CHAPTER 6

TOPOLOGICAL DESIGN

6.1 Introduction

In this chapter we study the problem of minimizing the cost of a S/F communication network, when topology, routing of the flows and capacity assignment are all considered to be variable.

The exact solution of such a problem for large networks is computationally prohibitive even with the largest computers available today; our intention, therefore, is to develop heuristic algorithms for the determination of good, suboptimal solutions.

In Section 6.2 we give the formulation of the problem.

In Section 6.3 we review the existing techniques for the topological design and introduce the Concave Branch Elimination (CBE) method as an alternative to existing techniques, for the particular case of networks with a concave objective function.

In Section 6.4 the CBE method is applied to the linear and concave cost-cap case, and several examples are presented.

In Section 6.5 we discuss an efficient technique for preserving the 2-connectivity of the solutions.

In Section 6.6 the CBE method is applied to the discrete capacities problem; some additional heuristics are discussed and several examples are presented.

In Section 6.7 we formulate the problem of partitioning large networks into subnets connected by a higher level net, and discuss a decomposition technique for the topological design.

In Section 6.8 we mention possible improvements to the CBE algorithm and extensions of some of the results.

In Section 6.9 we give some concluding remarks and an evaluation of the CBE method as compared to other topological approaches.

6.2 The Topological Problem

Problem (6.1)

given: requirement matrix R

cost-cap functions $D_i = d_i(C_i)$, $\forall$ i

minimize: $D(A, \underset{\sim}{C}) = \sum_{i \varepsilon A} d_i(C_i)$
over $A, \underset{\sim}{C}, \underset{\sim}{f}$

where A is the set of arcs which corresponds to a specific topology*

s.t.: (a) $\underset{\sim}{f}$ is a m.c. flow satisfying the requirement matrix R

(b) $\underset{\sim}{f} \leq \underset{\sim}{C}$

(c) $T = \frac{1}{\gamma} \sum_{i \varepsilon A} f_i \left[\frac{1}{C_i - f_i}\right] \leq T_{max}$

(d) The set A must correspond to a 2-connected topology (see Section 2.3.5).

* Here it is assumed that A is a subset of the set of arcs corresponding to a fully connected network, in which multiple links and self loops are excluded.

6.3 Review of Topological Design Methods for Networks. Introduction of the Concave Branch Elimination (CBE) Method.

As we already mentioned in Chapter 5, the topology is a variable of combinatorial type and the exact solution of the topological problem requires the exploration of a large number of topologies (in the limit all possible combinations); as a consequence, the amount of computation increases exponentially with the number of nodes. We believe that the exact solution is computationally prohibitive already for networks on the order of ten nodes, and that only good heuristic solutions can be found in a reasonable computational time for networks of larger size.

Several examples of heuristic solutions to large topological problems can be found in the literature. In [LIN 65], Lin describes a suboptimal algorithm for the solution of the Traveling Salesman Problem; the algorithm is based on the random generation of several hamiltonian circuits, which are successively improved by means of topological transformations, involving only three arcs at a time, until a local minimum is obtained. The minimum of the local minima is the heuristic solution to the problem. Lin applied the algorithm to a variety of examples, for which the exact solution was known, and found the exact solution for all of them!

A similar approach is used by Frank et al. for the determination of the minimum cost topology of a pipeline network connecting gas fields to separation plants in the Gulf of Mexico [FRAN 69]. The network is assumed to have a tree structure, and the algorithm consists of the random generation of several different trees, whose cost is successively reduced by topological transformations, in which arcs are added and deleted one at a time. A very efficient dynamic programming algorithm finds the optimal capacities which yield the minimum cost for the new topology obtained after each transformation.

Frank et al. applied the same dynamic programming approach to the design of centralized computer networks [FRAN 71A].

A heuristic approach to the design of minimum cost survivable networks is described by Steiglitz et al. in [STEI 69]. The method consists of a starting routine, which generates random feasible topologies, and of an optimizing routine, which improves the cost of the starting topology by means of local transformations, called X-changes. An X-change corresponds to the deletion of two arcs, say (i, m) and (j, ℓ), and the introduction of two new arcs (i, ℓ) and (j, m). The practicality of the algorithm is based on a very efficient technique for testing the feasibility of the new topology after each X-change [KLET 69].

A heuristic method for the design of minimum cost, 2-connected computer networks is proposed by Frank et al. in [FRAN 70]. The approach is similar to that described in [STEI 69], and applies the same techniques for random generation of topologies and topological transformations. In addition, a very efficient heuristic routing algorithm is developed.

Common features of the above heuristics are: random generation of several starting topologies (which ensures wide sampling of the solution space); availability of fast and efficient techniques for the evaluation of each topology.

Less sophisticated heuristics do not apply the randomization of the starting topology: to such a category belong several techniques recently proposed for the design of minimum cost centralized computer networks [MART 67, ESAU 66, WHIT 72B]. Typically, the suboptimal configuration is obtained after the repeated application of simple topological operations (e.g., insertion, deletion or replacement of a branch, etc.). An interesting evaluation of some of the methods, as compared to the optimal solution, is presented by Chandy and Russel in [CHAN 72B].

We might classify all the above methods as <u>branch-exchange</u>, or <u>branch-insertion methods</u>: branches are systematically exchanged or inserted, following some well defined criteria.

In the special case of a multicommodity flow network, in which the objective to minimize is a <u>concave</u> function of the flows, another heuristic method can be proposed as an alternative (or as a complement) to the branch-exchange methods. The method is based on the property that the flow patterns, which are local minima of the CFA problem, typically concentrate the flows on some links, and leave some other links with zero flow (see Chapter 5): the initial topological configuration is therefore automatically reduced in the process of finding local minima. We will refer to such topological reduction, induced by concavity, as <u>Concave Branch Elminiation</u> (CBE).

The CBE method consists of two routines: the <u>random starting routine</u>, which generates several random starting topologies and, for each topology, several random starting flow configurations; the <u>optimizing routine</u>, which improves a given starting topology with progressive "concave elimination" of expensive arcs, until a local minimum is reached.

The random starting routine must generate initial topologies which are likely to contain the optimal topology as a subgraph, and, at the same time, that can be conveniently processed by the optimizing routine. In the choice of such initial topologies, the human interaction can be very useful; in fact, in many examples introduced later in the chapter, the initial topologies were generated by hand. A method for the automatic generation of initial topologies is outlined in Section 6.8.

The idea of using concave branch elimination for the topological design of networks, with concave link costs and multicommodity flow requirement, is not new. Yaged in [YAGE 71] applies such an approach to the determination of minimum cost topologies for a large telephone network, where the total cost is the sum of the concave link costs; several minimum cost topologies (corresponding to different link costs) are obtained, starting from a common, highly connected, planar topology (which is implicitly assumed to contain all minimum cost topologies as subgraphs).

The CBE method here proposed is a generalization of Yaged's technique: it applies to nonseparable objective functions and guarantees a wider sampling of the solution space, through the random generation of initial topologies and initial flow assignments.

The CBE method is, therefore, applicable to the topological design of S/F networks, as we showed in Chapter 5 that the CFA problem leads to the minimization of a nonseparable concave objective function.

6.4 Concave Cost-Cap Case Without the 2-Connectivity Constraint

In the present section we assume that the cost-cap functions are linear or concave; we also relax the 2-connectivity constraint.

With the above assumptions, Problem (6.1) can be regarded as a capacity and flow assignment problem (see Section 5.4) in which the initial topology is fully connected: the CBE method, therefore, reduces to the FD method. In some applications (typically, the applications with moderate concavity of the cost-cap functions) the fully connected starting net produces very satisfactory results. In some other applications (pronounced concavity of the cost-cap functions and, in the limit, presence of start up costs), a fully connected start leads typically to locals, which are very far from optimum. For the latter applications, the CBE method is greatly improved by selecting initial topologies, which are likely to contain the optimal topology and which exclude, on the other hand, obviously bad links. For networks on the order of 20 to 50 nodes, a large sample of good initial topologies can be generated by hand. For larger networks, the generation can be done with the aid of the computer (see Section 6.8).

The CBE method has been applied to the design of topologies connecting 26 ARPA sites (see Figure 6.4.1). Several concave channel costs have been considered (α fitted; uniform $\alpha = 1.0, 0.8, 0.6, 0.5, 0.1$),* The traffic requirement r was assumed uniform (in some cases, $r = 1.0$ [kbits/sec]; in some others $r = 0.74$ [kbits/sec]). A maximum delay $T_{max} = .200$ was required. For each value of α several initial

*See Section 5.8 for the analytical expression of the channel costs.

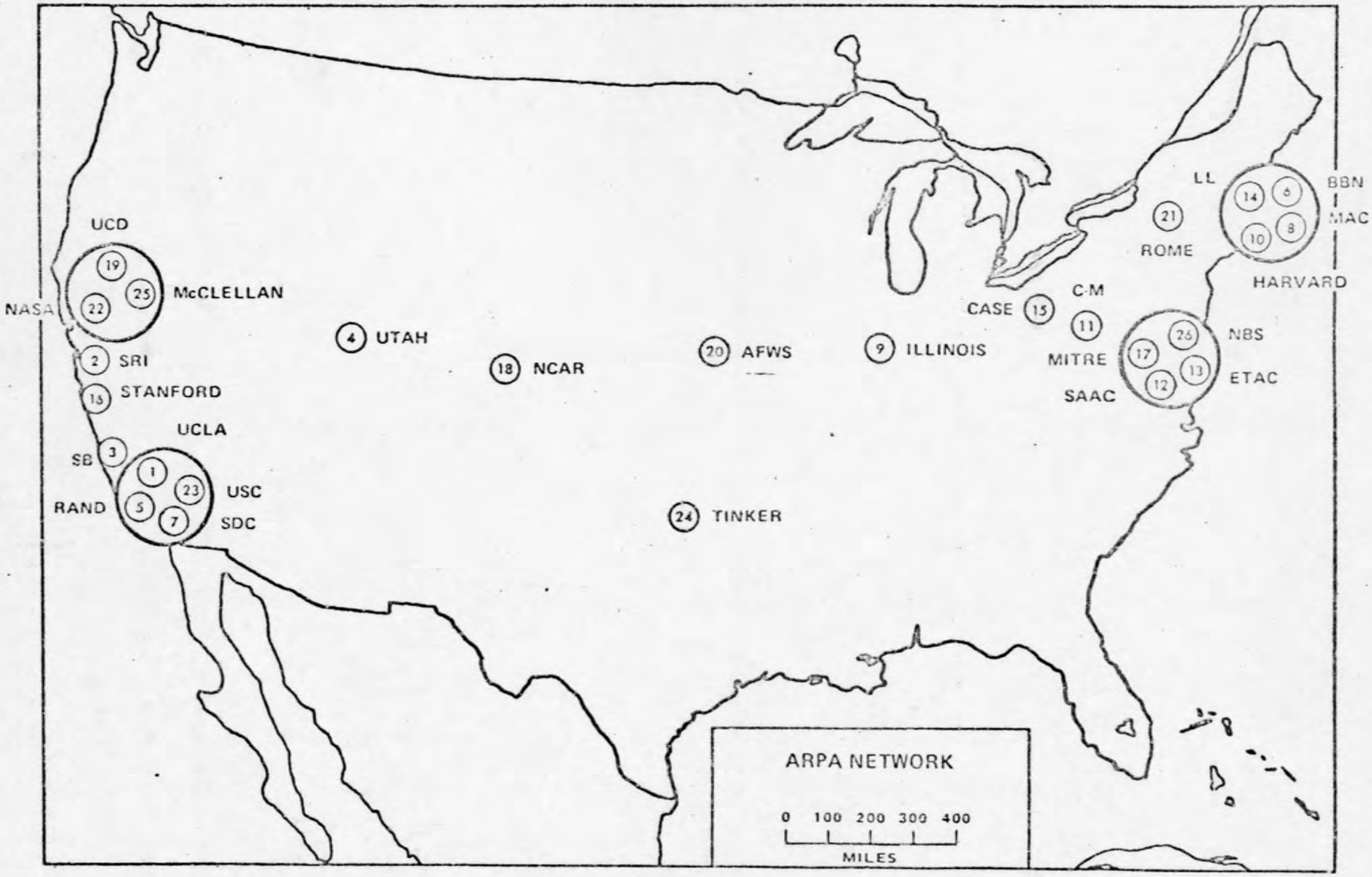


Figure 6.4.1. 26 ARPA Sites

topologies were considered; for notational convenience we classify them in the following way:

- fully connected (325 arcs)
- highly connected (above 40 arcs)
- medium connected (30-40 arcs)
- low connected (26-29 arcs)
- trees (25 arcs)

Notice that the arcs are nondirected (i.e., each arc corresponds to two directed arcs, with opposite directions, and is physically implemented with a full duplex channel).

A summary of the results is shown in Tables 6.4.2a and 6.4.2b. The results are subdivided into six classes, each class corresponding to a different value of α. For each CBE application we give:

- degree of connection: it defines the type of initial topology considered (e.g., fully connected, minimum spanning tree,* shortest hamiltonian, etc.).
- NA_0: number of arcs of initial topology.
- NLOC: number of local minima explored.
- D_1, NA_1: cost (\$/month) and number of arcs of the best local minimum.
- D_2, NA_2: cost (\$/month) and number of arcs of the second best local minimum.

