1,I) =A
C  ***     DAT(2,I) =B
C  *** 5   CONTINUE
C  ***     RETURN
C  ***     END
C  ***     IN THE FIRST ROW OF MATRIX DAT ALL THE A CONSTANTS ARE STORED
C  ***     IN THE SECOND ROW OF MATRIX DAT ALL THE B CONSTANTS ARE STORED
C  ***     THE CONTENTS OF DAT ARE TRANSFERRED TO SUBROUTINE COST BY THE
C  ***     COMMON STATEMENT
C  ***     N.B. THIS SUBROUTINE CAN BE SUPPLIED AS A DUMMY PROVIDING THE
C  ***     PLANT COST FUNCTIONS ARE IDENTICAL FOR ALL PLANTS OVER ALL
C  ***     TIME PERIODS. IN SUCH A CASE THE COST FUNCTION CAN BE DIRECTLY
C  ***     DEFINED IN SUBROUTINE COST
C  *** 2). SUBROUTINE COST ---
C  ***     THIS SUBROUTINE CALCULATES THE COST FUNCTION SUPPLIED BY THE
C  ***     USER. ITS ARGUMENT LIST MUST CONTAIN THE FOLLOWING; SUM THE
C  ***     CURRENT PLANT SIZE WHICH IS CALCULATED IN THE MAIN PROGRAM, LR
C  ***     THE INDEX REFERRING TO SOME PLANT, AND Y THE COST OF PRODUCING
C  ***     AMOUNT SUM AT SOURCE LR. - E.G. SUPPOSE THE COST FUNCTION IS OF
C  ***     THE FORM Y =A+BX, AND THAT THE CONSTANTS A AND B ARE NOT
C  ***     IDENTICAL FOR EACH PLANT. THEN, ASSUMING THIS INFORMATION TO
C  ***     HAVE BEEN READ IN VIA SUBROUTINE DATA AS JUST DEMONSTRATED WE
C  ***     HAVE
C  ***     SUBROUTINE COST (SUM,Y,LR)
C  ***     DOUBLE PRECISION Y
C  ***     COMMON DAT (2,10)
C  ***     Y =DAT(1,LV) + (DAT(2,LV)*SUM)
C  ***     RETURN
C  ***     END
C  ***     IF PLANT COSTS ARE THE SAME OVER ALL PLANTS THEN THE INDEX LV
C  ***     WILL NOT BE SPECIFICALLY ACTIVATED BUT IS NEEDED IN THE
C  ***     THE SUBROUTINE ARGUMENT LIST. N.B. VARIABLE Y MUST ALWAYS BE
C  ***     DOUBLE PRECISIONED.
C       ******************* DIMENSIONING INFORMATION *******************
C  *** DIMENSION SIZE OF ARRAY AS MUST BE GREATER THAN OR EQUAL TO
C  *** (N*M*KS) WHERE N IS THE NUMBER OF NETWORK NODES, M IS THE NUMBER
C  *** OF MEDIANS OR COLUMNS IN THE SHORT PATH MATRIX, AND KS IS THE
```

