A Dynamic Programming Based Method for Developing Optimal Microgrid Architectures

Joydeep Mitra Shashi B. Patra Satish J. Ranade
Electric Utility Management Program
The Klipsch School of Electrical and Computer Engineering
New Mexico State University
Las Cruces, New Mexico 88003, USA

Abstract - Distributed Energy Resources (DER) have been receiving increasing attention over the past decade. DERs are most effective at or near centers of utilization. Eventually, as their penetration increases, they will be interconnected in a grid-like fashion for stability and enhanced reliability. These grids, called 'microgrids,' will be expected to operate in both grid-connected and islanded modes. This paper presents a rational method of building microgrids optimized for cost and subject to reliability constraints. The method is based on dynamic programming and consists of determining the optimal interconnection between microsources and load points, given their locations and the rights of way for possible interconnections. A new approach, called 'unit link addition,' is also introduced. The paper describes the formulation, as well as its implementation on a parallel computing environment. Applications to both possible situations — design of a microgrid from scratch and expansion of an existing distribution system — are described. The method is demonstrated using a 22-bus system.

Keywords - reliability, distributed energy resources, microgrid, dynamic programming, parallel computing

1 Introduction

OVER the last decade, distributed energy resources (DER) have been receiving increasing attention as alternatives to centralized generation [1]–[3]. This is partly because of the many advantages that DERs offer, such as compact size, modularity, portability, lower emission, sometimes higher conversion efficiency, and lower costs and losses in transmission and distribution (T&D), and partly because they are often an expedient alternative to the expensive and long drawn out processes of expanding or upgrading T&D. In the next few decades, the penetration of microgrids is expected to grow dramatically. The bulk of DERs will be integrated into existing distribution systems, with necessary upgrades. To maximize the economic and reliability benefits of these installations, distribution systems that have very often been radially connected will evolve into networked microgrids .

Significant research has been conducted in the areas of transmission expansion planning [4] and developing microgrids [1]–[3], [5]–[8] . Most notable is the "CERTS Microgrid Concept" which focuses on a self-sustained heat and power supply to a compact cluster of loads. Many of the approaches have focused on the optimal placement of DERs in the distribution network [1]–[3]. Except [3], these approaches have assumed and have attempted to preserve the fundamentally radial structure of the distribution grid. In contrast, we address the issue of distribution network expansion with a reliability criterion using deployment of distributed resources. Our approach is directed toward the development of microgrids that are networked in structure and conform to the US Department of Energy's vision of microgrids that can operate in both grid-connected and islanded modes [9], [10].

This paper describes a rational method of developing microgrid architectures in a manner that maximizes the economic benefits of DER deployment, taking into account the anticipated growth in the region served by the microgrid. This will lead to more efficient utilization of resources and higher system reliability than ad hoc installations. The rational approach is developed as follows. First, a method is developed that meets stipulated reliability requirements, given the locations of sources and loads, rights of way, the capacities and availabilities of the sources, and the sizes of the loads. The approach is then extended to the case where an optimal microgrid is developed starting with an existing distribution system.

Dynamic Programming has been used in transmission planning [11] and distribution system planning [12]. Our method incorporates an explicit reliability criterion and applies to networked distribution architectures. The DP formulation presented in this paper seeks to make the most economical connections that will satisfy the stipulated reliability criteria. The reliability index used is the energy index of reliability (*EIR*), i.e., the supplied energy as a fraction of the energy demand. The problem is decomposed into stages by means of a unit link addition algorithm developed by the authors. This algorithm consists of determining at each stage the optimal pair of nodes between which to connect a unit link, which is a link of fixed impedance per unit length and cost per unit length. Stronger connections consist of multiple unit links in parallel. Because of the size of the problem, measures were implemented to trim the search-space and to parallelize the algorithm. Further, recognizing that it is the final stage that is of interest, a scheme is formulated to eliminate the need for storing intermediate DP stages. This reduces the memory usage significantly and eliminates the need for backtracking. The algorithm has been implemented on a BeoWulf cluster. To further improve the efficiency of the parallel algorithm, a load sharing scheme is devised that takes into account the time spent on communication between the nodes in the cluster.

