CHAPTER 3

THE MULTI-CRITERIA ELECTRIC POWER DISTRICTING PROBLEM

INTRODUCTION

Single criteria optimization methods often fail to capture the complexity of problems faced by decision makers (DMs) in today's rapidly changing business environment. The mathematical foundation for multi-criteria decision making (MCDM) was developed over a century ago (Pareto 1896). Characteristic of a typical MCDM problem is the absence of a unique global optimum. Rather, multiple solutions to the problem often exist that are superior to (dominate) the others in the solution space. These solutions are known as Pareto optimal solutions. This chapter considers the EPDP as a MCDM problem.

MULTI-CRITERIA DECISION MAKING CONCEPTS

Many real world problems involve a variety of objectives, often conflicting in nature. While it is possible to aggregate multiple objectives into a single weighted objective, many problems are better posed as MCDM problems. The MCDM problem can be expressed as follows:

Minimize (or Maximize): $F(x) = (f_1(x), f_2(x), \ldots, f_Q(x))$ (3.1)

Subject to: $A(x) \leq b$ (3.2)

When solving MCDM, it is generally assumed that decision makers prefer or desire to obtain Pareto optimal or non-dominated solutions. We define these concepts as follows:

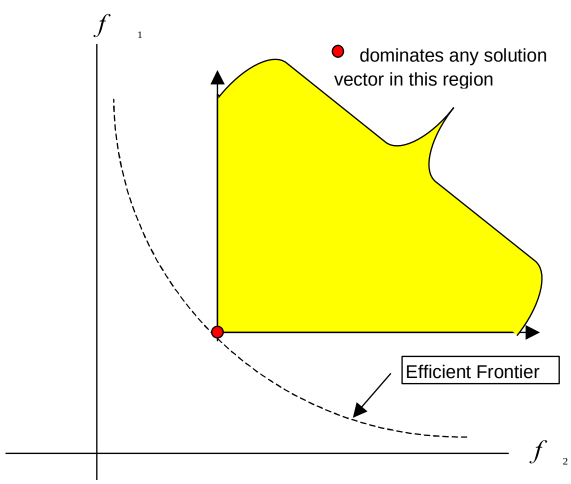
> *Definition 1 - Pareto Dominance.* *A solution vector $x^* \in R^n$ is said to dominate $x \in R^n$ if $F(x^*)$ is at least as good as $F(x)$ in all of its components and better in at least one component. For a Minimization problem $f_i(x^*) \leq f_i(x)$ for all i and $f_i(x^*) \neq f_i(x)$ for at least one i. For a Maximization problem $f_i(x^*) \geq f_i(x)$ for all i and $f_i(x^*) \neq f_i(x)$ for at least one i. See Figure 3.1.*

> *Definition 2 - Pareto Optimal Solutions.* *A solution vector $x^* \in R^n$ is a Pareto optimal solution if there exists no other solution vector $x \in R^n$ for which x dominates x^*. Such solutions are also referred to as non-dominated, efficient, or non-inferior. See Figure 3.2.*

According to Definition 1 and 2, a solution x^* is Pareto optimal if no other solution exists that dominates x^*. From a practical perspective, when using heuristic population based search techniques, it if often difficult or impossible to know if another (yet to be observed) solution exists that dominates a particular known observed solution. As a result, in this work we relax the definition of Pareto optimality to refer to known observed solution vectors rather than to all possible solution vectors that theoretically exist in R^n. That is, for a given population of solution vectors, we say that x^* is (currently) non-dominated if there exists no other solution x in the population for which x dominates x^*. Similar relaxed definitions of Pareto optimality have been used by (Goldberg 1989, Horn and Nafpliotis 1993, Fonseca and Flemming 1995).