An accurate analysis of the results permits us to establish interesting properties of the suboptimal solutions. Some of these

*The minimum spanning tree was computed with link lengths proportional to the geographical distances.

TABLE 6.4.2a

RESULTS OF THE CBE METHOD, WITH CONCAVE COST CURVES
AND NO 2-CONNECTIVITY CONSTRAINT

α fitted, r = 1.0 [kbits/sec x node pair]

Degree of connection	NA_0	NLOC	D_1	NA_1	D_2	NA_2
fully conn.	325	30	82,583	55	82,961	52
highly conn.	53	30	81,202	40	81,979	39
highly conn.	52	30	81,988	40	82,077	42
med. conn.	35	30	82,606	30	82,743	30
med. conn.	33	30	84,719	30	84,721	30
sh. hamilt.	26	30	94,977	26	94,977	26
min. sp. tree	25	1	91,775	25	-	
tree 2	25	1	95,456	25	-	

α = 1.0, r = 0.74 [kbits/sec x node pair]

Degree of connection	NA_0	NLOC	D_1	NA_1	D_2	NA_2
fully conn.	325	30	62,459	52	64,079	52
highly conn.	40	30	62,029	39	62,062	39

α = 0.8, r = 0.74 [kbits/sec x node pair]

Degree of connection	NA_0	NLOC	D_1	NA_1	D_2	NA_2
fully conn.	325	30	65,439	34	65,443	39
highly conn.	40	30	62,751	34	62,922	35
min. sp. tree	25	1	65,073	25	-	

TABLE 6.4.2b

RESULTS OF THE CBE METHOD, WITH CONCAVE COST CURVES AND NO 2-CONNECTIVITY CONSTRAINT

$\alpha = 0.6$, $r = 1.0$ [kbits/sec x node pair]

Degree of connection	NA_0	NLOC	D_1	NA_1	D_2	NA_2
fully conn.	325	50	71,233	27	72,325	27
highly conn.	40	30	65,834	27	70,291	31
med. conn.	33	30	68,780	28	68,821	27
min. sp. tree	25	1	66,207	25	-	
tree 2	25	1	65,158	25	-	

$\alpha = 0.5$, $r = 1.0$ [kbits/sec x node pair]

Degree of connection	NA_0	NLOC	D_1	NA_1	D_2	NA_2
highly conn.	53	50	64,421	26	65,012	27
highly conn.	40	30	60,712	26	63,117	28
med. conn.	32	50	63,582	27	67,088	28
min. sp. tree	25	1	60,505	25	-	
tree 2	25	1	59,719	25	-	

$\alpha = 0.1$, $r = 0.74$ [kbits/sec x node pair]

Degree of connection	NA_0	NLOC	D_1	NA_1	D_2	NA_2
fully conn.	325	30	64,100	25	65,575	25
highly conn.	40	30	48,040	25	49,821	25
med. conn.	33	30	47,711	25	49,476	25
min. sp. tree	25	1	42,601	25	-	
tree 2	25	1	43,003	25	-	

properties were already pointed out in Chapter 5. In addition, some new properties, which relate the topological characteristics of the solutions to the input parameters, were observed. The properties can be summarized as follows:

(a) When α decreases (i.e., the economy of scale increases), the number of arcs of the suboptimal solution decreases. In fact, for $\alpha = 1.0$ and α fitted, good topologies have a number of arcs varying from 30 to 60; topologies with higher or lower numbers of arcs exhibit poor performance (as in the case of the two trees or the shortest hamiltonian circuit, for α fitted). For $\alpha = 0.8$, the optimal number of arcs seems to be between 40 and 30. For $\alpha \leq 0.6$, all of the best solutions that we found had a tree structure.

(b) When α decreases, the range of variation of NA (final number of arcs) for the good suboptimal topologies becomes smaller. We already mentioned that, for $\alpha = 1$, many good solutions have NA between 60 and 30. For $\alpha = 0.1$, the good topologies all have a tree structure (NA = 25).

(c) When α decreases, the range of variation of the costs within each run becomes larger (we already observed this property in Chapter 5). Considering the distribution of the costs of the local minima obtained from a given initial topology, we noticed that, for α fitted and $\alpha = 1.0$, more than 30% of the costs were within 2-3% of the best cost for that run. For $\alpha = 0.6$ and $\alpha = 0.5$, only 10-20% of the costs were within 10% of the best. The spread of the distribution of the costs was increasing with the degree of connection of the initial topology, and was maximum for the fully connected topology.

(d) When α decreases, the range of variation of the costs obtained from different runs (i.e., using different initial topologies) becomes larger. For α fitted, all the initial topologies with $NA_0 \geq 35$ produced solutions in a 2% range. For $\alpha = 0.6$ and 0.5, the fully connected start produced poor results; other initial topologies, with NA_0 between 50 and 25, gave solutions in a 5-10% range. For $\alpha = 0.1$, initial topologies with NA_0 between 30 and 40 produce solutions with costs which are 15% higher than the cost of the minimum spanning tree (which is the exact solution for $\alpha \to 0$).

Properties (a) and (b) can be attributed to the fact that small α corresponds to strong economy of scale and favors topologies with large capacities concentrated in a few arcs. Properties (c) and (d) are a consequence of the fact (already mentioned in Chapter 5) that, when α decreases, the number of local minima increases and the costs of such local minima are widely diversified (see Figure 5.8.11).

This fact also explains the performance of different initial topologies for different values of α. Highly connected (h.c.) topologies are more likely to contain the optimal topology, as a local minimum, than low connected (l.c.) topologies; on the other hand, for the same value of α, h.c. topologies contain a much larger number of local minima (we conjecture that such a number increases exponentially with NA_0). For $\alpha = 0.8\text{-}1.0$, h.c. topologies offer a good probability of obtaining, if not the optimal solution, at least very good solutions, because the number of local minima is relatively small, and the values of the minima are close; l.c. topologies, on the other hand, restrict arbitrarily the set of solutions to a region which might be far from optimum. For small α , the number of local minima is so large (for $\alpha \to 0$, all extreme flows are stationary flows), and their values are so diversified that h.c. topologies lead usually to bad locals; carefully chosen l.c. topologies can perform better, as they eliminate many bad locals.

The above considerations indicate that the CBE method is very useful for applications with $\alpha = 0.8\text{-}1.0$: the choice of the initial topology is not very critical, and the exploration of a few local minima gives, in general, already good solutions. In the range of $\alpha = 0.5\text{-}0.8$, the CBE method can still be applied, but a careful choice of the initial topology (see Section 6.8) and the exploration of a large number of locals are advisable. For $\alpha < 0.5$, the CBE method seems to be of little use. This does not mean that we cannot find good solutions for small α ; in fact, as we showed, the good solutions have a tree structure and therefore the topological problem corresponds to the problem of finding the minimum cost tree that satisfies $T \leq T_{max}$. Notice that, for a tree, the routing assignment is unique, therefore, given the tree, we can compute immediately $\underset{\sim}{f}$, $\underset{\sim}{C}$ and $D(\underset{\sim}{C})$. The efficiency with which we can evaluate $D(\underset{\sim}{C})$ for new topologies suggests the use of a <u>branch X-change</u> method for the search of the minimum cost tree [STEI 69, FRAN 70]. As an alternative approach, one could determine several local minima with the CBE method, and then improve them with branch X-change techniques. The simple inspection of two solutions obtained from a highly connected topology, with $\alpha = 0.1$ (see Figures 6.4.3 and 6.4.4) suggests that a few branch X-changes could considerably reduce the cost.