This paper describes the method and the implementation, and illustrates it by means of an example.

2 System Modeling

In this section, the reliability models of the microgrid components are briefly described. Although simple, two-state models have been assumed for the purpose of demonstration, the method itself is capable of accommodating more detailed and complex component models.

Generators: These are modeled as two-state devices. Each generator i is described by its maximum generating capacity G_{max} and its forced outage rate FOR_i.

Load: For the purpose of system planning, the load has been assumed to remain constant at the coincident peak.

Transmission Lines: The basic element of the transmission network is the *unit-link* which is a transmission line with the following characteristics:

> A *unit-link* connecting a given pair of buses is a line of fixed capacity, fixed cost per unit length, fixed impedance per unit length, and of length corresponding to the right of way between the buses.
>
> Unit links between different bus pairs will have different lengths, impedances and costs, but same capacity.
>
> A *link* between a given pair of buses can consist of one or more (an integral number of) unit links connected in parallel between that bus pair. The cost of the link is equal to the total cost of the unit links that constitute the link plus other costs. For the sake of simplicity, other costs have been neglected.

Network Model: A linearized network model in the form of DC Load Flow has been used in this work.

3 Problem Formulation

This work aims at determining the optimal network configuration which satisfies a specified reliability requirement. Mathematically, this problem can be posed as follows.

Minimize:

$$J = \sum_{i,j} J_{ij} \times x_{ij} \qquad \forall\, 1 \le i < j \le N_n \tag{1}$$

subject to:

$$EIR > R_0 \tag{2}$$

where

J = cost of the transmission network
J_{ij} = cost of a unit link between nodes i and j
x_{ij} = number of unit links in parallel between nodes i and j
N_n = number of nodes in the system
EIR = Energy Index of Reliability, and
R_0 = minimum required reliability.

Eq (1) assumes that each possible path between the nodes of the system is available as a right of way. In general, this may not be true. In such a case the above equation may be rewritten as:

$$J = \sum_{i=1}^{N_\ell} J_i \times x_i \tag{3}$$

where

J_i = cost of a unit link for the i^{th} right of way
x_i = number of unit links in parallel for the i^{th} right of way
N_ℓ = number of rights of way

This problem is amenable to a stage-wise decomposition. In each stage one unit link is added to the existing network. Therefore dynamic programming becomes a suitable tool for the solution method. This approach with the concept of the *unit-link* gives the optimal network along with the capacity of each link.

4 Solution Strategy

The solution is obtained in a stage-wise fashion where in each stage the network is incremented by a unit link. Before we describe the strategy, first we need to define the different structures of the DP.

4.1 The Dynamic Programming Formulation

DP Stage: Each stage represents the total number of unit links that have been added to the system so far. This means that stage S_k of the DP has utilized k unit links to build a network configuration. These k unit links can be distributed among the possible rights of ways in different ways, giving different system configurations. In light of Eq (3), all configurations in stage S_k will be such that

$$\sum_{i=1}^{N_\ell} x_i = k \tag{4}$$

It is appropriate to remark here that the unit link as defined earlier is central to this formulation. The concept of the unit link, and its formulation, makes it possible to deal with the dependencies between the line characteristics, such as those between capacity and impedance, impedance and length, length and cost, and capacity and cost.

DP State: Each configuration mentioned above has an associated cost and has an associated measure of reliability. This measure of reliability is chosen as the system state.

DP Decisions: As mentioned earlier, consecutive stages of the DP are built by adding more and more unit links. Each possible unit link that can be added across the rights of way is an alternative. These are defined as the "decisions" of the DP.