FIGURE 3.1

Dominance for Minimization of f_1 & f_2



Dominance for Maximization of f_1 & f_2

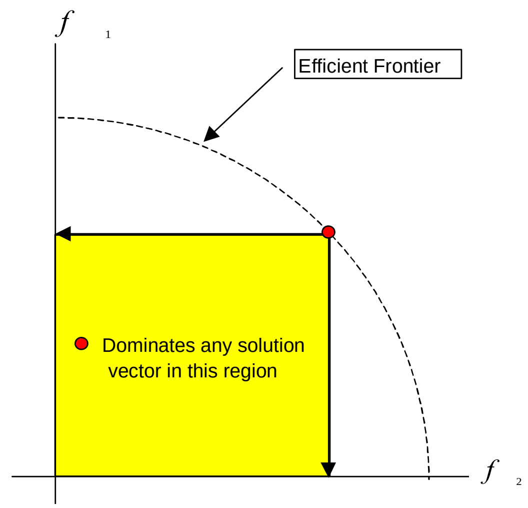
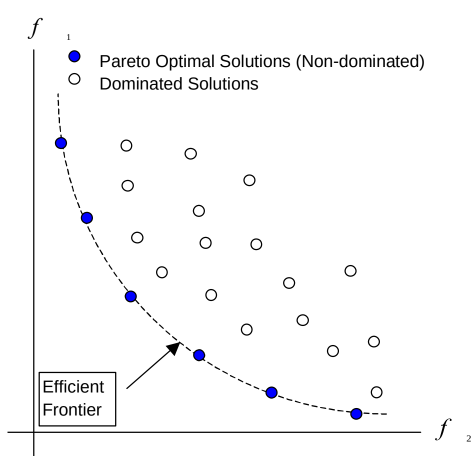
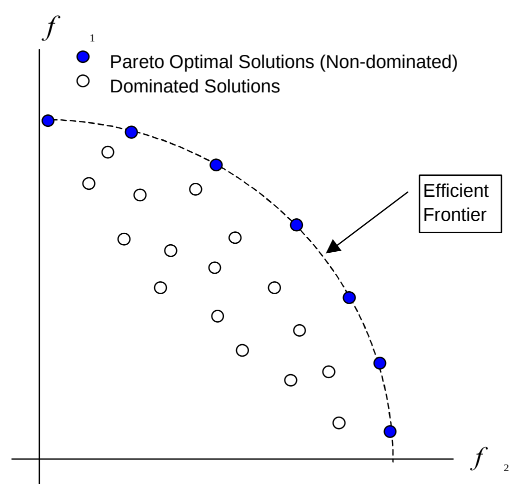


FIGURE 3.2

Pareto Set for Minimization of f_1 & f_2



Pareto Set for Maximization of f_1 & f_2



A MULTI-CRITERIA ELECTRICAL POWER DISTRICTING MODEL

The first multi-criteria redistricting model was proposed by (Zoltners 1979) for the sales territory alignment problem. The objective was to balance the relative merits of multiple territory attributes. The objective function used in the model was a weighted sum of the various criteria to be considered, which essentially reduced the problem back to a single objective and applied a single criteria optimization method.

We propose the model below for simultaneously minimizing the $Q=2$ components of $F(x)$ for the EPDP. Recall from chapter 2 that f_1 is defined as the total revenue deviation of a districting plan and f_2 is defined as the total Euclidean distance (compactness) of a districting plan. An objective function that seeks to minimize multiple criteria (attributes) for the EPDP problem can be expressed as follows:

Minimize: $$F(x) = (f_1(x), f_2(x)) \tag{3.3}$$

Where: $$f_1(x) = \sum_{j=1}^{k} \left| R_j - \overline{r} \right| \tag{3.4}$$

$$f_2(x) = \sum_{j=1}^{k} K_j \tag{3.5}$$

$$\overline{r} = \frac{1}{k} \sum_{i=1}^{n} r_i \tag{3.6}$$

$$R_j = \sum_{i=1}^{n} r_i X_{ij} \qquad , \; j=1 \ldots k \tag{3.7}$$

$$\left| R_j - \overline{r} \right| \leq \delta \; \overline{r} \qquad , \; j=1 \ldots k \tag{3.8}$$