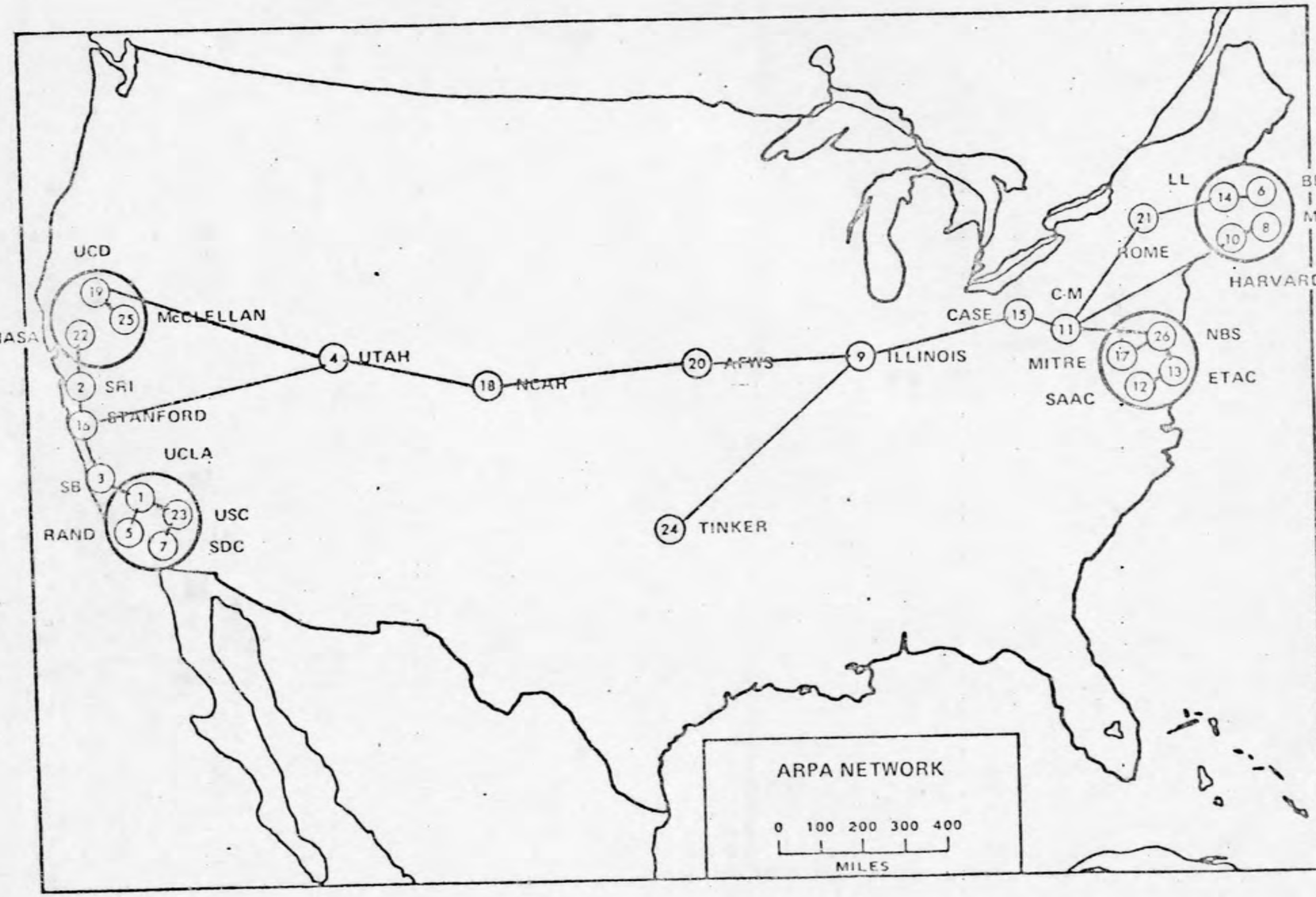


Figure 6.4.3. Local Solution for $\alpha = 0.1$; D = 47,711

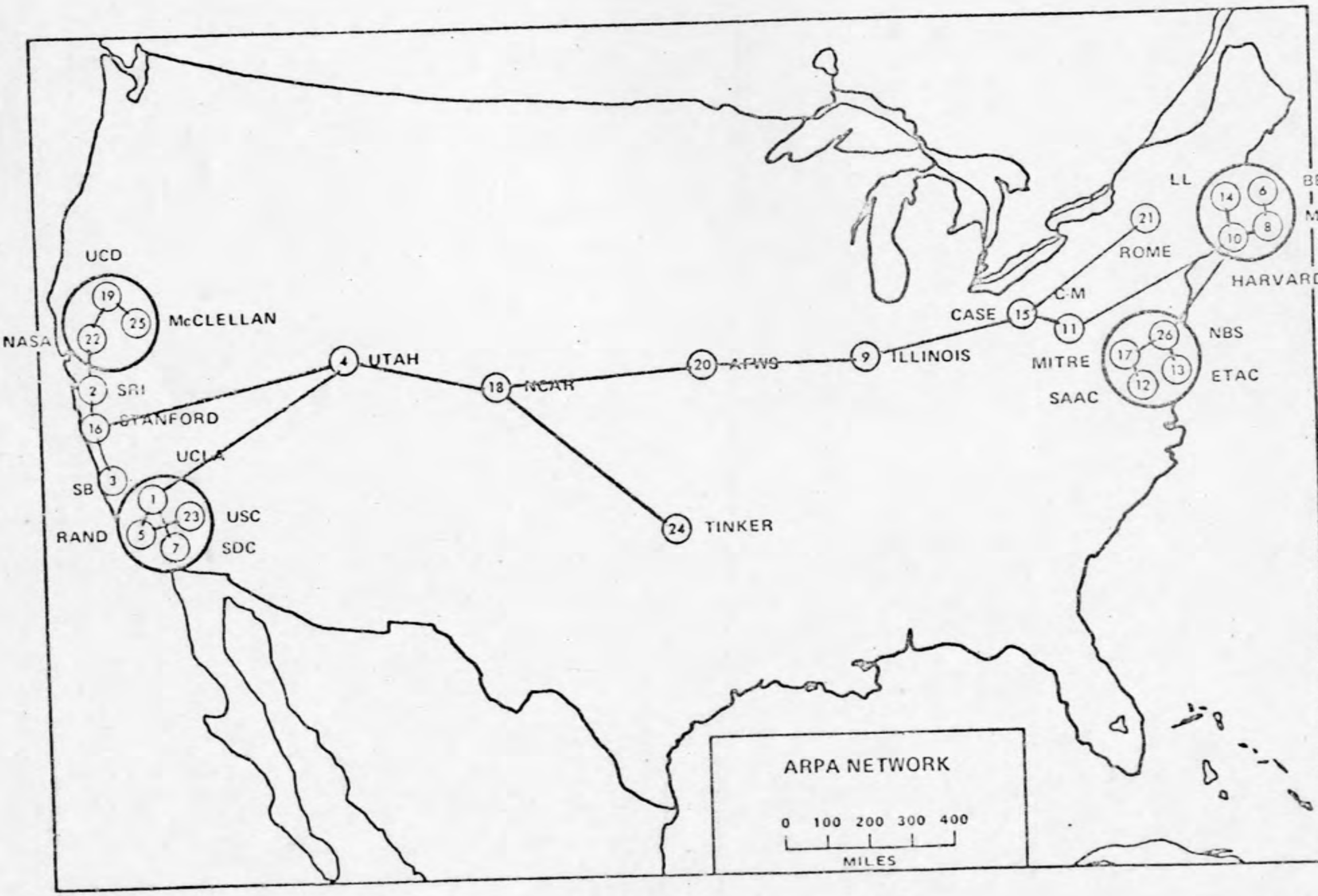


Figure 6.4.4. Local Solution for $\alpha = 0.1$; D = 48,040

The above considerations were introduced here in relation to the topological design of a S/F communication network; in fact, they are very general and can be applied to the topological design of any network such that:

- a given m.c. flow requirement must be satisfied.
- the total cost $D(\underset{\sim}{f})$ (to be minimized) is a continuous,* concave function of the m.c. flow $\underset{\sim}{f}$; also, $D(f_i)$ is increasing with respect to f_i, $\forall i$.
- the m.c. flow problem is unconstrained (see Chapter 4); this implies that no additional constraints (i.e., constraints on the value of admissible flow for each arc, topological constraints, etc.) are imposed.

Clearly the topological problem can be considered as a m.c. flow problem on a fully connected topology: each m.c. flow $\underset{\sim}{f}$ induces a topology in which only the arcs that carry nonzero flow are present.

We know that the optimal solutions are extremal flows; in particular, if $D(\underset{\sim}{f})$ is linear, i.e.:

$$D(\underset{\sim}{f}) = \sum_i C_i f_i$$

the optimal flow is the shortest route flow corresponding to the metric $\{C_i\}$, and the optimal topology is obtained from the fully connected one

*The FD method was defined for differentiable objective functions; however, in the concave case, it can be extended to piecewise differentiable functions, and in general, to any continuous function (see Section 3.4). Furthermore, the method can be extended to the noncontinuous case of set up costs for the links, if the discontinuities are replaced by proper continuous (and concave) approximations [YAGE 71].

by eliminated arcs with the triangle inequality.* If, on the other hand, there are only set up costs, i.e.:

$$D(\underset{\sim}{f}) = \sum_{i \in I} C_i$$

$$\text{where} \quad I \triangleq \{i \mid f_i > 0\}$$

the optimal topology is the minimum spanning tree corresponding to the metric $\{C_i\}$. Between the linear case (no economy of scale) and the set up cost case (maximum economy of scale) there is a large variety of cases with different degrees of economy of scale. We can extend our results to such cases as follows:

- in the case of moderate economy of scale, the good topologies tend to be highly connected and the CBE method is a convenient tool for their determination.

- in the case of strong economy of scale, the good topologies are trees, and branch X-change techniques (eventually combined with the CBE method) seem to be the convenient approach.

- in the intermediate cases we believe that the CBE method, combined with a heuristic search of the starting topology and with branch insertion techniques (see Section 6.8), is the proper approach.

Until now we have discussed the dependence of the results upon the characteristics of the cost function $D(\underset{\sim}{f})$. We expect that the

*Typically, such a topology is highly connected; it is fully connected if the metric $\{C_i\}$ satisfies the triangle inequality [HU 69].

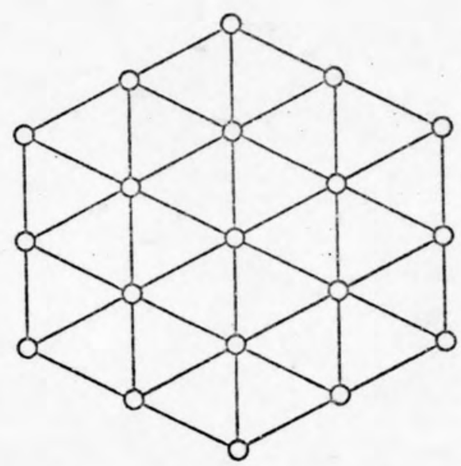
results should depend also on the degree of "balance" of the traffic requirement (see Section 4.8); on the total thruput; on the number of nodes NN; on their geographical distribution, etc. A rigorous investigation of such dependence would require the study of a large number of different cases. Here we limit ourselves to some simple considerations.

We can intuitively expect that a highly unbalanced requirement would drive the optimal topology to a tree: such expectation is motivated by the fact that, for multiterminal, centralized networks (i.e.: $r_{ij} = 0 \ \forall i, j$, except for $i = C$, or $j = C$, where C is the "central" node), the optimal topology is a tree [ZANG 68].

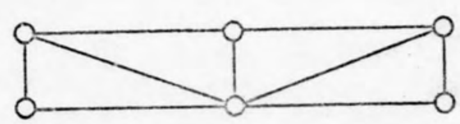
Also, we would expect that a uniform geographical distribution* of the nodes tends to level off the differences in cost between the various topologies obtained with the CBE method. Consider, as an example, the two very different node distributions shown in Figure 6.4.5 (a) and (b). Suppose that we want to determine the optimal topologies for both distributions, using the CBE method and assuming concave link costs, with $\alpha \to 0$ (see Equation (5.31)). Let us also assume that, as initial topologies for the CBE method, the two very reasonable planar topologies of Figure 6.4.5 (a) and (b) are used. The well known exact solutions are the minimal spanning trees. Such minimal trees, as well as many other trees, can be generated from the initial topologies by the CBE method. Notice, however, that any spanning tree for topology (a) is minimal; on the other hand, notice that, for topology (b), many

---

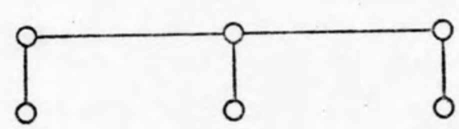
*Here we assume that the link costs are related to the geographical lengths. More generally, the regularity of the cost matrix should be considered.



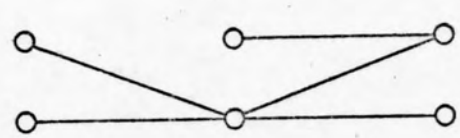
(a) UNIFORM GEOGRAPHICAL DISTRIBUTION OF THE NODES



(b) IRREGULAR NODE LOCATION



(c) MINIMUM SPANNING TREE FOR TOPOLOGY (b)



(d) ANOTHER SPANNING TREE FOR TOPOLOGY (b)

Figure 6.4.5. Impact of Geographical Node Location on the Topological Design.

spanning trees (see, for example, the tree in Figure 6.4.5 (a)) have a much higher cost than the minimal spanning tree (c). We conclude that the application of the CBE method, with $\alpha \to 0$, is successful for topology (a) and disastrous for (b). This example is clearly an extreme case, but it motivates our expectation that the CBE method can handle uniform node distributions better. These considerations might also help to understand why Yaged, having a fairly uniform node distribution (see [YAGE 71]), obtained much better results* for very small α, than we did with the 26 ARPA sites, which are more irregularly distributed.

Finally, as an example of the relation between optimal topological structure and thruput, we can consider the case of link costs represented by the contribution of a set up cost plus a cost which is linearly increasing with the capacity. For small thruput, the set up cost is predominant, and the optimal solutions are trees. For large thruput, the variable cost is predominant, and the optimal solutions are highly connected topologies.

6.5 The 2-Connectivity Constraint

A very important requirement for a communications network is the survivability to failures: the network must remain operational (i.e., nodes must be able to communicate with each other) even after the failure of n elements (nodes or arcs). It can be shown [FRIS 67] that a network, in order to survive to $n - 1$ arbitrary failures, must

*Yaged claims, in [YAGE 71], that, starting from a highly connected planar topology and applying concave branch elimination techniques, he could generate, in the case $\alpha \to 0$, the exact solution, i.e., the minimum spanning tree.

provide n independent paths (i.e., with no common intermediate nodes or arcs) between each pair of nodes; the network is then referred to as n-connected.

Usually, computer networks are considered to be sufficiently reliable if they survive one simple failure at a time [ROBE 70]; for that reason, only 2-connectivity is required in the formulation of Problem (6.1).* However, many of the considerations that follow can be applied to the general n-connectivity case.

Very efficient techniques were recently developed for the analysis of the connectivity in communication networks [FRIS 67, KLET 69]. As for the design of minimum cost, n-connected networks, exact solutions are available only in the very special case of link costs which are merely set up costs and which are identical for all the links. A heuristic approach to the solution of the case of set up costs which are different from link to link has been discussed by Steiglitz et al. [STEI 69]: the approach utilizes the branch X-change technique, in which only X-changes that preserve n-connectivity and reduce the cost are accepted. The case of communication networks with link costs which depend on the capacity was considered by Frank et al. in [FRAN 70], in relation to the design of a 2-connected computer network; the method applies the branch X-change technique, in which a branch X-change is accepted only if it preserves 2-connectivity and improves the

---

*A more complete definition of survivability should include, in addition to 2-connectivity, the maximum tolerable degradation in performance (e.g., thruput or delay) when an element fails. However, the test of such degradation at each step of the design would represent too severe an overhead; therefore, it is more convenient to verify it a posteriori.

performance. The method is very similar to that of Steiglitz et al. in [STEI 69]; notice, however, that the evaluation of the cost after each X-change, while it is trivial for set up costs, is extremely difficult for costs which vary with the capacities, as it involves an optimal assignment of routing and capacities.

If the CBE method is applied to the design of minimum cost, 2-connected networks, the obvious way to obtain 2-connected solutions is to start from a 2-connected topology and to test 2-connectivity after each iteration;* the algorithm terminates when the test fails or when no improvement is obtained. In both cases, we retain the result of the iteration before the last.

The presence of the 2-connectivity requirement increases the cost of the optimal solution (if the optimal, unconstrained solution is not 2-connected). This effect can be seen in Tables 6.5.1 (a), (b) and (c), where the results with and without 2-connectivity test, obtained from various initial topologies, are compared.

Notice that, for α fitted, the degradation is not too severe (the constrained minimum for each run is within 2% of the unconstrained one); for $\alpha = 0.5$, on the other hand, the degradation is dramatic (the difference between constrained and unconstrained minimum is on the order of 20-30%). This is no surprise, as the good topologies for

---

*The 2-connectivity test can be implemented with a labeling algorithm, which requires from $(NN)^2$ to $(NN)^3$ elementary operations, depending on the degree of connection of the topology under consideration. The overhead due to the introduction of such a test is not too severe, if we consider that one shortest route computation alone requires $(NN)^3$ operations.