It is quite possible that in a given stage, different configurations can have the same value of reliability, though with different costs. Therefore, it is very important during the building of stages to check for such states. Whenever a new state for the next stage is generated by testing a decision, a check is performed to see if any other state with the same value of reliability exists. If such a state exists and has a higher cost, then it is replaced with the newly generated state. In, other words, this ensures that the configuration represented by a state in a given stage is the most optimal way of achieving the reliability it offers by adding as many unit links as depicted by the stage.

4.2 Reliability Evaluation

In a distribution network, a measure of reliability based on the energy supplied is more appropriate. Therefore, an energy index of reliability is chosen. Given a set of generating units, the expected minimum curtailment evaluated over generation contingencies up to first order is used as the measure of system reliability.

As mentioned before, a linearized power flow model has been used. This is implemented in the form of a Linear Programming problem. The aim is to determine a dispatch which minimizes the load curtailment subject to generation availability and network constraints as follows [13]:

$$\text{Loss of Load = Min } \left(\sum_{i=1}^{N_b} C_i \right) \tag{5}$$

subject to:

$$\begin{aligned} \hat{B}\theta + G + C &= D \\ G &\leq G^{max} \\ C &\leq D \\ b\hat{A}\theta &\leq F_f^{max} \\ -b\hat{A}\theta &\leq F_r^{max} \\ G, C &\geq 0 \\ \theta & \quad \text{unrestricted} \end{aligned} \tag{6}$$

where

N_b = number of buses
N_t = number of transmission lines
C = N_b-vector of bus load curtailments
C_i = i-th element of C, i.e., unsatisfied demand at bus i
D = N_b-vector of bus demands
G^{max} = N_b-vector of available generation at buses
F_f^{max} = N_t-vector of forward flow capacities of transmission lines
F_r^{max} = N_t-vector of reverse flow capacities of transmission lines
G = N_b-vector of dispatched generation at buses
θ = N_b-vector of bus voltage angles
b = $N_t \times N_t$ primitive (diagonal) matrix of transmission line susceptances
$\hat{A}$ = $N_t \times N_b$ element-node incidence matrix
$\hat{B}$ = $N_b \times N_b$ augmented node susceptance matrix $= \hat{A}^T b \hat{A}$

Although Eq. (6) is a standard representation of the DC flow constraints, it is usually necessary to fix one of the bus angles at zero to be able to obtain a feasible solution.

The above minimization procedure is performed for each contingency. In this work, first order generation contingencies have been considered. Higher order contingencies and transmission line contingencies are a simple extension to the reliability evaluation module.

Let LOL_i be the loss of load obtained for the i^{th} contingency, with a probability of $prob_i$. Then the expected power not served is given by:

$$EPNS = \sum_{i=1}^{N_c} LOL_i \times prob_i \tag{7}$$

where

$EPNS$ = Expected Power Not Served
N_c = Number of contingencies

The reliability of the network is then given by:

$$EIR = 1 - \frac{EPNS}{D_T} \tag{8}$$

where

EIR = Energy Index of Reliability
D_T = Total Power Demand

Normally, the EIR is computed using the equation:

$$EIR = 1 - \frac{EUE}{E_T} \tag{9}$$

where

EUE = Expected Unserved Energy
E_T = Total Energy Demand

Equation (9) becomes equivalent to (8) when the demand is constant over the period of interest. While this method is amenable to any suitable representation of load, in this work the peak coincident load was used. This was considered appropriate for system planning.

4.3 Computer Implementation

4.3.1 Data Structures

Though the problem can be solved in a stage-wise fashion, it is only the final stage in which we are interested. Therefore, given the present stage, there is no need to store stages other than the current one, which is used to build the next stage. So, along with storing the system reliability, the DP State contains additional information regarding the network configuration.

This information is stored in an array. The dimension of this array is equal to N_ℓ. Each element of the array stores the number of unit capacity links that have been allocated to the corresponding right of way.

It is evident that the entire trajectory or series of decisions leading to the present state is compactly encapsulated in this array. This approach eliminates the need for backtracking. Further, this is memory efficient as we do not have to store the previous stages.