$$K_j = \sum_{i=1}^{n} \left((C_x(j) - N_x(i))^2 + (C_y(j) - N_y(i))^2 \right)^{1/2} X_{ij} \qquad , \; j=1 \ldots k \tag{3.9}$$

$$C_x(j) = \frac{1}{m_j} \sum_{i=1}^{n} N_x(i) X_{ij} \qquad , \; j=1 \ldots k \tag{3.10}$$

$$C_y(j) = \frac{1}{m_j} \sum_{i=1}^{n} N_y(i) X_{ij} \qquad , \; j=1 \ldots k \tag{3.11}$$

$$m_j = \sum_{i=1}^{n} X_{ij} \qquad , \; j=1 \ldots k \tag{3.12}$$

$$X_{ij} = \begin{cases} 1, x_i = j \\ 0, o.w. \end{cases} \qquad , \; i=1 \ldots n \qquad , \; j=1 \ldots k \tag{3.13}$$

$x = (x_1 , x_2 , \ldots, x_n)$
x_i= *district assignment for node i*
k = *number of districts (partitions)*
R_j = *sum total of revenue potential in district j*
r_i = *amount of revenue potential contained in node i*
n = *number of nodes in the network*
m_j = *number of nodes in district j*
$100\delta, (0 \leq \delta_i \leq 1)$, *is the maximum allowable percentage deviation of the actual attribute in a district from the target*
K_j =*Compactness of district j*

$C_x(j)$= X-Coordinate for the Centroid of district j
$C_y(j)$= Y-Coordinate for the Centroid of district j
N_x (i)= X-Coordinate for node i
N_y (i)= Y-Coordinate for node i

EVOLUTIONARY ALGORITHMS AND MCDM

Evolutionary algorithms (EAs) represent a powerful, general purpose optimization paradigm where the computational process mimics Darwin's theory of biological evolution. The popular components of EAs include Genetic Algorithms (GAs) (Holland 1975), Evolution Strategies (Rechenberg 1973), and Genetic Programming (GP) (Koza 1992).

In a nutshell, most EAs start with a set of chromosomes (numeric vectors) representing possible solutions to a problem. The individual components (numeric values) within a chromosome are referred to as genes. New chromosomes are created by crossover (the probabilistic exchange of values between vectors) or mutation (the random alteration of values within a vector). Chromosomes are then evaluated according to a fitness (or objective) function with the fittest surviving into the next generation. The result is a gene pool that evolves over time to produce better and better solutions to a problem.

The notion of a non-dominated solution set is particularly suitable to a population based search strategy. By exploiting the characteristics of the currently non-dominated solutions in a population, better solutions often eventually emerge that dominate the currently non-dominated solution set. Applying EAs to MCDM was motivated by their effectiveness in locating multiple non-dominated solutions in a single optimization run. The seminal work in this area was accomplished using a GA by (Schaffer 1986) and using an ES by (Kursawe 1991).

Schaffer's Vector Evaluated Genetic Algorithm (VEGA) was centered on multiple populations evolving using a separate fitness function (criteria) for each population. In each generation, offspring are produced by applying genetic operators (crossover & mutation) that merge the populations of chromosomes. The populations are monitored during the evolutionary process for non-dominated solutions. VEGA has been shown to split into population species when the trade-off surface is concave (chromosomes strong in only one objective), forming clusters of solutions near the extremum of each objective. For a detailed description of VEGA the reader is referred to (Schaffer 1986). For an excellent general overview of evolutionary approaches to the MCDM problem see (Fonseca and Flemming 1995).