```
C *** NUMBER OF SHORT PATH MATRICES READ IN.
C *** DIMENSION SIZE OF ARRAY B MUST BE GREATER THAN EQUAL TO (N*NP)
C *** WHERE N IS THE NUMBER OF NETWORK NODES AND NP IS THE NUMBER OF
C *** DIFFERENT SETS OF NODE WEIGHTS READ IN.
C *** DIMENSION SIZE OF ARRAYS DS,DC,NB,MB,MUST BE GREATER THAN OR
C *** EQUAL TO 1 PLUS THE SUM OF ALL THE ENTERING STATES OVER ALL THE
C *** TIME PERIODS
C *** DIMENSION SIZE OF ARRAYS KC,JC,CV,AND CJ MUST BE GREATER THAN OR
C *** EQUAL TO THE LARGER OF A) THE MAXIMUM OF (KK*J) WHERE KK IS THE
C *** NUMBER OF ENTERING STATES IN SOME TIME PERIOD AND J IS THE NUMBER
C *** OF TERMS IN THAT COMBINATORIAL SERIES OR B) N THE NUMBER OF
C *** NETWORK NODES. DIMENSION TO THE LARGER OF THESE TWO VALUES. N.B.
C *** IF THE NUMBER OF TIME PERIODS IS 10 THEN THE MAXIMUM OF THE
C *** EXPRESSION (KK*J) WILL BE LESS THAN 4000
C *** DIMENSION SIZE OF ARRAYS MC,KB,LB,JQ,P,E,AND V MUST BE GREATER
C *** THAN OR EQUAL TO M WHERE M IS AS PREVIOUSLY DEFINED
C *** DIMENSION SIZE OF ARRAY JN MUST BE GREATER THAN OR EQUAL TO N
C *** SIZE OF ALL OTHER ARRAYS IS AS DEFINED
C     ***************************************************************
C ***
C *** PROGRAMMER - R.A.WHITAKER - DEPARTMENT OF GEOGRAPHY,UNIVERSITY OF
C *** BRITISH COLUBIA, JANUARY 1971
C ***
C     ***************************************************************
      DIMENSION IAC(5),MAC(5),IMT(15),LMT(9),FMT(20),PROB(20),MC(50)
      DIMENSION AS(1000),KB(50),MB(1000),B(1000),LB(50),DC(1000)
      DIMENSION DS(500),JC(500),KC(500),JQ(50),NB(200),JN(200),P(50)
      DIMENSION E(50),V(50),NH(65),CV(2000),CJ(2000),FMT1(20)
      DIMENSION BS(1000)
      DIMENSION LX(50),MX(50)
      DATA JW,JK/1H*,5/
      DATA IAC/1H1,1H2,1H3,1H4,1H5/
      DATA MAC/1H4,1H8,2H12,2H16,2H20/
      DATA IMT/4H(I4,,3HI7,,4HF12.,3H2,F,4H14.2,4H,2F1,4H2.2,,3H3X,,1H5,
     13HI4,,1H4,2HX,,1H5,4HF8.1,1H)/
      DATA LMT/3H(64,2HX,,1H5,3HI4,,1H4,2HX,,1H5,4HF8.1,1H)/
      EQUIVALENCE (CV(1),KC(1)),(CJ(1),JC(1))
3     FORMAT (1H1)
4     FORMAT (36X,25HPLANT LOCATION-ALLOCATION,I3,16H MEDIAN SOLUTION,14
     1H - TIME PERIOD,I3/)
5     FORMAT (1H1,19HPROBLEM NAME ..... ,20A4//)
7     FORMAT (65A1,65A1/)
8     FORMAT (5X,6HMEDIAN,2X,10HPLANT SIZE,2X,12HFREIGHT COST,2X,10HPLAN
     1T COST,2X,10HTOTAL COST,6X,14HNODES SERVICED,12X,30HVOLUME SUPPLIE
     2D TO EACH MEDIAN/)
10    FORMAT (3I3,6I1)
11    FORMAT (4X,7HNETWORK,F12.2,F14.2,2F12.2/)
12    FORMAT (1X,37HOPTIMUM SEQUENCING OF PLANT LOCATIONS//)
13    FORMAT (1X,11HTIME PERIOD,3X,13HPLANT LOCATED,3X,11HPERIOD COST,3X
     1,15HCUMULATED COSTS/)
14    FORMAT (8X,I4,12X,I4,3X,F11.2,6X,F12.2)
```

```
15      FORMAT (20I4)
25      FORMAT (20A4)
1       READ (5,25) (PROB(I),I =1,20)
        WRITE (6,5) (PROB(I),I =1,20)
        READ (5,10) N,NP,NK,LY,MW,NZ,IKK,KRZ,LFIN
C ******** READ IN NODE DESIGNATIONS (LABELS) OF BOTH FORCED PLANTS,IF
C ******** ANY,(I.E.EXISTING LOCATIONS ON THE NETWORK), AND THE PLANTS
C ******** TO BE ADDED TO THIS SYSTEM SEQENTIALLY OVER TIME
        NW =NP+NK
        KW =NW
        READ (5,15) (MC(I),I =1,NW)
C ******** READ IN SPECIFICATIONS TO INDICATE WHETHER SOME PLANT IS TO
C ******** TO BE REFERENCED AS FORCED  OR IS TO BE SEQUENCED OVER TIME
        READ (5,15) (MB(I),I =1,NW)
C ******** SET UP INDEXING SYSTEM FOR REFERENCING PLANT STATUS
        J =0
        JJ =0
        DO 20 I =1,NW
        IF (MB(I).EQ.1) GO TO 18
        J =J+1
        KB(J) =I
        MX(J) =I
        GO TO 20
18      JJ =JJ+1
        LX(JJ) =I
20      CONTINUE
C ******** READ IN FORMAT FOR SHORT PATH MATRIX
        READ (5,25) (FMT(I),I =1,20)
C ******** READ IN SHORT PATH MATRIX -- ALL N NETWORK NODES TO BOTH
C ******** FORCED PLANTS,IF ANY, AND PLANTS TO BE SEQUENCED OVER TIME
        MM =1
        IF (IKK.NE.0) MM =NP
        JM =0
        DO 28 II =1,MM
        DO 28 I =1,N
        READ (5,FMT) (DC(J),J =1,NW)
        DO 28 J =1,NW
        JM =JM+1
        AS(JM) =DC(J)
28      CONTINUE
        MY =1
        IF (NZ.NE.0) MY =NP
C ******** READ IN SET(S) OF NODE WEIGHTS
        READ (5,25) (FMT1(I),I =1,20)
        JS =0
        DO 30 I =1,MY
        READ (5,FMT1) (DC(J),J =1,N)
        DO 30 J =1,N
        JS =JS+1
        B(JS) =DC(J)
30      CONTINUE
```