4.3.2 *Treatment of the State Space*

The reliability of the system is used as the state of the DP state. In this situation, the number of states that any stage of the DP can have is very large, limited by the precision of the numbers used. But, we recognize that reliability indices in power systems are typically computed to a finite precision. As a result, all network configurations that have the same reliability index, to a given precision, are treated as configurations that offer the same reliability. In this work, for the sake of demonstration, reliability indices were computed to four decimal places.

4.3.3 *Parallelization of the DP*

Dynamic Programming is inherently parallelizable. At any given stage, the set of decisions must be tested for each state in the current stage. The testing of decisions for each state is independent of the computations for the other states. So the entire computational burden can be divided among many processors.

The organization consists of a main processing node and a group of auxiliary processing nodes. The main node divides the set of states and allocates them to the auxiliary nodes in a manner which is described in section (4.3.4). Each auxiliary node builds its portion of the next DP stage. After all the nodes have finished computation, results are communicated back to the main node. The main node then merges the individual results into a single DP stage. It is noteworthy that each auxiliary processor performs its share of searching for states that offer same reliability but with different costs, and so does the main node after receiving the partial results.

4.3.4 *Optimal Task Allocation*

During the implementation of the parallel algorithm, it was found that the communication of the states from the main node to the auxiliary nodes and back took considerable time. The data was distributed sequentially, where each processors had to wait until data had been distributed to the previous processors. So a scheme was devised where data is not distributed equally, but depending on the rank of the processing node.

Let the number of processors be p. Then the processors are numbered $0, 1, \cdots, p-1$. The main processor has the rank 0 while all other processors are auxiliary processors. But, the main processor can also process states and can act as an auxiliary node after the computational task of the current stage has been parallelized and tasks have been allocated to the other nodes.

Let T_t be the time taken to transmit one state from processor 0 to an auxiliary node. Note that the transmission time is meaningless when node 0 is allocating a task to itself.

Let T_c be the time taken to process one state. This time is equal to the time required for testing the decision set against this state and for building new states.

Let n_i be the number of states allocated to the i^{th} processor.

The time-line for one execution loop for the parallelized process is shown in Fig. 1.

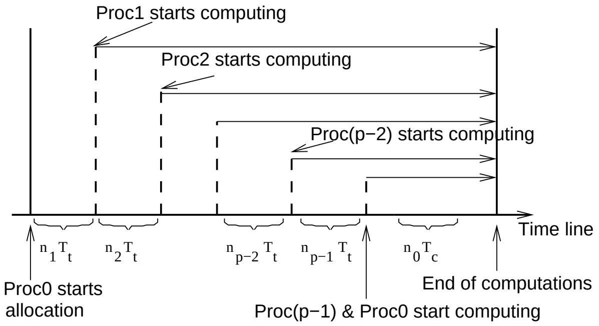

Figure 1: Time-line for task allocation

From Fig. 1, the following equations can be immediately derived:

$$\begin{aligned}
n_{p-1} \times T_c &= n_0 \times T_c \\
n_{p-2} \times T_c &= n_{p-1} \times T_t + n_0 \times T_c \\
n_{p-3} \times T_c &= (n_{p-2} + n_{p-1}) \times T_t + n_0 \times T_c \\
&\vdots \\
n_1 \times T_c &= (\sum_{i=2}^{p-1} n_i) \times T_t + n_0 \times T_c \\
\sum_{i=0}^{p-1} n_i &= N_s \qquad (10)
\end{aligned}$$

where N_s is the total number of states in the current stage. The solution to this set of equations gives us the number of states for each processor. When the solutions are not integral, as they are supposed to be, then a reasonable approach is to truncate the numbers, and then adjust the number of states for the last processor accordingly.

4.3.5 *Parallel Computing Environment*

The computer program was implemented and executed on a BeoWulf cluster consisting of 10 computers. Each computer was a dual Pentium III machine clocked at 1 GHz and had 128 MB of RAM. The parallelism was achieved using the *Message Passing Interface* (MPI) In particular, the MPICH implementation of the MPI was used.