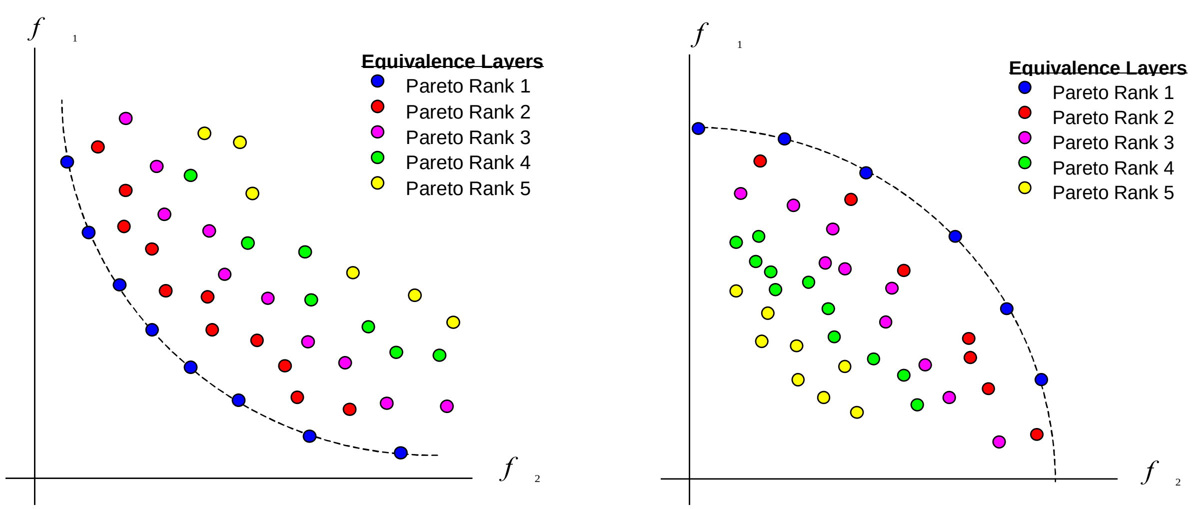
A Pareto ranking technique was subsequently proposed in the Pareto Genetic Algorithm (PGA) (Goldberg 1989). Goldberg's method proved to be effective on non-convex trade-off surfaces that present difficulties to some other techniques and is described in detail in the following section. Other EA based methods for the MCDM problem that have been developed recently include: niched Pareto GA (Horn and Nafpliotis 1993), non-dominated sorting GA (Srinivas and Deb 1995), multi-sexual GA (Lis and Eiben, 1997), and the first hybrid multi-objective GA (Ishibuchi and Murata, 1998).

Pareto Ranking

An effective form of selection which considers the notion of a non-dominated solution set of is known as Pareto ranking and selection (Goldberg 1989). The Pareto ranking and selection technique consists of ranking the population members in layers according to our relaxed notion of dominance. Strictly speaking, a solution vector $x^* \in R^n$ is said to dominate $x \in R^n$ *if* $F(x^*)$ is at least as good as $F(x)$ in all components and better than $F(x)$ in at least one component. The Pareto ranking technique relaxes the definition of a dominating solution vector by limiting the scope of R^n to the members of the current population. Thus, solution vectors that are non-dominated in the current population are given a rank of one and an equal probability of reproducing. Then, the non-dominated solutions are removed from the current population to expose a second layer of previously dominated solutions that have now become non-dominated. These solutions are given a rank of two and an equal but lower probability of reproducing than those of higher rank. The different layers of non-dominated solutions identified by the Pareto ranking technique are also sometimes referred to as equivalence class layers. The process continues until the entire population of solution vectors is ranked. By recombining non-dominated solution vectors in the current population, new solutions are likely to emerge in later generations that dominate their predecessors. The result is an evolving gene pool that is compelled toward the efficient frontier. See Figure 3.3.