```
      IF (MW.EQ.0) GO TO 35
      CALL DATA
35    SX =10.0**15
C ******** CALCULATE OBJECTIVE FUNCTION FOR THE LAST TIME PRIOD;I.E.FOR
C ******** THE PLANNING HORIZON BASED UPON THE SET OF PREDETERMINED
C ******** LOCATIONS WHOSE TIME SEQUENCING IS SOUGHT
      J =NK
      DO 32 I =1,NP
      J =J+1
      KB(J) =LX(I)
32    CONTINUE
      CALL ALLOC (N,P,V,E,B,XR,AS,KB,LY,MY,KW,JM,SX,J,JN,JS,MM)
C ******** INITIALIZE INDEXES AND FIRST EXIT STATE
      NX =1
      NA =NP-1
      NR =1
      LR =0
      LZ =NP
      NY =NP-1
      DC(NR) =XR
      DS(NR) =XR
      DO 40 I =1,NP
40    JC(I) =I
      CALL FAC(NP,LP)
      DO 95 II =1,NY
      MK =LZ-1
      IF (NZ.NE.0) MY =MY-1
      IF (IKK.NE.0) MM =MM-1
      IF (LY.NE.2) GO TO 44
      CALL DATA
44    CALL FAC (MK,LJ)
      JJ =NP-MK
      CALL FAC(JJ,IJ)
      LW =LP/(LJ*IJ)
      LJ =NR
      J =MK
C ******** SET DECISION CELLS FOR THIS STAGE TO AN ARBITRARILY HIGH VALU
      DO 45 I =1,LW
      LJ =LJ+1
      DS(LJ) =SX
45    CONTINUE
C ******** DEFINE SOME ENTERING STATE AND EVALUATE ALLOCATIONS TO THIS
C ******** COMBINATION OF LOCATIONS
      JV =0
      MRJ =0
      MT =1
      MA =1
50    JQ(MT) =MA
      IF (MT.EQ.MK) GO TO 55
      MR =MA+1
      IF (MR.GT.NP) GO TO 70
```

```
      MA =MR
      MT =MT+1
      GO TO 50
55    JX =NK
      DO 60 I =1,MK
      J =JQ(I)
      JX =JX+1
      KB(JX) =LX(J)
60    CONTINUE
C ******** ASSIGN NODES TO CURRENT MEDIAN SET AND CALCULATE OBJECTIVE
C ******** FUNCTION
      CALL ALLOC (N,P,V,E,B,S,AS,KB,LY,MY,KW,JM,SX,JX,JN,JS,MM)
      MRJ =MRJ+1
      BS(MRJ) =S
C ******** STORE THIS ENTERING STATE
      DO 63 I =1,MK
      JV =JV+1
      KC(JV) =JQ(I)
63    CONTINUE
C ******** GET NEXT COMBINATION
65    MA =JQ(MT)+1
      IF (MA.LE.NP) GO TO 50
70    MT =MT-1
      IF (MT)65,72,65
72    MRJ =0
      KSJ =0
      NV =MK+1
      MJ =LR
C ******** GET DESCENDENT COMBINATIONS OF EACH ENTERING STATE BY SOME
C ******** EXIT STATE
      DO 90 IV =1,NX
      KSJ =KSJ+LZ
      KX =KSJ+1
      MT =1
      MA =1
75    JQ(MT) =MA
      IF (MT.EQ.MK) GO TO 78
      MR =MA+1
      IF (MR.GT.LZ) GO TO 87
      MA =MR
      MT =MT+1
      GO TO 75
78    DO 79 I =1,MK
      J =JQ(I)+MRJ
      JN(I) =JC(J)
79    CONTINUE
C ******** GET RANK OF COMBINATION
      ISUM =0
      DO 80 I =1,MK
      LJ =NA-(JN(NV-I)-1)
      IF (LJ.LT.I) GO TO 80
```