4.4 *Algorithm*

This problem is solved in two phases, which are described below.

Phase I

This phase is the computationally intensive one. Therefore, this stage was implemented using a parallel paradigm. Given the current stage, the states are grouped into blocks which are allocated to the auxiliary nodes in a manner as described in Sec. (4.3.4).

After the tasks have been allocated, the auxiliary nodes build their share of the next stage. When the computations are finished, the results are sent to the main node. The computational details are described later in this section.

During the process of receiving the results from the auxiliary nodes, the main node tests for the existence of states having same reliability but different costs as described in Sec. (4.1). Other than this, the main node also checks whether one of the states in the stage being built has just just crossed the threshold reliability. If the test returns false, then it releases the memory allocated to the current stage, renames the next stage, which is the stage that was just built, as the current stage and continues processing in a loop until the test is true. Once such a state is identified, a check is performed if there are other states that have a reliability value that is higher than the threshold. All such states are identified and the state that has the lowest cost is chosen and marked as the reference state. The cost of this state is J_{ref} and the reliability is R_{ref}.

Details of Computations at the Auxiliary Nodes: The main node allocates tasks to the auxiliary nodes. A task is nothing but a set of states from which new states for the next stage are to be built. After an auxiliary node receives the information, it begins to process the states sequentially.

For each state, it builds the base network represented by the state. Then it tests the decision set against the base network. It adds a unit link from the list of possible rights of way. Then it determines the reliability of the network and checks for the existence of states having same reliability but different costs and treats them accordingly as described in Sec. (4.1). After that it removes the unit link. This process is continued until all the rights of way have been tested.

When all the states are processed in the manner described above, the results are communicated back to the main node.

When the main node has identified a reference state, it moves into the second phase.

Phase II

The second stage is performed only by the main node because it is not as computationally intensive as the first phase. In this phase, all states from the last stage of Phase I satisfying the following criteria are chosen:

1. Cost $\leq J_{ref}$
2. Reliability $\leq R_{ref}$

These states are then collated into one stage which forms the first stage of the second phase. From this stage subsequent stages are built. However, in this phase, the building of stages along any trajectory ends if the cost of that trajectory is no longer less than J_{ref}. In other words, pursuing the trajectory any further will not offer any additional advantage over the reference state. In due course, all trajectories will hit a frontier or a boundary with a cost J_{ref}, and the final stage in this phase will not contain any states. The algorithm then terminates. During this procedure, all states that meet the threshold reliability criterion are identified. Among all such states, the state that has the lowest cost is the optimal state.

The configuration represented by the optimal state is the optimal solution.

5 Demonstration: Case Studies and Results

The algorithm was applied to a 22-bus system. This system was derived from the distribution network at Bus-2 of the RBTS [14]. From the layout given in the single line diagram of the above mentioned network, rights of way between nodes and costs thereof were established. For the sake of demonstration, the cost of interconnecting nodes i and j was assumed to be proportional to the distance as shown in the layout.

It was further assumed that the location and size of the generating units are known before hand. Typical values of forced outage rates were assumed as shown in Table 1.

The capacity of each unit-link was taken to be 0.2 MW, with an impedance of 0.006 p.u. per mile.

The program was executed with a setting of maximum $EPNS$ of 4% of the total demand which is equivalent to $EIR \geq 0.96$.

Two cases were studied: one, where the microgrid is built from scratch, and, two, where the microgrid is built from an existing network. In the first case, only the main buses, i.e., the load buses have been considered. The nodes arising from the points where feeder segments met were not used in the analysis. Those nodes were eliminated because the microgrid was supposed to be built from scratch and hence, from the analysis point of view, the distribution backbone did not exist and neither did the "intermediate" nodes. In the second case, these nodes were duly considered.

In both the cases a worst case scenario was analyzed where there is no assistance from the grid and the load is at the system peak. The cases are described in detail below.