FIGURE 3.3

Pareto Ranking for Minimization of f_1 & f_2 — Pareto Ranking for Maximization of f_1 & f_2



A HYBRID ALGORITHM FOR THE MULTI-CRITERIA EPDP

In this section, we introduce a hybrid algorithm to solve the multi-criteria formulation of the EPDP provided in (3.3). Our hybrid approach is similar to the first hybrid multi-criteria GA proposed by (Ishibuchi and Murata 1998). However, our procedure has several features that depart from the method described in (Ishibuchi and Murata 1998). First, the hybrid component in their algorithm implemented a deterministic local search procedure for improving solutions. In contrast, our algorithm performs a stochastic local search. Second, their directed search component reassigned random weights to solution vectors in each generation. We also reassign weights in each generation, however, the weights are evenly distributed about the multi-criteria solution space and are assigned based upon the relative location of the solution vectors in the solution space. Third, their technique does not make use of the Pareto dominance concept directly, while ours implements Pareto ranking and selection. Finally, our genetic representation departs from the traditional GA model to enforce connectivity while partitioning a graph into contiguous districts.

Simulated Annealing Genetic Algorithm (SAGA)

We refer to our hybrid multi-criteria algorithm as the Simulated Annealing Genetic Algorithm (SAGA) because we employ a composite concept of a population based evolutionary search and a point based local search similar to simulated annealing. The crossover operator used in SAGA is identical to the one described for the single criteria GA in chapter 2. Recall that the purpose of the crossover operator is to rapidly explore large search spaces and effectively reduce the search to regions of greater promise. The shortcoming of traditional genetic operators is that they lack the "killer instinct," meaning a solution vector can be very close to a better solution yet is unable to locate it using crossover or random mutation (Ross 1997). This is the motivation for combining GAs with local search operators (Hohn and Reeves 1995). Such hybrid algorithms are sometimes referred to as Memetic Algorithms (Radcliff and Surry 1994).

The selection operator that we implement in SAGA is the Pareto ranking and selection method described above. In addition, we invoke another algorithm during Pareto ranking which detects and eliminates duplicate solutions in each Pareto Layer. This enhanced feature provides two desirable characteristics to our hybrid algorithm. First, it efficiently utilizes storage space in the population of solution vectors that are retained. Second, it helps to maintain diversity in the population because duplicate solutions never survive. It is significant to note that the check for duplicate solutions is computationally efficient because it is invoked only when all of the function values for a pair of solution vectors is the same. During Pareto ranking and selection it is necessary to compare the function values of each solution vector. Thus, the added computational effort required to detect and eliminate duplicate solutions is marginal.

Selection takes place in SAGA by first combining the parent and child populations and then Pareto ranking the combined population. The solution vectors are then reordered according to their assigned Pareto rank into equivalence class layers. The combined population is then split in half and only the top half is retained to become the new parent population. Because of the transitivity property of the Pareto dominance relation, non-dominated solution vectors are always retained in the current population (provided the size of the non-dominated set does not exceed the size of the parent population). This characteristic is referred to as an "Elitist" selection strategy, where the best solution(s) is always guaranteed to survive (although one may wish to call this "Pareto Elitist"). For the test problems studied herein (both Ghana's EPDP and the randomly generated test problems) there were no instances where the size of the non-dominated set exceeded the population size.

Recall that the mutation operator we employed in the single criteria GA was based upon an enumerative local search for an improving solution for a single objective function. In this sense, it too was a hybrid (memetic) algorithm. However, the local search in SAGA is not an enumerative hill climbing operator. Rather, it is capable of uphill (downhill) moves in a minimization (maximization) problem. It is designed to mimic the annealing process of the SA procedure and thus, will accept inferior solutions when the temperature of the system is sufficiently high and/or the measured change in energy is sufficiently low. To preclude our local search procedure from consuming all of the computational effort we invoke certain limitations that restrict its operation. For example, we limit the standard SA control parameters, setting the initial system temperature to 0.2^o rather than 5.0^o and the cooling factor to 0.8 rather than 0.95. The latter figures were the settings used in our empirical study for chapter 2. Furthermore, we also add a control parameter which limits the local search to a fixed number of moves. We call this parameter *M* for Maximum Iterations. Thus, for obvious reasons we refer to our pseudo SA procedure as a Rapid Cooling (RC) operator. Rapid cooling takes place immediately following crossover on each solution vector in the population. The following procedure describes the RC operator in SAGA:

SAGA Rapid Cooling Procedure (Pass In: Child in state s = $F(x_s)$)

Set T= 0.2 (initial temperature)
Set T_{stop}=0.1 (final temperature)
Set d = 0.8 (cooling factor)
Set M = 25 (Max Iterations)

Do

b = 0 ('do loop' counter)
Create neighbor in state t consisting of k instances of $G'(N', E') \subseteq G(N, E)$
Evaluate Child =$F(x_t)$

If $F(x_t)$ Dominates $F(x_s)$

Set s = t
Exit Sub

Else
Randomly Select f(x) from F(x)
Evaluate t
$\Delta E = E(t) - E(s)$

If $\Delta E < 0$ then

Set s = t

Else if $e^{-\frac{\Delta E}{T}}$ > Random [0,1] Then

Set s = t

End if

End if

b=b+1
*T=d*T*

Loop Until $T \leq T_{stop}$ *or b>M*

End Procedure

<u>Parallel Simulated Annealing (PSA)</u>

In addition to our hybrid algorithm we introduce a population based simulated annealing algorithm which uses constant weights. The weights are evenly distributed about the multi-criteria search space among the solution vectors in the population. We refer to this algorithm as Parallel Simulated Annealing (PSA) because the solution vectors are annealed in parallel rather than in series as would be typical of the classical single point SA procedure. A variant of population based SA was explored in (Goldberg 1990). The underlying representation of the solution vectors in SAGA and PSA are identical, meaning we use a chromosome representation for the solution vectors in both procedures. Thus, any performance difference between the procedures can be attributed to the effectiveness of each algorithm on the test problems under study. The only change required in the GA object model (in chapter 2) is the addition of a temperature property in the chromosome object so that each solution vector independently tracks its own temperature. In a sense, PSA can be considered a GA that uses SA as a mutation step, and does not recombine members of the population.

EMPIRICAL STUDY WITH MULTIPLE CRITERIA – PART I

The purpose of Part –1 of this empirical study is to extend the analysis performed in chapter 2 to the multi-criteria version of the EPDP using Ghana's current power transmission configuration. To this end, we examine and compare the performance of SAGA and PSA in Part-1 of this empirical study. We are also motivated to understand the robustness and sensitivity of our solution techniques. In Part II of this empirical study, we depart from the Ghana model and examine the performance of SAGA and PSA with a random test problem generator specifically designed for the EPDP.

At the conclusion of chapter 2, it appeared that the GA was more suitable for our needs in the DSS. However, neither procedure dominated the other with regard to our performance measures. Given that the EPDP is better posed as a multi-criteria problem, we seek to understand the behavior of these solution techniques in the context of the multi-criteria problem formulation. Indeed, both the GA and SA have redeeming qualities that can be combined into a composite search technique for this purpose.

We now wish to simultaneously minimize the total revenue deviation function f_1 provided in (3.4) and the total Euclidean distance function f_2 provided in (3.5). Recall that we have identified the range for the number of districts that we wish to study as $4 \le k \le 6$. A population size of 25 solution vectors was used and each procedure was allowed to run for exactly 10,000 function evaluations. Similar benchmarks for empirical studies of hybrid GAs were established by (Ishibuchi and Murata 1998).

Non-dominated Solution Set

We are no longer concerned with finding a single best solution, but wish to find a set of solutions on or near the efficient frontier in a single optimization run. This is the reason why we have limited our study to the examination of population based approaches. To measure the quality of the final solution set, we analyze the difference between the average performance of SAGA and PSA over 30 independent optimization runs.