```
      IJ =LJ-I
      CALL FAC (LJ,KK)
      CALL FAC (I,K)
      CALL FAC (IJ,L)
      ISUM =ISUM+KK/(L*K)
80    CONTINUE
      KR =LW-ISUM
C ******** CHECK LOWER BOUND IN APPROPRIATE DECISION CELL--UPDATE
C ******** DECISION CELL,COST CELL,LEFT AND RIGHT INSET BOXES
      S =BS(KR)
      LJ =KR+NR
      KJ =MJ+IV
      KX =KX-1
      SS =S+DS(KJ)
      IF (SS.GE.DS(LJ)) GO TO 85
      DS(LJ) =SS
      DC(LJ) =S
      MB(LJ) =IV
      NB(LJ) =JC(KX)
C ******** GET NEXT DESCENDENT COMBINATION
85    MA =JQ(MT)+1
      IF (MA.LE.LZ) GO TO 75
87    MT =MT-1
      IF (MT)85,88,85
88    MRJ =MRJ+LZ
90    CONTINUE
C ******** REPLACE EXIT STATES WITH ENTERING STATES READY FOR STAGE II+1
      LJ =LW*MK
      DO 92 I =1,LJ
      JC(I) =KC(I)
92    CONTINUE
C ******** UPDATE OTHER INDICES FOR NEW STAGE
      LB(II+1) =LW
      LZ =LZ-1
      LR =LR+NX
      NX =LW
      NR =NR+LW
95    CONTINUE
C ******** TRACE OPTIMAL SEQUENCING PATH
      JJ =0
      J =LB(NP)
      JX =(NR-J)+1
      SUM =SX
      DO 97 I =JX,NR
      JJ =JJ+1
      IF (DS(I).GE.SUM) GO TO 97
      SUM =DS(I)
      JV =JJ
97    CONTINUE
      LJ =1
      JQ(LJ) =JV
```

```
      JJ =NP+1
      SJ =0.0
      DO 100 I =1,NY
      JJ =JJ-1
      JR =NR-LB(JJ)
      JX =JR+JV
      NR =JR
      LJ =LJ+1
      JQ(LJ) =NB(JX)
      CJ(I) =DC(JX)
      CV(I) =CJ(I)+SJ
      SJ =CV(I)
      JV =MB(JX)
100   CONTINUE
      CJ(NP) =DC(1)
      CV(NP) =DC(1)+SJ
C ******** PRINT OPTIMAL SEQUENCING ROUTE
      WRITE (6,12)
      WRITE (6,13)
      DO 105 I =1,NP
      J =JQ(I)
      K =LX(J)
      WRITE (6,14) I,MC(K),CJ(I),CV(I)
105   CONTINUE
C ******** CHECK OPTION FOR PRINTING MARKET ALLOCATIONS IN EACH PERIOD
C ******** IF NOT ACTIVATED GET NEXT PROBLEM
      IF (KRZ.EQ.0) GO TO 180
      WRITE (6,3)
      IF (LY.NE.0) GO TO 115
      DO 110 I =1,NW
      E(I) =0.0
110   CONTINUE
115   DO 120 I =1,65
120   NH(I) =JW
      WRITE (6,7) (NH(I),I =1,65),(NH(I),I =1,65)
      IV =1
      MM =1
      DO 175 II =1,NP
C ******** GET OPTIMAL ALLOCATIONS FOR TIME PERIOD II IN THIS LOOP
      IF (NK.EQ.0) GO TO 123
      DO 122 I =1,NK
      KB(I) =MX(I)
122   CONTINUE
123   IF (NZ.NE.0) IV =II
      IF (IKK.NE.0) MM =II
      IF (LY.NE.2) GO TO 124
      CALL DATA
124   KS =(IV*N)-N
      NW =NK
      DO 125 I =1,II
      J =JQ(I)
```