5.1 *Building a Microgrid from Scratch*

The first case was to build a microgrid from scratch. For this, all the lines from the above network were removed. The loads were taken to be the peak loads at the respective buses. The generation and load data is provided below in Table 1 and Table 2 respectively.

Bus	Gen (MW)	FOR
1	5 .0	0.06
5	10.0	0.10
10	3.5	0.08
11	5.0	0.10
22	7.5	0.10

Table 1: Case A: Generation Data for the 22-Bus System

Bus	Load (MW)	Bus	Load (MW)	Bus	Load (MW)
1	0.8668	9	1.8721	16	0.7500
2	0.8668	10	0.8668	17	0.7291
3	0.8668	11	0.8668	18	0.7291
4	0.9167	12	0.7291	19	0.7291
5	0.9167	13	0.9167	20	0.9167
6	0.7500	14	0.9167	21	0.9167
7	0.7500	15	0.7500	22	0.7500
8	1.6279				

Table 2: Case A: Load Data for 22-Bus System

The resulting network is shown in Fig. 2. The network configuration is tabulated in Table 3.

LINE	Cap (MW)	LINE	Cap (MW)	LINE	Cap (MW)
1-18	0.8	11-12	0.8	21-22	1.0
1-2	1.0	5-6	0.8	10-16	0.6
4-5	0.8	5-7	0.8	8-11	1.6
15-22	0.8	1-19	0.8	10-17	0.6
3-5	0.2	20-22	0.8	17-22	0.4
1-3	0.8	1-16	0.4	1-17	0.4
1-4	0.2	10-14	0.8	11-16	0.4
10-13	0.8	2-5	0.2	16-22	0.2
5-9	1.8	14-22	0.2	11-13	0.2
5-16	0.2	11-14	0.2	11-17	0.4
1-20	0.2	9-11	0.2	5-8	0.2
5-11	0.6				

Table 3: Resulting network configuration for 22-Bus System

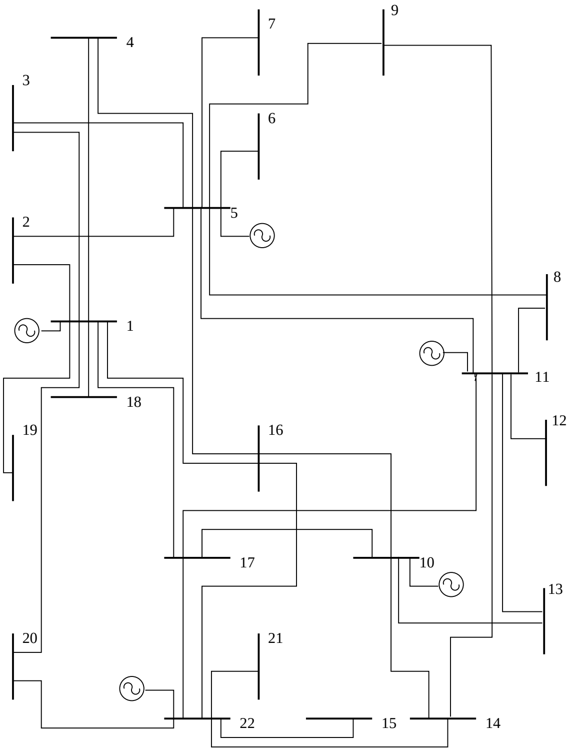

Figure 2: Case A: Resulting Microgrid of the 22-Bus System

5.2 *Expanding an Existing Distribution Network into a Microgrid*

The analysis consisted of growing a microgrid from an existing distribution network. The original backbone network was retained and used as a starting point for determining the architecture of the microgrid. The line capacities were assumed to be at 150% of their base loading.

For the analysis, as an example, a load growth of 100% was assumed. So the new loads at the buses are twice the values shown in Table 2. Correspondingly the generation capacities of the generators were increased. The forced outage rates were not changed. The new generating capacities are tabulated in Table 4. Again, the reliability criterion of $EIR \geq 0.96$ was used.