We can offer no guarantee of strict Pareto optimality of the final non-dominated set since we are using a relaxed definition of Pareto dominance. However, it is interesting to note that for the case of k=4 districts and k=5 districts, none of the final solutions that were deemed to be Pareto optimal for any single optimization run, were later dominated by a previous or subsequent optimization run. For the case of k=6 districts, it turns out that we uncovered a single occasion for both SAGA and PSA where one such scenario occurred. This situation is referred to as false domination and described in detail in (Horn and Nafpliotis 1993). The fact that false domination was rarely experienced throughout our performance testing provides us with high confidence in the quality of the non-dominated set. Figures 3.4 – 3.6 show the non-dominated sets (with Pareto rank of 1) produced by SAGA and PSA for Ghana's current transmission network configuration.

FIGURE 3.4
SAGA vs. PSA Non-dominated Solutions – 4 Districts

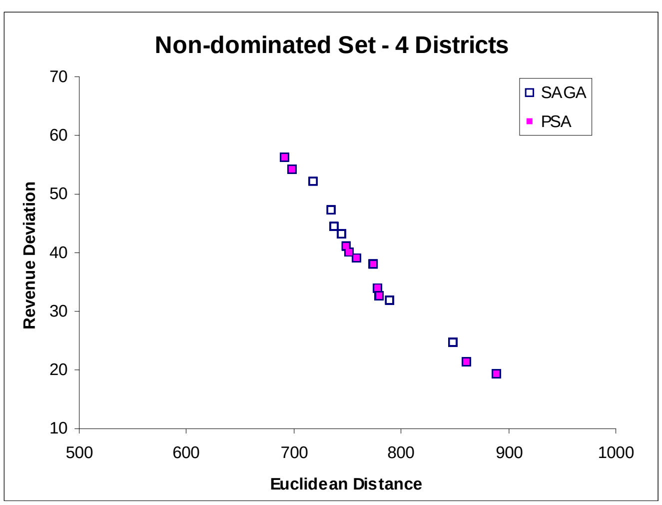


FIGURE 3.5
SAGA vs. PSA Non-dominated Solutions – 5 Districts

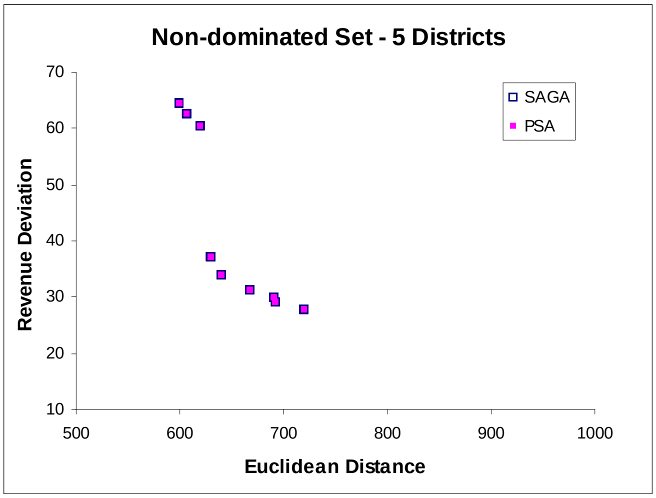
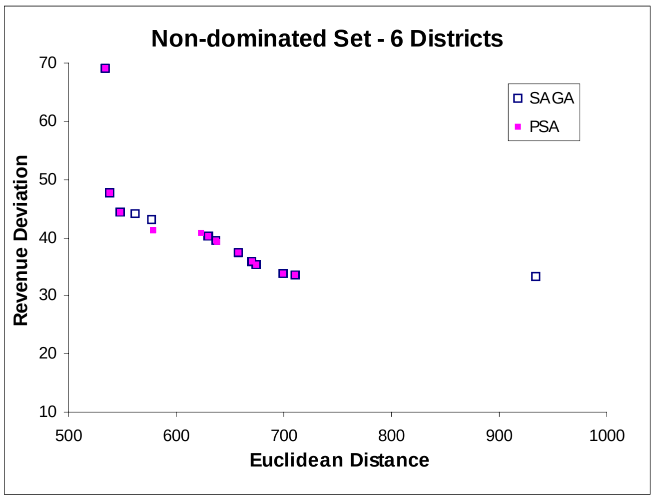