TABLE 6.5.1a

COMPARISON OF THE CBE RESULTS
WITH AND WITHOUT 2-CONNECTIVITY TEST

α fitted, r = 1.0 [kb/sec x node pair]

Example 1: fully connected initial topology (NA_0 = 325),

number of local minima NLOC = 30

No 2-connectivity test

D[K$]	number of solutions
82 - 84	2
84 - 86	4
86 - 88	6
88 - 90	7
90 - 92	5
> 92	6

2-connectivity test present

D[K$]	number of solutions
82 - 84	1
84 - 86	3
86 - 88	6
88 - 90	6
90 - 92	6
> 92	8

Example 2: highly connected initial topology (NA_0 = 53),

number of local minima NLOC = 50

No 2-connectivity test

D[K$]	number of solutions
82 - 84	15
84 - 86	20
86 - 88	5
88 - 90	5
90 - 92	5
> 92	0

2-connectivity test present

D[K$]	number of solutions
82 - 84	0
84 - 86	11
86 - 88	4
88 - 90	3
90 - 92	2
> 92	30

TABLE 6.5.1b

COMPARISON OF THE CBE RESULTS
WITH AND WITHOUT 2-CONNECTIVITY TEST

$\underline{\alpha \text{ fitted}}$, r = 1.0 [kb/sec/node x pair]

Example 3: medium connected initial topology (NA_0 = 32),

number of local minima · NLOC = 50

No 2-connectivity test

D[K$]	number of solutions
88 - 90	50
90 - 92	0
92 - 94	0
94 - 96	0
96 - 98	0
> 98	0

2-connectivity test present

D[K$]	number of solutions
88 - 90	2
90 - 92	12
92 - 94	6
94 - 96	16
96 - 98	10
> 98	4

TABLE 6.5.1c

COMPARISON OF THE CBE RESULTS
WITH AND WITHOUT 2-CONNECTIVITY TEST

$\underline{\alpha = 0.5}$, $r = 1.0$ [kb/sec x node pair]

Example 1: highly connected initial topology ($NA_0 = 53$),

number of local minima NLOC = 50

No 2-connectivity test

D[K$]	number of solutions
64 - 68	7
68 - 72	13
72 - 76	16
76 - 80	12
80 - 84	2
> 84	0

2-connectivity test present

D[K$]	number of solutions
64 - 68	0
68 - 72	0
72 - 76	0
76 - 80	0
80 - 84	1
> 84	49

Example 2: medium connected initial topology ($NA_0 = 32$),

number of local minima NLOC = 50

No 2-connectivity test

D[K$]	number of solutions
62 - 66	1
66 - 70	30
70 - 74	19
74 - 78	0
78 - 82	0
> 82	0

2-connectivity test present

D[K$]	number of solutions
62 - 66	0
66 - 70	0
70 - 74	16
74 - 78	34
78 - 82	0
> 82	0

α fitted tend to be highly connected, thus including a fairly large number of 2-connected configuration; for $\alpha = 0.5$, on the other hand, the good topologies tend to have a tree structure, and therefore 2-connectivity and low cost are contradictory design criteria. In general, for small α , low cost 2-connected configurations are difficult to locate with the CBE method.

The above results suggest that, in general, the CBE method plus 2-connectivity test is a good approach for the applications with moderate economy of scale ($\alpha = 0.8 - 1.0$ in our case). For applications with medium economy of scale ($\alpha = 0.6$), the CBE method should be combined with a branch insertion routine (see Section 6.8), which preserves 2-connectivity by introducing a proper set of arcs whenever the topology, during the CBE optimization, becomes monoconnected. For applications with very strong economy of scale (see, as a limiting case, the problem considered by Steiglitz et al. in [STEI 69]), branch X-change seems to be the only reasonable approach.

In order to be reliable, a 2-connected network must also be able to contain within acceptable limits the degradation in performance following a failure (see footnote on page 196). For instance, if the links have no set up costs, any monoconnected solution can be made 2-connected by introducing appropriate links with infinitesimal capacity, without virtually increasing the cost: such a solution would obviously not meet the reliability requirements. The solutions obtained with the CBE method must be, therefore, a posteriori verified, to make sure that their reliability is acceptable. This additional reliability test, however, is not too critical for CBE solutions, for the following reasons:

- the final flow configuration is an extremal flow (see Section 4.3), therefore the capacity in each arc is $\geq \min \{r_{ij} \mid r_{ij} > 0\}$; this excludes pathological cases with infinitesimal capacities assigned to some arcs.

- the CBE method tends to eliminate arcs with small capacity, as their marginal cost is very high. Therefore, small capacities are not likely to be found in the final configuration.

- typically, the CBE method generates a large set of good solutions: it is very likely that some of them will meet the reliability requirement.

6.6 Discrete Cost-Cap Case

If the discrete channel costs can be reasonably approximated by continuous, concave costs $D_i(C_i)$, such that $D_i(0) = 0$ (i.e., no set up cost), then the topology (or, better, several good topologies) can be designed with the CBE method using the continuous approximation; from each of the continuous solutions a discrete capacity assignment can be derived with the techniques described in Chapter 5.

This continuous-discrete approach was applied to the topological design of a network connecting the 26 ARPA sites shown in Figure 6.4.1. The discrete costs are given in Table 5.8.1; as a continuous, concave approximation to such costs, we used the α fitted, power law curves described in Section 5.8. A uniform traffic $r = 1.0$ [kb/sec x node pair] is required. The delay $T_{max} = 0.200$ [sec] is prescribed, as maximum admissible average delay.

In order to obtain 2-connected solutions, the CBE method with 2-connectivity test was applied. The degradation produced by the test was not too severe for α fitted (as shown in Section 6.5).

For the assignment of discrete capacities, the following simplified version of DisCap (see Section 5.7) was used:

- let $\underset{\sim}{f}$ be the flow of the continuous solution
- let $\underset{\sim}{C}$ be the minimum cost discrete capacities assignment such that $C_i \geq f_i$, $\forall$ i (minimum fit assignment)
- with assignment $\underset{\sim}{C}$, let ρ be the maximum traffic level such that $T \leq T_{max}$.

$\underset{\sim}{C}$ is the discrete capacities assignment and ρ is the traffic level associated with it; if $\rho \geq 1$, the solution is feasible. Experimentally, we found that with T_{max} = .200 sec most of the minimum fit assignments satisfy $\rho > 1$.

In order to ensure a wide sampling of the solution space, many different initial topologies have been tried. About 30 different entries were generated by hand by different people: most of those entries had a number of arcs between 30 and 35. In addition to such entries, the fully connected and two highly connected initial topologies were used. For each entry 30 local minima were determined.

The results are shown in Table 6.6.1. For each entry (identified by the degree of connection or by the initials of the designer) the following values are given:

$DCONT_{min}$: cost of the best continuous solution (the cost is evaluated on the continuous, α fitted, curves)

TABLE 6.6.1

COST DDISC AND TRAFFIC LOAD ρ FOR VARIOUS 26 NODE ARPA TOPOLOGIES

NAME	DDISC	ρ	$DCONT_{min}$	$DCONT_{disc}$	NA_0	NA
Fully conn.	89,580	1.05	82,583	86,164	325	61
JAW	94,288	1.00	88,792	88,799	29	29
JON	94,314	1.00	84,881	86,154	33	33
MAX3	94,357	1.03	88,877	88,892	29	29
High.conn.1	95,191	1.01	82,149	82,466	41	39
KLE	95,621	1.04	89,485	90,529	31	31
TOB1	96,017	1.03	89,134	89,134	29	29
CAK	97,100	1.00	88,997	88,997	39	33
High.conn.2	97,215	1.02	82,765	82,991	41	38
MAG	97,240	1.08	83,006	83,006	34	34
BAN	98,055	1.02	90,331	91,427	33	33
DGC2	98,478	1.03	87,320	87,930	33	33
DPD	98,554	1.10	86,616	86,616	35	35
JOP2	99,783	1.03	88,405	88,764	31	31
TOM	99,992	1.06	87,513	87,574	38	35
MAG2	100,207	1.00	86,748	87,019	30	30
VCG	100,815	1.03	86,302	92,354	35	35
HHO	101,075	1.00	86,814	86,814	33	33
KRI1	101,703	1.06	84,078	84,874	31	31
BAN2	103,164	1.01	87,181	91,427	34	34
MAX2	103,371	1.00	87,306	87,728	33	32
ARI1	105,652	1.06	84,860	87,889	34	32
MAX	106,840	1.00	89,270	97,396	35	35
LAM	108,540	1.10	90,134	90,134	51	42
DUF	108,644	1.00	87,908	97,841	42	38
JOP	112,669	1.00	88,941	91,602	34	34
GAS	118,579	1.00	91,956	94,558	30	30
KLE2	122,309	1.01	90,470	90,481	29	29
DGCIP	133,251	1.03	89,591	89,591	29	29
DGC	141,396	1.10	90,247	90,431	28	28
NAM	150,968	1.10	92,991	92,991	31	30

$DCONT_{disc}$: cost of the continuous solution which generated the best discrete solution.

DDISC : cost of the best feasible (i.e., $\rho \geq 1$) discrete solution

ρ : relative traffic level associated with the best discrete solution.

NA_0 : number of arcs of the initial topology

NA : number of arcs of the final topology.

The computation time for each entry was between 30 and 60 seconds on an IBM 360/91.

The results for the various entries are presented in Table 6.6.1 in order of increasing DDISC. The best solution (DDISC = 89,580, ρ = 1.05) was obtained from a fully connected initial topology; it has 61 arcs (see Figure 6.6.2) and uses 9.6 and 19.2 kb capacities for the medium and long lines, and 50 kb capacities for the short lines (see the distribution of the capacities with respect to link lengths in Table 6.6.2). The second best solution (DDISC = 94,288, ρ = 1.00) was obtained, on the other hand, from a low connected topology; it has 29 arcs and uses prevalently 50 and 100 kb capacities on medium and long lines, and 230 kb on very short lines (see Figure 6.6.3). The other solutions have a number of arcs variable from 28 to 42 and a cost DDISC variable from 94,314 to 150,968.

From the analysis of the results, it can be noticed that most of the entries produce continuous solutions with cost $DCONT_{min}$ between 82,000 and 90,000; in particular, highly and medium connected initial

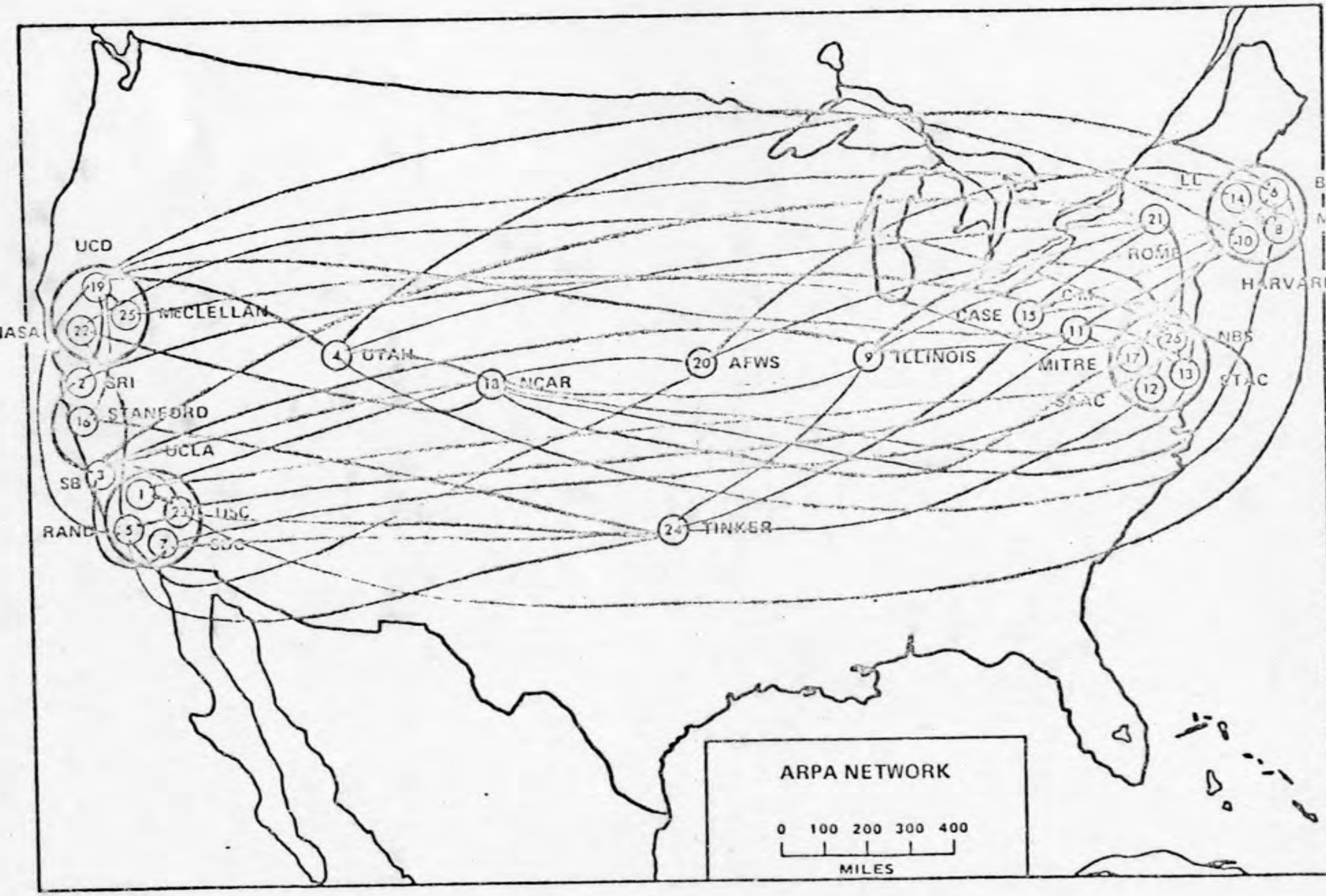


Figure 6.6.2. Best Solution: DDISC = 89,580, ρ = 1.05

TABLE 6.6.2

DISTRIBUTION OF CAPACITIES VERSUS LINK LENGTHS
FOR THE 61 ARC TOPOLOGY OF FIGURE 6.6.2

capacity [kb/sec]	Link length (miles)			
	< 100	100-500	500-1000	> 1000
9.6	0	3	8	20
19.2	1	8	2	6
50	11	2	0	0

Note: each entry represents the number of links which have the specified capacity and are within the specified length range.

topologies tend to produce better $DCONT_{min}$ than low connected topologies. Such behavior agrees with the results of Sections 6.4 and 6.5. However, it should be noticed that, for a given initial topology, the best discrete solution is not obtained from the minimum cost continuous solution (typically, $DCONT_{disc} > DCONT_{min}$); furthermore, there is almost no correlation between DDISC and $DCONT_{min}$. This fact clearly shows that the most critical step in the discrete capacities design, is the continuous-discrete transformation, rather than the determination of a good continuous solution: any effort should be directed therefore to the improvement of such transformation.