Bus	Gen (MW)
1	7.5
5	20.0
10	10.0
11	10.0
22	15.0

Table 4: Case B: Generation Data for the 22-Bus System

The resulting microgrid is shown in Fig. 3. In this figure, the original backbone network is shown in straight lines, while the lines added by the algorithm are shown as arcs. The new lines that were added and their capacities are tabulated in Table 5.

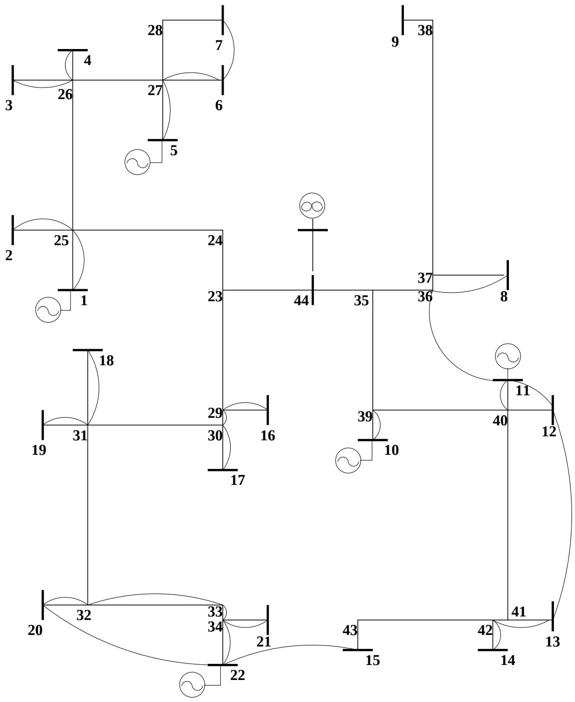

Figure 3: Case B: Resulting Microgrid of the 22-Bus System

LINE	Cap (MW)	LINE	Cap (MW)	LINE	Cap (MW)
29-30	2.0	33-34	1.2	26-4	0.6
39-10	7.0	42-14	0.6	29-16	0.4
30-17	0.4	31-19	0.4	32-20	0.4
34-21	0.6	34-22	3.4	41-13	0.2
25-1	4.4	25-2	0.4	26-3	0.4
27-5	5.4	27-6	0.8	40-11	0.3
6-7	0.4	31-18	0.4	11-12	0.8
11-36	5.4	36-8	0.8	32-33	0.2
22-15	2.4	20-22	1.8	13-12	0.4

Table 5: Resulting Microgrid from an Existing Network

5.3 *Discussion of Results*

The two cases presented here show very different characteristics. We find that the microgrid created from scratch is a heavily meshed one. The one that is developed from an existing backbone has relatively few loop closing lines. This result is not very surprising. The line capacities in a distribution network diminish as we go farther away from the substation. This means that the feeder sections nearer to the substations have very high capacities as compared to the ones near the load buses. In the distribution network presented here, Bus 44 is connected to a substation. Therefore the lines immediately connected to this bus have very high capacities, the sum of the capacities is comparable to the total system load. Now, when we use distributed generation, they are placed at or near the load buses. This effectively reverses the direction of power flow. So in this scenario, the lines near the generators are at a deficit in capacity, while those farther away have surplus capacities, sometimes to the extent of being underutilized. An intuitive approach would be to strengthen the segments near the generators, and try to utilize the available "upstream" transmission capacity as much as possible.

The algorithm recognizes this and takes advantage of the existing transmission capacities. Therefore, we find very few loop closing lines.

6 Conclusions

This paper presented a rational approach with an explicit reliability constraint to designing cost-optimized microgrid architectures.

A dynamic programming method was developed to determine the optimal interconnections between microsources and load points, given their locations and rights of way for possible interconnections. A scheme was implemented to render the size of the problem manageable and to minimize the storage requirements. A new approach, called "unit link addition," was demonstrated on a 22-bus system.