```
      NW =NW+1
      KB(NW) =LX(J)
125   CONTINUE
C ******** ARRANGE MEDIANS IN ASCENDING ORDER
      LJ =NW-1
      IF (LJ.EQ.0) GO TO 136
130   IN =1
      DO 135 I =1,LJ
      IF (KB(I+1).GE.KB(I)) GO TO 135
      K =KB(I+1)
      KB(I+1) =KB(I)
      KB(I) =K
      IN =I
135   CONTINUE
      IF (IN.EQ.1) GO TO 136
      LJ =IN-1
      GO TO 130
C ******** DEFINE NODE TO MEDIAN ALLOCATIONS - CALCULATE OBJECTIVE
C ******** FUNCTION
136   CALL ALLOC (N,P,V,E,B,S,AS,KB,LY,IV,KW,JM,SX,NW,JN,JS,MM)
C ******** ORDER OPTIMAL ALLOCATIONS FOR PRINTING
      DO 140 I =1,NW
      LB(I) =0
140   CONTINUE
      DO 145 I =1,N
      J =JN(I)
      LB(J) =LB(J)+1
145   CONTINUE
      L =0
      DO 150 I =1,NW
      LL =LB(I)
      LB(I) =L+1
      L =L+LL
150   CONTINUE
      DO 155 I =1,N
      J =JN(I)
      K =LB(J)
      KS =KS+1
      JC(K) =I
      CV(K) =B(KS)
      LB(J) =K+1
155   CONTINUE
C ******** PRINT OUT ALLOCATIONS
      WRITE (6,4) NW,II
      WRITE (6,8)
      S =0.0
      S1 =0.0
      S2 =0.0
      RS =0.0
      JZ =0
      JR =0
      DO 170 I =1,NW
      JV =JZ+1
```

```
      JZ =LB(I)-1
      K =KB(I)
      RC =P(I)+E(I)
      JJ =(JZ-JV)+1
      JY =JK
      IF (JY.GT.JJ) JY =JJ
      JX =1
      JP =(JY-JX)+1
      LP =(JK-JP)+1
      JZ =(JV+JY)-1
      IMT(9) =IAC(JP)
      IMT(11) =MAC(LP)
      IMT(13) =IAC(JP)
      WRITE (6,IMT) I,MC(K),V(I),P(I),E(I),RC,(JC(J),J =JV,JZ),(CV(J),J
     1=JV,JZ)
160   JX =JY+1
      IF (JX.GT.JJ) GO TO 165
      JY =JY+JK
      IF (JY.GT.JJ) JY =JJ
      JP =(JY-JX)+1
      LP =(JK-JP)+1
      JV =JZ+1
      JZ =JR+JY
      LMT(3) =IAC(JP)
      LMT(5) =MAC(LP)
      LMT(7) =IAC(JP)
      WRITE (6,LMT) (JC(J),J =JV,JZ),(CV(J),J =JV,JZ)
      GO TO 160
165   S =S+P(I)
      S1 =S1+E(I)
      S2 =S2+RC
      RS =RS+V(I)
      JR =JR+LB(I)-1
170   CONTINUE
      WRITE (6,11) RS,S,S1,S2
      WRITE (6,7) (NH(I),I =1,65),(NH(I),I =1,65)
175   CONTINUE
C ******** CHECK FOR LAST JOB IN THIS RUN
180   IF (LFIN.NE.1) GO TO 1
      STOP
      END
      SUBROUTINE ALLOC (N,P,V,E,B,S,AS,KB,LY,KRJ,KW,JM,SX,JX,JN,JS,MM)
C ******** THIS SUBROUTINE ASSIGNS EACH NETWORK NODE TO ONE MEMBER OF
C ******** THE CURRENT SET OF MEDIANS
      DIMENSION P(KW),E(KW),V(KW),JN(N),KB(KW),AS(JM),B(JS)
      S =0.0
      KS =(KRJ*N)-N
C ******** INITIALIZE PLANT SIZE AND TRANSPORT COST ARRAYS
      DO 5 I =1,JX
      P(I) =0.0
      V(I) =0.0
```