In most of the cases, the difference between DDISC and DCONT can be attributed to the presence of little utilized, high cost capacities: in such cases, some clever heuristics (e.g., reduction of the load level, rerouting of some commodities, etc) could be added to the

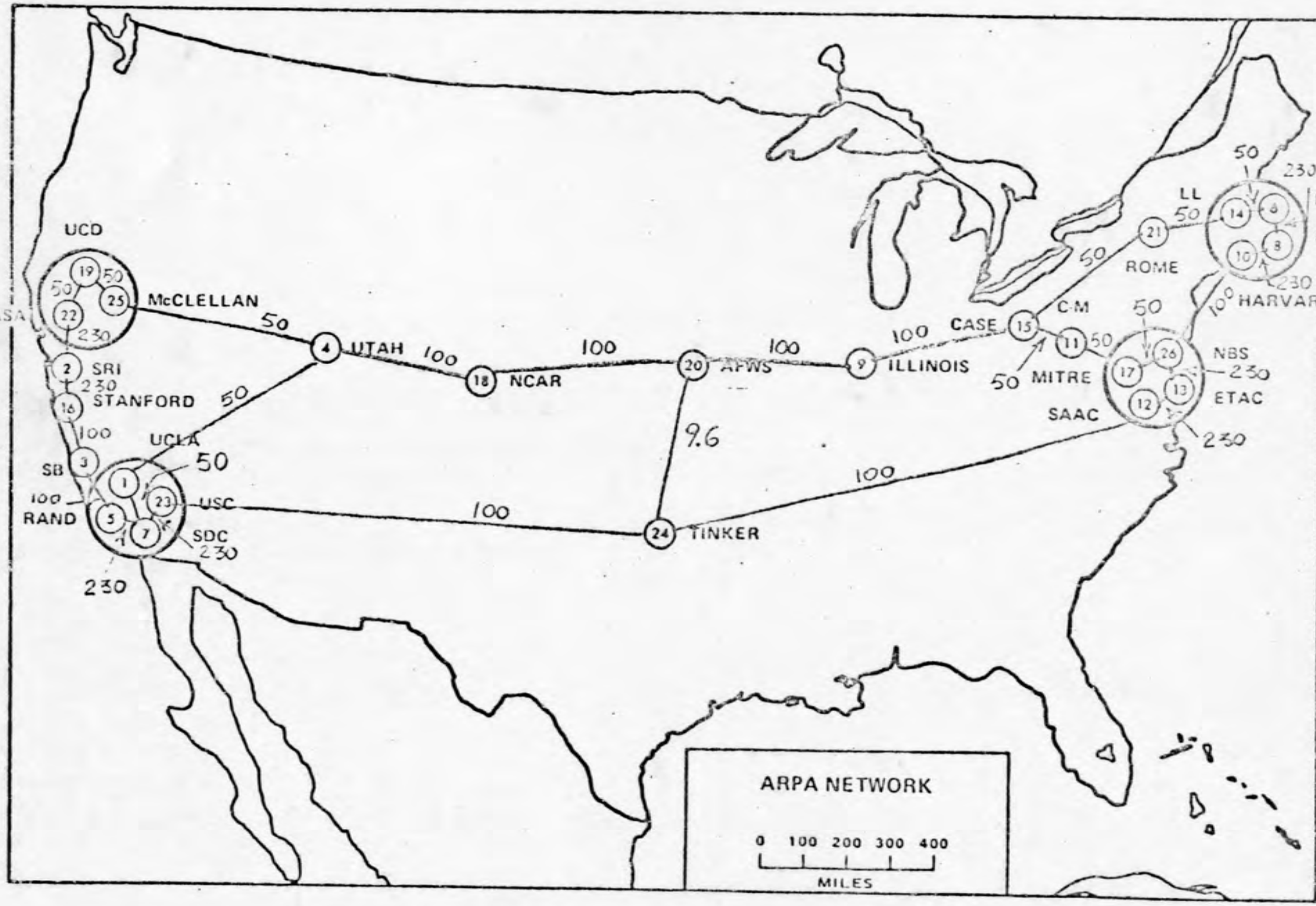


Figure 6.6.3. Second Best Solution: DDISC = 94,288, ρ = 1.00

minimum fit discrete capacity assignment, in order to ensure a uniform utilization of all the channels and consequently reduce the difference between continuous and discrete cost. In any case, the efficiency of a continuous-discrete transformation is very much dependent on the data (number and distribution of the discrete capacity levels, existence of a good concave approximation for the discrete costs, etc.); therefore no general considerations can be made.

It also should be noticed that the best topology is very highly connected and uses almost exclusively 9.6 and 19.2 kb capacities: this fact seems to indicate that, for the specific costs given in Table 5.8.1, better results can be obtained with very highly connected (and certainly not very intuitive!), low capacitated topologies.

Some of the results of Table 6.6.1 have been represented on a ρ versus D diagram in Figure 6.6.4a. An approximate lower bound on D, for any ρ between 1.00 and 1.10, was obtained by joining with a straight line the lower bound $D = 81,000$ for $\rho = 1.00$, and the lower bound $D = 87,000$ for $\rho = 1.10$.* We notice that the excellent solution obtained from the fully connected topology ($D = 89,580$; $\rho = 1.05$) is only 6% from the lower bound; several solutions are available in the range 10-15% from the lower bound. We conjecture that the use of some clever hueristics in the continuous-discrete transformation, the exploration of a broader sample of initial topologies (including very highly connected topologies) and a more accurate** evaluation of ρ

* Such lower bounds were obtained with continuous, α fitted, cost curves (see Section 6.4).

**In order to keep the computation time within reasonable limits, the CBE algorithm allows only 10 FD iterations for the maximization of ρ. A more accurate, a posteriori, evaluation of ρ is recommended for a selected set of solutions.

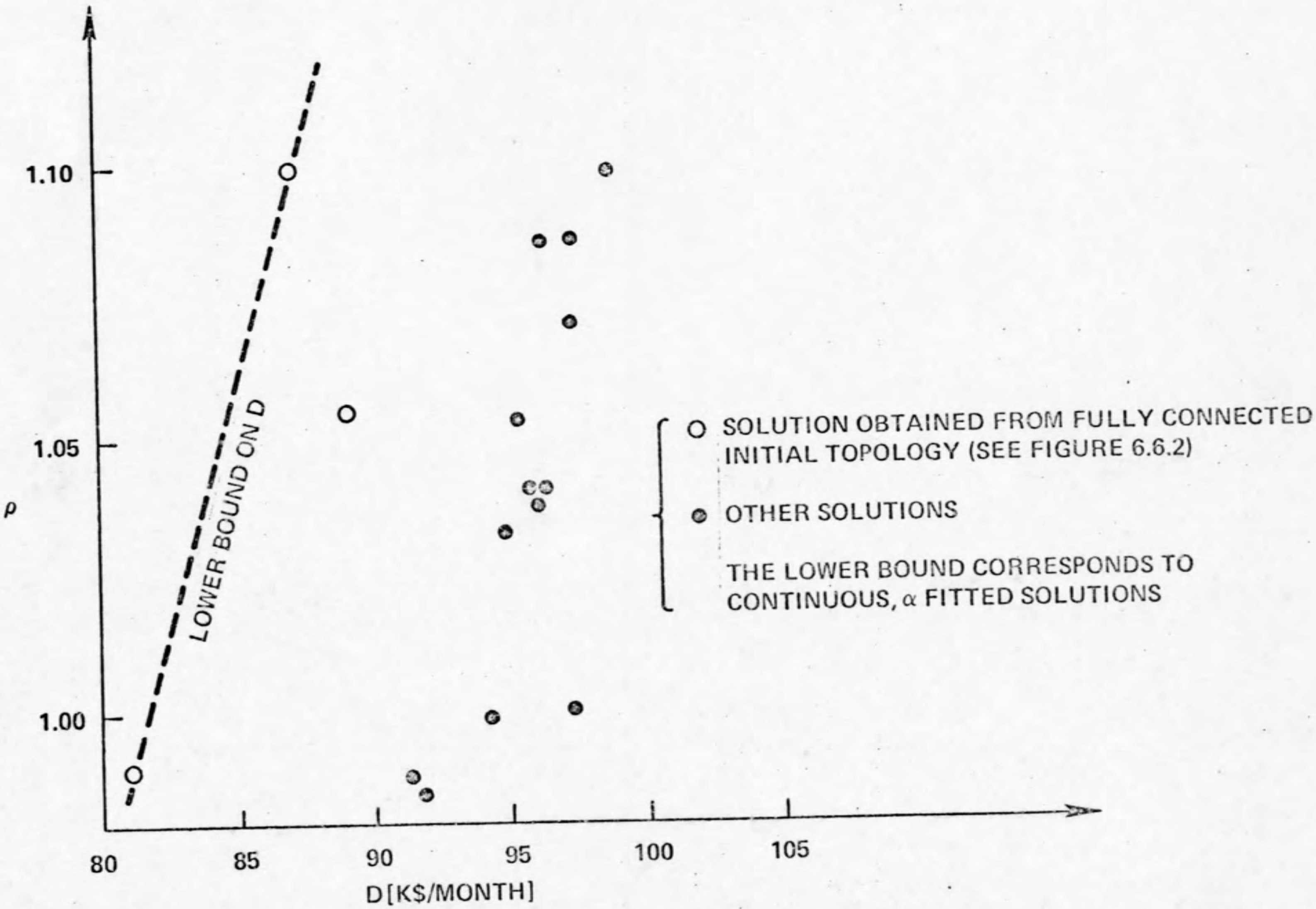


Figure 6.6.4a. Thruput ρ versus D of Some of the Discrete Solutions Shown in Table 6.6.1.

could reduce the gap and provide several solutions in the range 5-10% from the lower bound.

Lower bounds, for α fitted, and a few good discrete solutions have been computed also for a larger range of traffic level $(0.5 \leq \rho \leq 1.10)$. The results are plotted in Figure 6.6.4b. Most of the solutions were obtained from the fully connected initial topology.

The above results and considerations indicate that the CBE method is a valid tool for the topological design, in those cases in which the discrete costs can be reasonably approximated by continuous, concave costs, and the discrete capacities by continuous capacities. If these conditions are not verified, the method can still be used in order to obtain interesting lower bounds.

As an example of lower bound, consider a discrete case in which only one value of capacity C_{i0} can be assigned to link i. In such a case, the cost-cap function for each link is a step function and can be lower bounded by power law cost curves with arbitrary α in the interval (0, 1) (see Figure 6.6.5). We can solve the topological problem for a variety of values of α (possibly different from link to link) and obtain a set of minimum costs $D(\alpha)$ which are lower bounds on the minimum discrete cost. Obviously the tightest lower bound D_{LB} is given by:

$$D_{LB} = \max_{\alpha} D(\alpha)$$

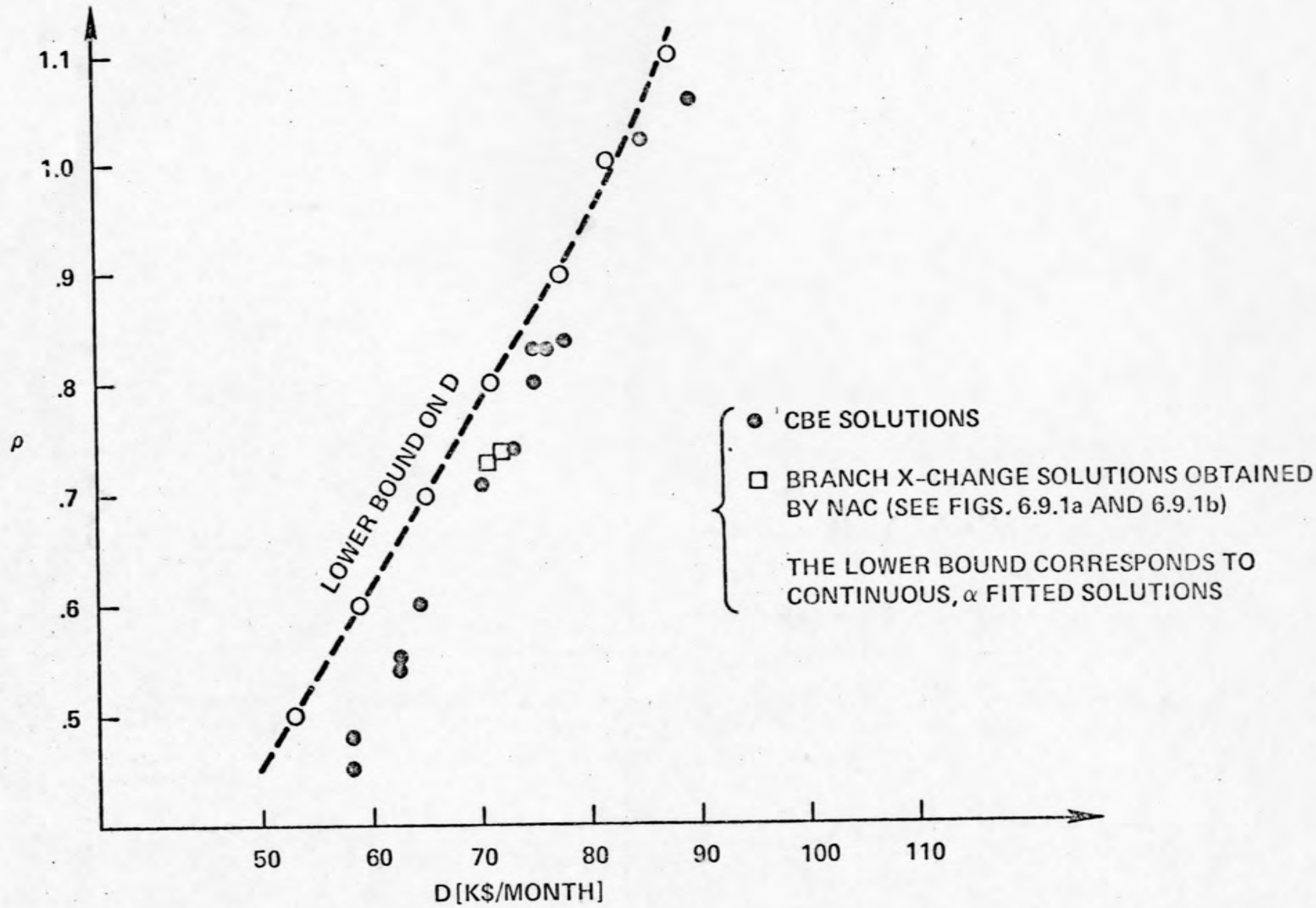


Figure 6.6.4b. Thruput ρ versus Cost D of Some Discrete Solutions Obtained in the Range $\rho = 0.5 \div 1.1$.