In this work, we demonstrated the applicability of the method by first building a microgrid from scratch, and then by building a microgrid from an existing distribution backbone. This method would indeed be a more rational approach to enabling the evolution of a distribution network than *ad hoc* expansion in response to load growth.

The work reported here represents the authors' first steps in the direction of developing a methodology for rational design of microgrids. This is also an initial step towards fulfilling an immediate need for system planning tools that take into account the changes going on in the industry.

Acknowledgments

The work reported in this paper was supported in part by the National Science Foundation, under grants ECS-0134598 and ECS-0140092, and by the Electric Utility Management Program. This support is gratefully acknowledged. The authors benefited much from discussions with Prof. A. Rangan of the Department of Mathematics, Indian Institute of Technology, Chennai, and express their gratitude to him. The authors are also grateful to the Klipsch School of Electrical and Computer Engineering and the Department of Computer Science, New Mexico State University, for granting access to their high performance computing facilities. Thanks are also due to Kjell Eikland and Peter Notebaert, developers of the open source Linear Programming package used in this work.

REFERENCES

[1] G. Celli and F. Pilo, "Optimal Distributed Generation Allocation in MV Distribution Networks," in *Proc. 22nd IEEE PES Int'l Conf. Power Industry Computer Appl. PICA 2001*, 20-24 May 2001, Sydney, Australia, pp. 81-86.

[2] A. R. Wallace and G. P. Harrison, "Planning for Optimal Accommodation of Dispersed Generation in Distribution Networks," in *Proc. 17th Int'l Conf. Electricity Distribution*, CIRED 2003, 12-15 May 2003, Barcelona, Spain.

[3] N. S. Rau and Y. H. Wan, "Optimum Location of Resources in Distributed Planning," *IEEE Trans. Power Systems*, vol. 9, no. 4, pp. 2014-2020, Nov. 1994.

[4] G. Latorre, R. D. Cruz, J. M. Areiza and A. Villegas, "Classification of Publications and Models on Transmission Expansion Planning," *IEEE Trans. Power Systems*, vol. 18, no. 2, pp. 938-946, May 2003.

[5] R. Lasseter and P. Piagi, *Industrial Application of Microgrids*, CERTS, Oct. 2001.

[6] R. Lasseter, et al, *Integration of Distributed Energy Resources: The CERTS MicroGrid Concept*, CERTS, Apr. 2002.

[7] C. Marnay and O. Bailey, *The CERTS Microgrid and the Future of the Macrogrid*, CERTS, Aug. 2004.

[8] R. H. Lasseter and P. Piagi, *Microgrid: A Conceptual Solution*, CERTS, June 2004.

[9] U. S. Department of Energy, *Strategic Plan for Distributed Energy Resources*, Sept. 2000.

[10] U. S. Department of Energy, *Transmission Reliability Multi-Year Program Plan FY2001-2005*, July 2001.

[11] Y. P. Dusonchet and A. H. El-Abiad, "Transmission Planning Using Discrete Dynamic Programming," *IEEE Trans. Power Apparatus and Systems*, vol. 92, pp. 1358-1371, July 1973.

[12] N. G. Boulaxis and M. P. Papadopoulos, "Optimal Feeder Routing in Distribution System Planning Using Dynamic Programming Technique and GIS Facilities," *IEEE Trans. Power Delivery*, vol. 17, no. 1, pp. 242-247, Jan. 2002.

[13] C. Singh and J. Mitra, "Composite System Reliability Evaluation Using State Space Pruning," *IEEE Trans. Power Systems*, vol. 12, no. 1, pp. 471-479, Feb. 1997.

[14] R. N. Allan, R. Billinton, I. Sjarief, L. Goel and K. S. So, "A Reliability Test System for Educational Purposes — Basic Distribution System Data and Results," *IEEE Trans. Power Systems*, vol. 6, no. 2, May 1991.