```
5       CONTINUE
C ******** ALLOCATE SOME NODE TO NEAREST (LEAST COST) MEDIAN
        J =KW*N
        L =(MM*J)-J
        DO 15 I =1,N
        TEMP =SX
        DO 10 J =1,JX
        K =KB(J)
        LL =L+K
        IF (AS(LL).GE.TEMP) GO TO 10
        TEMP =AS(LL)
        KK =J
10      CONTINUE
C ******** UPDATE MEDIAN PLANT SIZE AND TRANSPORT COST VALUES
        L =L+KW
        KS =KS+1
        V(KK) =V(KK)+B(KS)
        SJ =(B(KS)*TEMP)
        P(KK) =P(KK)+SJ
        S =S+SJ
        JN(I) =KK
15      CONTINUE
        IF (LY.EQ.0) GO TO 25
C ******** CALCULATE PLANT COST FUNCTIONS BASED ON PLANT SIZE IF OPTIOND
        DO 20 I =1,JX
        J =KB(I)
        SUM =V(I)
        CALL COST (SUM,Y,J)
        E(I) =Y
        S =S+Y
20      CONTINUE
25      RETURN
        END
        SUBROUTINE FAC (J,K)
C ******** THIS SUBROUTINE CALCULATES THE FACTORIAL OF A NUMBER
        K =1
        IF (J.LT.2) GO TO 10
        DO 5 I =2,J
        K =K*I
5       CONTINUE
10      RETURN
        END
        SUBROUTINE COST (SUM,Y,J)
C ******** OPTIONAL USER SUPPLIED SUBROUTINE FOR CALCULATING PLANT
C ******** COSTS BASED UPON CURRENT SIZE OF THAT PLANT
        RETURN
        END
        SUBROUTINE DATA
C ******** THIS OPTIONAL USER SUPPLIED SUBROUTINE READS IN PLANT COST
C ******** DATA FOR EACH PLANT. THIS INFORMATION MAY BE SUPPLIED TO
C ******** SUBROUTINE COST VIA A SUITABLE COMMON STATEMENT
```

```
RETURN
END
```

APPENDIX VIII

Program Deck set up for all Sample Problems Discussed in Chapter III

```
FIGURE 8(A)  2 MEDIAN SOLUTION FOR FIGURE 4
 10 10   2 1
   2
(10F1.0)
0453132456
4032455453
5305423121
3250243565
1442021345
3524203123
2533130234
4415312012
5526423102
6315534220
(10F2.1)
3010053020405010302 0
```

```
FIGURE 8(B)  3 MEDIAN SOLUTION FOR FIGURE 4
 10 10   2 1
   3
(10F1.0)
0453132456
4032455453
5305423121
3250243565
1442021345
3524203123
2533130234
4415312012
5526423102
6315534220
(10F2.1)
3010053020405010302 0
```

```
FIGURE 8(C)  3 MEDIAN SOLUTION FOR FIGURE 5
 11 11   2 1
   3
(11F2.0)
  10 51516455150505455
1000 51516455150505455
 5 50010114046454549 50
15151000013036353539 40
16161101003137363638 39
45454030310006050569 70
51514636370600111175 76
5050453536 51100107475
50504535360511100074 75
54544939386975747400 01
55555040397076757501 00
(11F2.0)
01012020010101010101 01
```

```
FIGURE 9      2 MEDIAN SOLUTION (LIMITED POTENTIAL SITES) FOR FIGURE 4
 10   6   12 1
    1    4    6    7    9   10
    4
    2
(F1.0,2X,F1.0,1X,F1.0,F1.0,1X,2F1.0)
0453132456
4032455453
5305423121
3250243565
1442021345
3524203123
2533130234
4415312012
5526423102
6315534220
(10F2.1)
3010053020405010302 0

FIGURE 10(A)   4 MEDIAN SOLUTION (PLANT SIZE CONSTRAINTS) FOR FIGURE 6
 10 10    2 11
(10F2.0)
0613101210110716050 5
    4
(10F2.0)
0001061217161725262 6
0100051116151624252 5
0605000611101119202 0
1211060005040513141 4
1716110500030412131 3
1615100403000109101 0
1716110504010010111 1
2524191312091000010 1
2625201413101101000 2
2625201413101101020 0
(10F2.0)
0110041002100110010 1

FIGURE 10(B)   2 AND 3 MEDIAN SOLUTIONS (PLANT SIZE CONSTRAINTS) FOR FIGU
 10   6   12 21
    1    3    5    6    8   10
    8
(10F1.0)
989489
    2    3
(F1.0,1X,F1.0,1X,2F1.0,1X,F1.0,1X,F1.0)
0453132456
4032455453
5305423121
3250243565
```