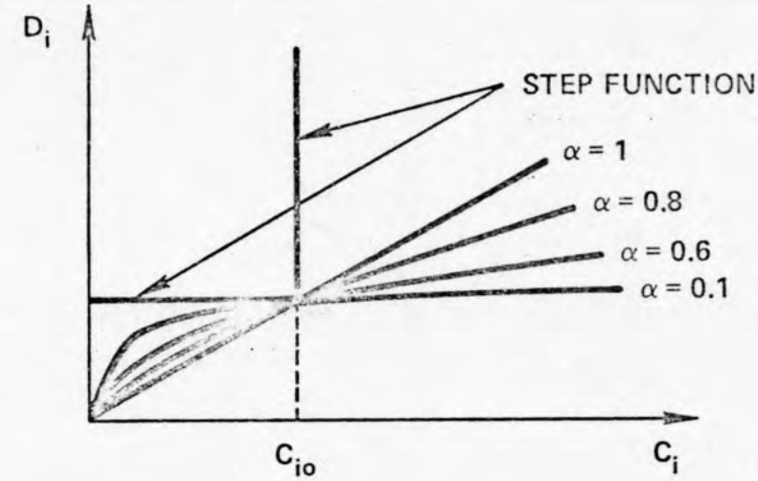


Figure 6.6.5. Lower Bounds on Discrete Costs.

6.7 Partitioning

It is generally recognized that large S/F communication networks should be partitioned into smaller components, for a better efficiency of both design and operation of the network [FULT 72, NAC 70B]. As far as the design is concerned, most of the algorithms presented in this study typically require a computation time proportional to $(NN)^3$ (shortest route computation) and a memory space proportional to $(NN)^2$ (distance matrix, routing tables, etc.); they seem, therefore, to be inadequate to handle problems with $NN > 100$. In such a case, a proper partitioning would transform the original problem into a set of smaller problems, to which the above algorithms are still applicable and for which an economy of computation may be achieved.

In this section, an approach to the topological design of partitioned networks is proposed, and the impact of the techniques so far developed on such a design is discussed.

Several definitions of partioning can be proposed. Here we use the following definition:

- the set of nodes NN is partitioned into "clusters" $NN^{(k)}$ $(k = 1, \dots p)$, such that:

$$NN = \bigcup_{k=1,\dots p} NN^{(k)}$$

$$NN^{(k)} \cap NN^{(\ell)} = \phi \quad \text{for} \quad k \neq \ell$$

$$NN^{(k)} \neq \phi \ , \forall \, k = 1, \dots p$$

- if node i and node j belong to the same cluster, say $NN^{(k)}$, all the traffic, which arrives (or is generated) at node i and is directed to node j, is routed through intermediate nodes $\varepsilon \ NN^{(k)}$. In other words, the traffic between two nodes belonging to the same cluster never leaves the cluster.*

According to such a definition, we can associate with each cluster its own subnetwork, and study it independently from the rest of the network. Notice, however, that the subnetwork carries internal traffic (generated, and with destination, inside the cluster) as well as external traffic (transit traffic, or traffic from internal to external

*Notice that such a definition implies that each subnetwork (i.e., the network connecting the nodes in each cluster) must be connected.

nodes and vice versa). In order to take into account the external traffic, equivalent requirements $r_{ij}^{(k)}$, $\forall$ i, j ε $NN^{(k)}$, must be defined as follows:

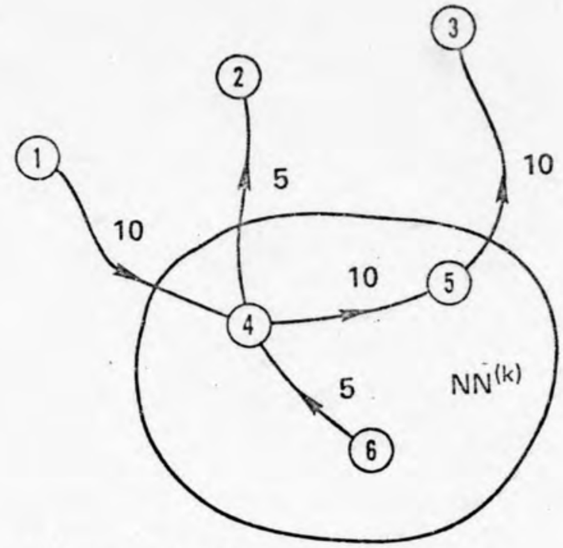
$$r_{ij}^{(k)} \triangleq r_{ij} + \underbrace{\sum_{\ell \notin NN^{(k)}} \sum_{m \notin NN^{(k)}} r_{\ell m} \cdot \sigma_{ij,\ell m}^{(k)}}_{\text{transit traffic}} +$$

$$+ \underbrace{\sum_{\ell \notin NN^{(k)}} r_{i\ell}\, \sigma_{ij,i\ell}^{(k)}}_{\text{outward traffic}} + \underbrace{\sum_{\ell \notin NN^{(k)}} r_{\ell j}\, \sigma_{ij,\ell j}^{(k)}}_{\text{inward traffic}} \tag{6.2}$$

$$\text{where } \sigma_{ij,\ell m}^{(k)} \triangleq \begin{cases} 1 & \text{if the route for commodity } (\ell, m) \ \underline{\text{enters}} \text{ the cluster } NN^{(k)} \text{ at node } i \text{, and } \underline{\text{leaves}} \text{ it at node } j \\ 0 & \text{otherwise} \end{cases}$$

The example in Figure 6.7.1 clarifies the above definition: if we assume that, for cluster $NN^{(k)}$, in addition to the internal traffic, there are 10 units of transit traffic from 1 to 3 and 5 units of outward traffic from 6 to 2, the equivalent requirements $r_{ij}^{(k)}$ have the expression shown in the figure. Definition (6.2) is valid only for a fixed routing assignment; however, simple modifications would extend its validity also to the alternate routing assignment.

If we assume that the external traffic has already been assigned, then we can compute the equivalent requirement matrix $R^{(k)}$ and design the subnetwork $NN^{(k)}$ as a separate entity; the use of



$$r^{(k)}_{45} = r_{45} + 10$$

$$r^{(k)}_{64} = r_{64} + 5$$

$r^{(k)}_{ij} = r_{ij}$ FOR ALL THE OTHER (i,j) PAIRS

Figure 6.7.1. Equivalent Requirements.

equivalent requirements ensures the conservation of the flows between the subnet and the remainder of the network.

Conversely, we can associate with each cluster a supernode, and study the <u>master network</u> that connects all supernodes. In such a study, we only consider the external traffic, i.e., the traffic between nodes belonging to different clusters. Therefore, if m and n are supernodes, we define the composite requirement $r^{(0)}_{mn}$ as follows:

$$\begin{cases} r^{(0)}_{mn} = \sum\limits_{i \in NN^{(m)}} \sum\limits_{j \in NN^{(n)}} r_{ij} \ , & \text{for } m \neq n \\ r^{(0)}_{mn} = 0 \ , & \text{for } m = n \end{cases}$$

At this point, if we associate with each supernode a geographical location, we can design the master network independently of the structure of each individual subnet.

The total average delay T can be expressed in terms of the delays of master net and subnets in the following way:

$$T = \sum_{k=0}^{p} \frac{r^{(k)}}{r} T^{(k)} \qquad (6.3)$$

where: $T^{(0)}$: delay of master net

$T^{(k)}$, $k = 1, \ldots, p$: delay of subnets

$r^{(0)} \triangleq \sum_{m=1}^{p} \sum_{n=1}^{p} r_{mn}^{(0)}$: thruput of master net

$r^{(k)} \triangleq \sum_{i,j \in NN^{(k)}} r_{ij}^{(k)}$: thruput of subnet k

$r \triangleq \sum_{i=1}^{NN} \sum_{j=1}^{NN} r_{ij}$: total thruput

The simple relation (6.3) was obtained using Little's result as shown in Section 1.2. The delays $T^{(k)}$ have an interesting physical interpretation: $T^{(0)}$ is the average delay in the internode communications, and $T^{(k)}$ is the average delay suffered by internal and transit traffic in cluster k.

Similarly, the total cost D can be expressed as follows:

$$D = \sum_{k=0}^{p} D^{(k)} \qquad (6.4)$$

where $D^{(0)}$ is the cost of the master net

$D^{(k)}$, $k = 1, \ldots, p$, is the cost of the kth subnet.

An approach for the topological design of a partitioned network can now be outlined:

(1) Determine the partitions $NN^{(k)}$, $k = 1, \ldots, p$

(2) Determine $\tau^{(k)}$, $k = 0, \ldots, p$, the contribution of component k to the maximum global network delay T_{max} , such that:

$$\sum_{k=0}^{p} \tau^{(k)} = T_{max}$$

It follows that the maximum admissible delay for the master net net, $T_{max}^{(0)}$, is given by (see Equation (6.3)):

$$T_{max}^{(0)} = \tau^{(0)} \frac{r}{r^{(0)}}$$

and for subnet k by:

$$T_{max}^{(k)} = \tau^{(k)} \frac{r}{r^{(k)}}$$

(3) Assign, for each partition, an arbitrary geographical location of the supernode

(4) Compute the composite requirements $\{r_{ij}^{(0)}\}$

(5) Design the master net:

$$\min D^{(0)}$$

$$\text{s.t. } T^{(0)} \leq T_{max}^{(0)}$$

(6) For each cluster k, $k = 1, \ldots, p$:

(6a) Establish the connections between the master net and specific nodes of the cluster

(6b) Compute the equivalent requirements $\{r_{ij}^{(k)}\}$ *

(6c) Design subnet k :

$$\min D^{(k)}$$

$$\text{s.t. } T^{(k)} \leq T_{max}^{(k)}$$

(7) Recompute $D^{(0)}$, using now the real link lengths

(8) Compute total cost D and total delay T :

$$D = \sum_{k=0}^{p} D^{(k)}$$

$$T = \sum_{k=0}^{p} \frac{r^{(k)}}{r} T^{(k)}$$

The network so designed represents a suboptimal solution to our problem. The critical steps of the above approach are (1), (2), (3) and (6a).

The determination of the partitions $NN^{(k)}$ is probably the most critical step. The following factors should be considered:

- the number of partitions p and the number of nodes within each partition determine the efficiency of the design of

---

*This computation can be performed very efficiently using the routing matrix (or routing tables) available from step (5).

master net and subnets respectively[*]

- the thruputs of the different partitions should be fairly balanced
- the geographical diameter of each partition should be considerably smaller than the average distance between different partitions
- the geographical diameter of each partition should be selected according to the link cost characteristic.

In order to illustrate the impact of the link costs on the selection of the diameter, let us consider the costs that we used so far in our applications (see Table 5.8.1 and Figures 5.8.2a and 5.8.2b). In such a case, a diameter of about 100-200 miles would be very convenient for the following reasons: first, the links within each subnet ($\ell \leq 200$ miles) exhibit a strong economy of scale (the good topologies tend to be trees), whereas the links of the master net ($\ell > 200$ miles) have moderate economy of scale (good topologies are highly connected): a uniform economy of scale within each component certainly facilitates the design.[**] Secondly, the links of the master net would be, on the average , much longer than 200 miles, and consequently the cost $D^{(0)}$ would be quite insensitive to the different choices of connecting the

---

[*] A good choice would be to divide the total number of nodes n into $\sqrt{n}$ subsets of $\sqrt{n}$ nodes each (approximating real values with closest integers), so that master net and subnets have the same number of nodes.

[**] We did not apply partitioning to the 26 ARPA sites example discussed earlier in this chapter; however we noticed that, in all minimum cost topologies, the nodes geographically very close were connected by a tree subnet

links of the master net to the nodes of each subnet.

Notice that, if we assume that the cost $D^{(0)}$ is independent of the particular choices made in each subnet, the approach, previously outlined, for the design of a partitioned net, becomes very powerful: given a master net solution $(D^{(0)}, T^{(0)})$, for each subnet k we can determine a variety of nondominated solutions $(D^{(k)}, T^{(k)})$ (see Section 3.6). In order to obtain the best global solution (or a variety of good global solutions), the solutions of the individual subnets are combined using Dynamic Programming or Lagrangian Decomposition techniques (see Chapter 3). This approach is possible because, with the above assumption, the problem becomes separable (see Equations (6.3) and (6.4)).

We conclude this section by mentioning that the design of a partitioned network allows much more flexibility than the design of a nonpartitioned one: for instance, in the design of each subnet specific reliability and delay criteria, different from case to case, could be implemented; also, the addition of a node to the net would probably require the new design of a subnet only.

On the other hand, it should be mentioned that partitioning adds new, difficult problems to the topological design (e.g., choice of partitions, choice of connections between master net and subnets, etc.). However, we conjecture that for large systems (like those to which partitioning is typically applied) the minimum is very broad and not extremely sensitive to changes in the above choices: if this conjecture is true, the trial of a convenient number of different alternatives should produce satisfactory results.

Network partitioning also effects the message routing. We already mentioned that, in our scheme, the traffic between nodes belonging to the same partition never leaves the partition. A more general discussion, regarding also the impact of partitioning on adaptive routing policies, can be found in [FULT 72].

6.8 Extensions

In this section we extend some of the techniques used in the CBE method. In particular, we discuss the random generation of initial topologies and the insertion of arcs.

6.8.1 Random Generation of Initial Topologies

We already mentioned that in some applications (e.g., strong economy of scale) it is more convenient to apply the CBE method to a large selection of initial topologies with an appropriate degree of connection, than to the initial fully connected topology. We need, therefore, a routine that generates random initial topologies, which are likely to contain low cost topologies as subgraphs (i.e., that exclude obviously bad arcs) and which have a variable degree of connection.

A random topology routine was discussed by Steiglitz et al. in [STEI 69]; however, that routine is not very adequate for our purposes, its primary goal being reliability rather than low cost.

The routine, that we propose here, constructs minimal spanning trees, corresponding to link lengths assigned at random. The minimal spanning tree is the low cost "frame" of the initial topology, and can be augmented with the insertion of properly chosen arcs, until the required degree of connection is obtained.

Random lengths, to be used for the minimal spanning tree construction, can be generated in many different ways. For instance, assume that the total flow in arc (i, j) is allowed to vary between $(f_{ij})_{min}$ and $(f_{ij})_{min} + \Delta_{ij}$; the random length ℓ_{ij} can be defined as follows:

$$\ell_{ij} \triangleq d_{ij} \left((f_{ij})_{min} + \alpha\Delta_{ij}\right)$$

where: $d_{ij}(\cdot)$ is the cost-cap function for arc (i, j)

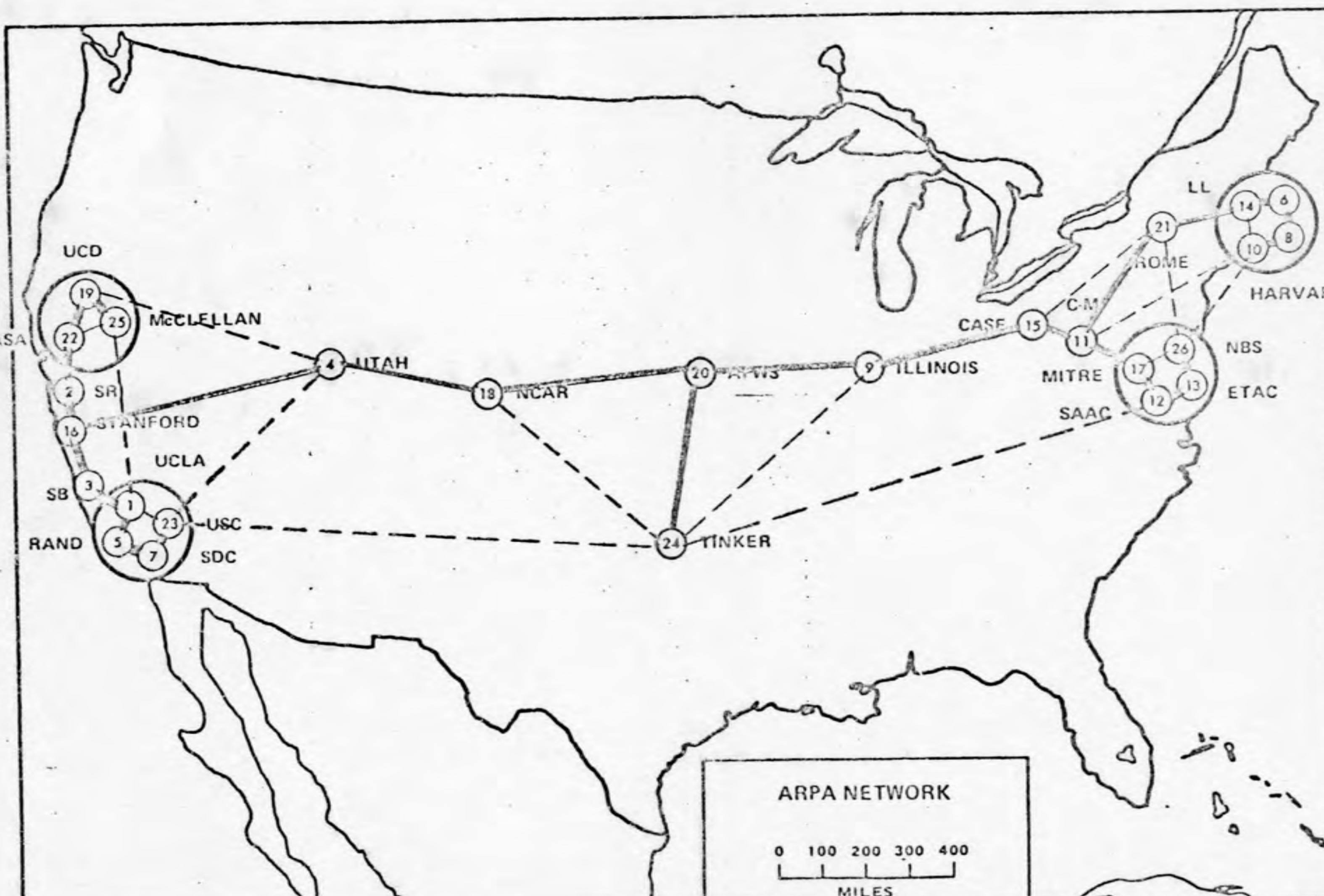
α is a random coefficeint with uniform distribution between 0 and 1 .

Another proper definition of random link length is provided by the geographical length, to which a random term is added.

Additional arcs can be inserted in the tree, with priority given to the arc that minimizes the ratio (geographical distance/topological distance), where geographical distance is the direct distance* between the terminal nodes,and topological distance is the length of the shortest path between such nodes in the present topology. In this way we favor the creation of "shortcuts."

An example of initial topology is shown in Figure 6.8.1: the minimal spanning tree was constructed assuming link lengths $\triangleq$ geographical distances. Such a topology showed very good performance for values of α between 0.5 and 1 (cfr., in Tables 6.4.2 (a) and (b), the entry corresponding to $NA_0 = 40$).

*The notion of distance could be replaced by a proper notion of cost.



—— minimal spanning tree
---- inserted arcs

Figure 6.8.1. Example of Initial Topology with 40 Arcs

In the applications that require 2-connectivity, good initial topologies can be obtained by inserting arcs between the leaves of the tree. In particular, we obtained good results from topologies which consisted of the original minimal spanning tree combined with a second minimal tree spanning the leaves of the first tree (see Figure 6.8.2); notice, by the way, that the combination of the two trees ensures 2-connectivity.

The generation of an initial topology at a given iteration could also use the information provided by the solutions of the previous iterations, in the following way: determine, for each link, the frequency with which it appeared in the previous solutions, and keep the "most frequently used" links in the initial topology. Such links are more likely than others to belong to the optimal topology.*

6.8.2 Insertion of Arcs

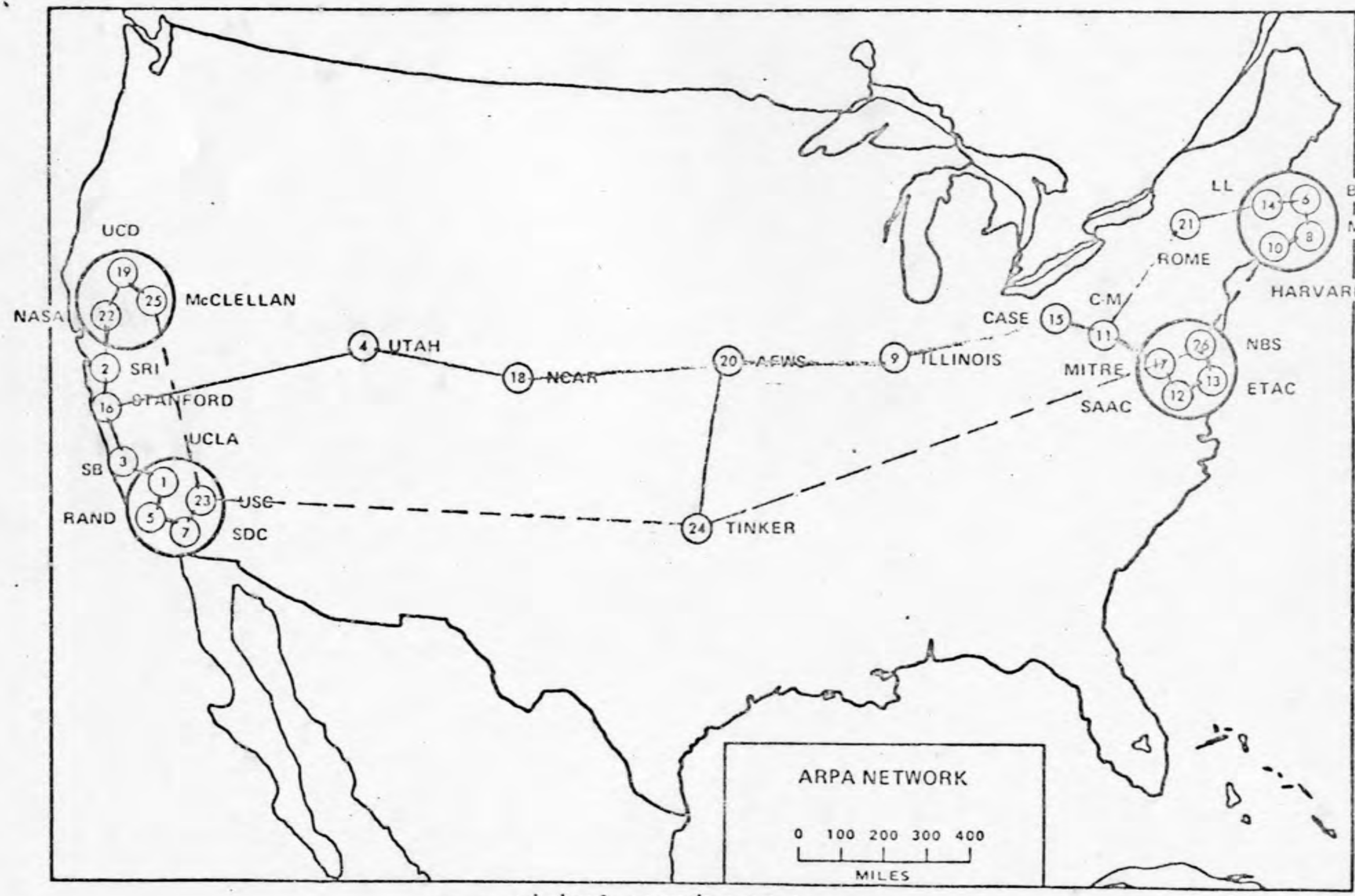
As we already mentioned, the insertion of arcs is, in some cases (2-connectivity, low α, etc.), an essential complement to the CBE method. In fact, the CBE method does not insert arcs automatically, as the marginal cost of an arc with zero flow is ∞ (see Section 5.6); therefore, arcs can be inserted only by forcing their marginal costs (i.e., equivalent lengths) to assume values $< \infty$.

Typically, there are $(NN)^2$ possible arc insertions, unless some specific requirement (e.g., 2-connectivity) restricts the choice to a smaller set.** In any case, the number of choices is generally

---

* For a similar approach, see Lin [LIN 66].

** We exclude insertion of arcs in parallel to already existing arcs.



—— minimal spanning tree
---- minimal tree spanning the leaves

Figure 6.8.2. Example of 2-Connected Initial Topology

too large to permit the systematic implementation of all possible insertions: some selection criteria have to be established.

Essentially, we need an estimate of the variation in cost ΔD corresponding to each arc insertion; such an estimate must be reasonably good and, at the same time, economical (we have $(NN)^2$ of them to do!).