```
1442021345
3524203123
2533130234
4415312012
5526423102
6315534220
(10F2.1)
30100530204050103020
```

FIGURE 10(C) 2 MEDIAN SOLUTION (MAXIMUM DISTANCE CONSTRAINT) FOR FIGURE

```
 10 10   2 1 1
(10F2.1)
05403070200530408050
   2
(10F1.0)
0453132456
4032455453
5305423121
3250243565
1442021345
3524203123
2533130234
4415312012
5526423102
6315534220
(10F2.1)
30100530204050103020
```

FIGURE 11(A) MINIMUM COMBINED PLANT-TRANSPORTATION SOLUTION FOR FIGURE

```
 10 10   1 1  1 1       1
   1
(10F1.0)
0453132456
4032455453
5305423121
3250243565
1442021345
3524203123
2533130234
4415312012
5526423102
6315534220
(10F2.1)
30100530204050103020
1221212211
```

FIGURE 11(B) 3 MEDIAN SOLUTION (PLANT SIZE CONSTRAINTS) FOR FIGURE 4

```
 10 10   2 11 1 1 121   1
(10F3.1)
 40 60 70100120105 80130 90100
   3
```

```
(10F1.0)
0453132456
4032455453
5305423121
3250243565
1442021345
3524203123
2533130234
4415312012
5526423102
6315534220
(10F2.1)
3010053020405010302 0
1221212211
```

FIGURE 13 FORWARD RECURSIVE DYNAMIC SOLUTION (NODE 9 FORCED) FOR FIGURE

```
 20 20   4 2
   1   3
(20F2.0)
0010030503020506040706070708080707070910
0100020402010405030605060607070606060809
0302000402010405030605060607070606060809
0504040002030607050607080809090808081011
0302020200010405030405060607070606060809
0201010301000304020504050506060505050708
0504040604030001040706070707030302020405
0605050705040100030605060606020201010304
0403030503020403000302030304050304040607
0706060604050706030001020203070607070809
0605050705040605020100010102060506060708
0706060806050706030201000203070607070809
0706060806050706030201020001050606070607
0807070907060706040302030100040605060506
0807070907060302050706070504000201020102
0706060806050302030605060606020002030304
0706060806050201040706070605010200010203
0706060806050201040706070706020301000203
0908081008070403060807080605010302020001
1009091109080504070908090706020403030100
(20F1.0)
31423443122123314232
```

FIGURE 12 FORWARD RECURSIVE DYNAMIC SOLUTION FOR FIGURE 7.

```
 20 20  14 2
   9
   1   3
(20F2.0)
0010030503020506040706070708080707070910
0100020402010405030605060607070606060809
0302000402010405030605060607070606060809
0504040002030607050607080809090808081011
```

```
030202020001040503040506060707060606080 9
020101030100030402050405050606050505070 8
050404060403000104070607070703030202040 5
060505070504010003060506060602020101030 4
040303050302040300030203030405030404060 7
070606060405070603000102020307060707080 9
060505070504060502010001010206050606070 8
070606080605070603020100020307060707080 9
070606080605070603020102000105060607060 7
080707090706070604030203010004060506050 6
080707090706030205070607050400020102010 2
070606080605030203060506060602000203030 4
070606080605020104070607060501020001020 3
070606080605020104070607070602030100020 3
090808100807040306080708060501030202000 1
100909110908050407090809070602040303010 0
(20F1.0)
3142344312212331423 2
```

FIGURE 14 BACKWARD RECURSIVE DYNAMIC SOLUTION FOR FIGURE 7

```
 20 20   3 2                                1
   3   1
(20F2.0)
0010030503020506040706070708080707070910
0100020402010405030605060607070606060809
0302000402010405030605060607070606060809
0504040002030607050607080809090808081011
0302020200010405030405060607070606060809
0201010301000304020504050506060505050708
0504040604030001040706070707030302020405
0605050705040100030605060606020201010304
0403030503020403000302030304050304040607
0706060604050706030001020203070607070809
0605050705040605020100010102060506060708
0706060806050706030201000203070607070809
0706060806050706030201020001050606070607
0807070907060706040302030100040605060506
0807070907060302050706070504000201020102
0706060806050302030605060606020002030304
0706060806050201040706070605010200010203
0706060806050201040706070706020301000203
0908081008070403060807080605010302020001
1009091109080504070908090706020403030100
(20F1.0)
3142344312212331423 2
```