In Appendix D we describe some estimates, which are applicable to concave problems. Essentially, such estimates take advantage of the fact that the objective function is concave and that the flow configuration is an extremal flow (see Section 4.3). These estimates are reasonably economical: they require an amount of computation on the order of $(NN)^3$ for the examination of all $(NN)^2$ possible arc insertions (notice that an equivalent amount of computation is required by each CBE iteration). We also believe that such estimates are reasonably accurate, even though we did not verify them on any large network: in fact, we make the very realistic assumption that the insertion of arc (i, j), say, produces only a <u>local</u> flow perturbation (i.e., it does not perturb commodities which flow on routes two or more links apart from node i or node j).

After arc (i, j) has been selected for insertion (according to some criteria which include the ΔD estimate) the equivalent length ℓ_{ij} is set to a proper value $< \infty$ (see Appendix D) and the CBE algorithm restarted, using the equivalent length vector $\underset{\sim}{\ell}$ so modified.

6.9 Conclusion

The CBE method is a topological design method applicable to multicommodity flow networks in which the objective, to be minimized, is (or can be reasonably approximated with) a continuous, concave function $D(\underset{\sim}{f})$.

The method is based on the key notion of vector of equivalent lengths $\underset{\sim}{\ell}$ (where $\ell_i \triangleq \partial D/\partial f_i$), which indicates the direction of steepest flow deviation (see Chapter 4). The method finds local minima of $D(\underset{\sim}{f})$; its impact on the network topology is due to the fact that, for the particular nature of $D(\underset{\sim}{f})$, * the arcs with low utilization are gracefully eliminated, as the correspondent lengths become ∞ . On the other hand, proper considerations, also based on $\underset{\sim}{\ell}$, permit us to insert arcs, which are likely to reduce $D(\underset{\sim}{f})$.

A peculiar feature of the CBE method is, therefore, the ability to perform topological modifications by taking into account the complex interaction between cost and flow assignment (a measure of such an interaction is given, as a first approximation, by $\partial D/\partial f_i$). In a sense, during the application of the CBE method, the flow itself designs the network topology, in the attempt to find the most convenient routes.

On the other hand, branch X-change methods (see Section 6.3) perform topological modifications systematically (or on the basis of some reasonable considerations), with little or no attention to the cost-flow interaction.

Clearly, the CBE method cannot be applied to those cases in which there is no meaningful notion of marginal cost (e.g., only one discrete capacity level; pure set up costs): in such cases, branch

---

*For an efficient topological reduction we also require that:

$$\lim_{f_i \to 0} \frac{\partial D(\underset{\sim}{f})}{\partial f_i} = \infty , \forall i$$

X-change methods are the only alternative. Also, branch X-change methods can be conveniently applied to problems where the evaluation of flow and cost after each topological transformation is straightforward (e.g., tree structures [FRAN 71A]).

In this chapter we discussed various applications and pointed out cases in which CBE performs very well (power law cost curves with $\alpha \simeq 1$); cases in which CBE gives reasonable results ($\alpha \simeq 0.6$; discrete problems with a sufficient number of capacity levels); cases for which the CBE method is not adequate (pure set up costs; $\alpha \to 0$; only one discrete capacity level).

An interesting comparison between CBE and branch X-change approach is possible if we analyze the method proposed by Frank et al. [FRAN 70] for the design of minimum cost computer network topologies. The method operates topological transformations with branch X-change techniques. After each transformation, the flow is assigned to shortest routes, computed with uniform link lengths $\ell_i = 1, \forall i$. In order to provide a more compact representation of such a method, let us consider a fully connected topology, and let us label the arcs in such a topology from 1 to NA. We can now associate with a specific topology a vector of equivalent lengths $\underset{\sim}{\ell}$, such that $\ell_i = 1$, if arc i belongs to the topology, and $\ell_i = \infty$ otherwise; for example:

$$\underset{\sim}{\ell} = (1, 1, \infty, \infty, \ldots, 1, \infty, \ldots, 1)$$

A topological transformation can be, then, represented as a change of some entries of $\underset{\sim}{\ell}$ from 1 to ∞ and vice versa. Similarly, flows are routed on shortest paths computed according to $\underset{\sim}{\ell}$.

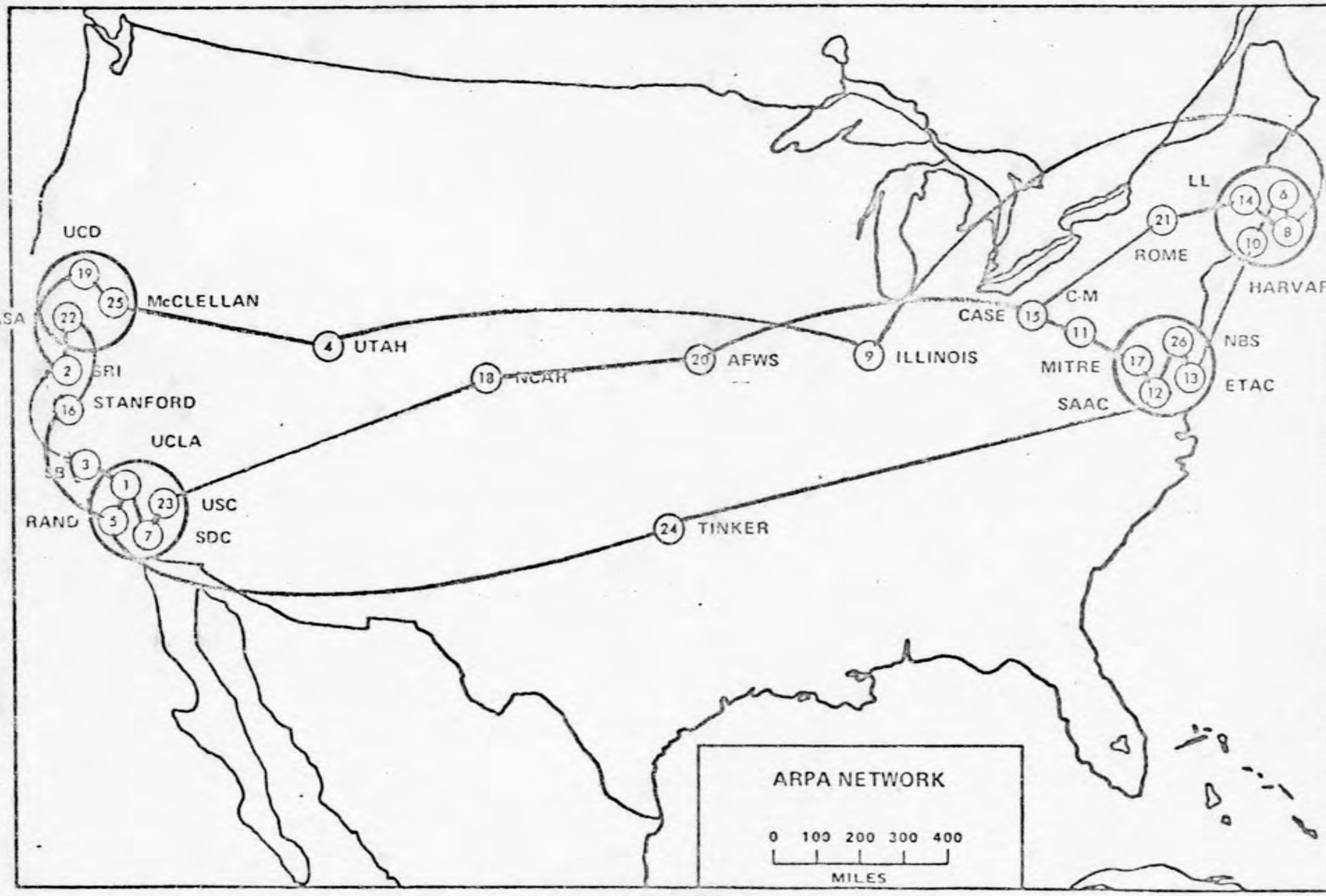
If we now apply the CBE method to the same problem, we have an equivalent lengths vector $\underset{\sim}{\ell}$ of the following form:

$$\underset{\sim}{\ell} = (\ell_1, \ell_2, \infty, \infty, \ldots, \ell_k, \infty, \ldots, \ell_{NA})$$

The vector $\underset{\sim}{\ell}$ now depends, not only on the topology, but also on cost and flow characteristics of the arcs. As for the topological transformations, arc i is automatically eliminated when ℓ_i becomes ∞ ; on the other hand, if $\ell_i = \infty$ and cost-flow considerations indicate that the insertion of arc i would reduce $D(\underset{\sim}{f})$, length ℓ_i is set to a proper value $< \infty$.

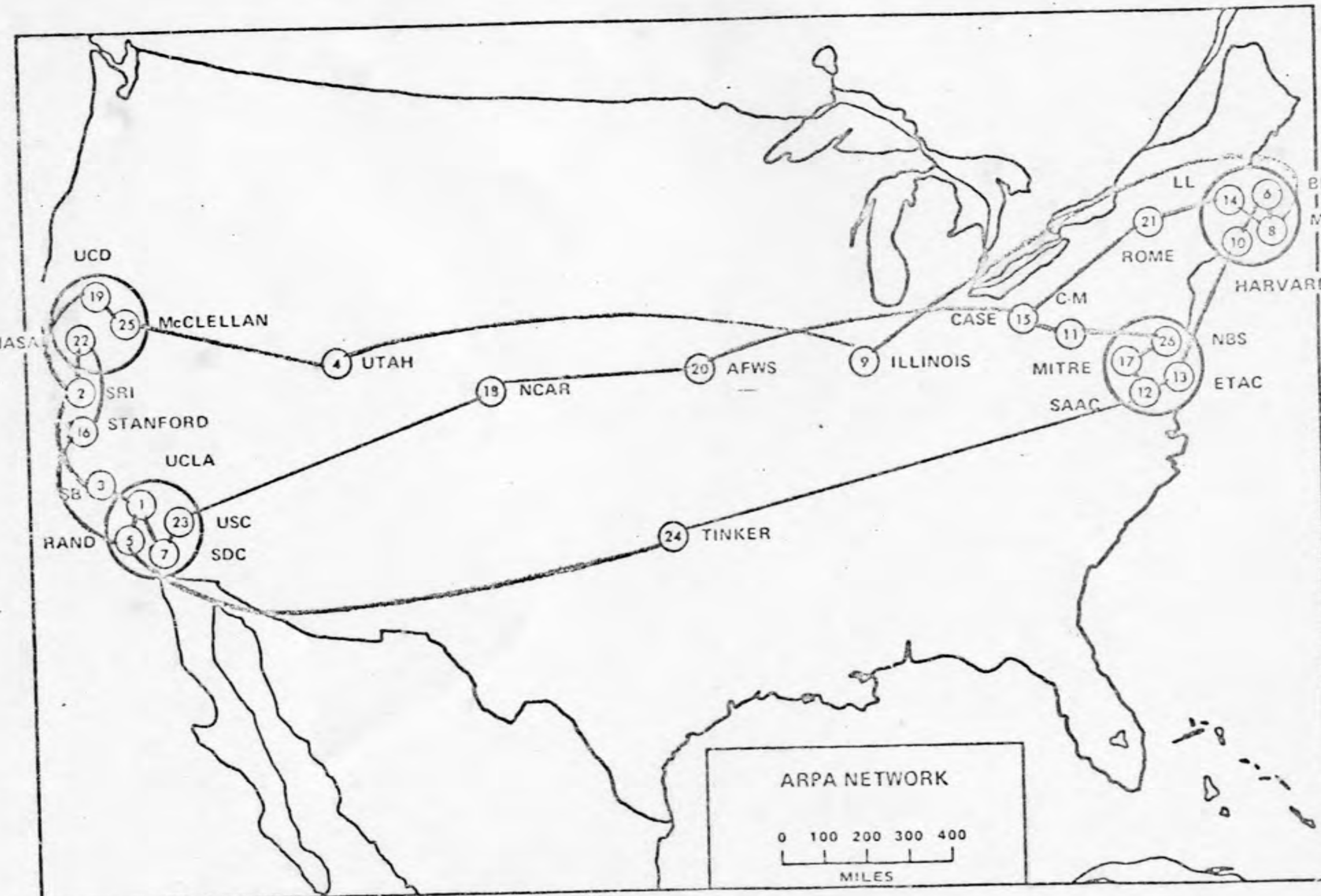
The first method can be therefore considered as an extreme simplification of the CBE method, in which $\underset{\sim}{\ell}$ carries no information about marginal costs. The method is certainly a good heuristic approach to problems that have no notion of marginal cost; in particular, the method was applied by NAC [NAC 71B] to the design of minimum cost ARPA topologies, in which only 50 kb capacities were allowed, and produced very satisfactory results. Two solutions obtained by NAC for the familiar 26 nodes problem,* are shown in Figures 6.9.1a and 6.9.1b and are compared to the CBE discrete solutions in the ρ vs. D plot of Figure 6.6.4b.

*The costs shown in Table 5.8.1 were used.



All capacities 50 kb
D = 70,512 [kR/month]
r = .73 [kb/sec x node pair]

Figure 6.9.1a 26 Node, 29 Arc ARPA Topology, Designed by NAC



All capacities 50 kb
D = 72,538 [k$/month]
r = .74 [kb/sec x node pair]

Figure 6.9.1b 26 Node, 30 Arc ARPA Topology, Designed by NAC