FIGURE 3.6
SAGA vs. PSA Non-dominated Solutions – 6 Districts



It is interesting to note that for *k*=4 districts shown in Figure 3.4, SAGA provided better overall coverage of the efficient frontier than PSA. SAGA produced every non-dominated solution found by PSA and in addition, was able to locate 6 more. For *k*=5 districts shown in Figure 3.5, the coverage of the efficient frontier appears to be identical with each technique finding the exact same non-dominated set. For *k*=6 districts shown in Figure 3.6, each procedure located 10 identical solutions on the efficient frontier. However, both SAGA and PSA uncovered 3 additional solutions not found by the other. It is significant to note that the observations pointed out above do not consider the frequency of occurance of each non-dominated solution experience in a single run. Rather, results in Figures 3.4-3.6 were aggregated across 30 optimization runs. The aggregated solution set provides substantial value in confirming the validity of non-dominated solutions produced in an single optimization run. Table 3.1 shows the number of non-dominated solutions produced in each of the 30 runs.

TABLE 3.1
Number of Non-dominated Solutions per Run for SAGA and PSA

Number of Non-dominated Solutions						
Districts	4		5		6	
Run	SAGA	PSA	SAGA	PSA	SAGA	PSA
1	8	4	5	5	3	2
2	10	3	8	4	1	5
3	12	4	3	2	1	3
4	11	6	7	2	3	2
5	11	3	6	5	0	2
6	9	5	6	2	3	3
7	9	4	6	3	2	3
8	9	4	6	4	4	4
9	8	5	5	3	5	2
10	8	4	6	3	1	4
11	8	4	8	3	4	2
12	8	4	4	2	4	4
13	11	2	5	5	2	2
14	10	4	5	4	4	3
15	8	7	4	4	2	3
16	10	4	5	1	2	2
17	11	5	4	5	3	4
18	10	4	5	4	3	1
19	10	3	5	4	2	2
20	9	6	7	2	2	4
21	11	7	5	3	4	7
22	12	4	7	5	1	2
23	7	7	8	5	2	3
24	10	6	5	5	4	4
25	13	6	6	4	3	2
26	7	6	4	3	3	2
27	10	5	4	2	4	5
28	10	5	5	5	2	3
29	9	3	5	3	2	2
30	7	5	5	3	2	4
Average	**9.533**	**4.633**	**5.467**	**3.5**	**2.6**	**3.033**

Table 3.2 shows a statistically significant difference in the average number of non-dominated solutions produced by SAGA and PSA for k=4 and k=5 districts. However, when k=6 districts the difference in the average number of non-dominated solutions is not significant. We point out that a similar behavior (trend) was revealed in the empirical studies in chapter 2 when optimizing f_2 independently.

TABLE 3.2
SAGA vs. PSA Test Results

Districts	t-value
4	13.169*
5	6.15*
6	-1.361

*t-values significant at α=.005

Equivalence Class Layers

We believe that the number of non-dominated solutions is an important measure to study because it represents an algorithm's ability to provide coverage on the efficient frontier (assuming that an equal amount of computing resources have been expended). Thus, a large number of non-dominated solutions would be preferred to a smaller number. However, our DSS is based upon the notion of a "soft frontier" of dominated solutions located in the vicinity of the efficient frontier. These solutions are organized and rendered as equivalence class layers by our DSS for the DM to consider. Thus, the number of equivalence class layers in the final population of solution vectors is also of interest to us.

We use the number of equivalence class layers as a secondary measure of performance in our empirical study to judge the quality of the soft frontier. In this case, fewer equivalence class layers would be preferred to more because it would represent a tighter overall fit of the dominated solutions (Pareto rank >1) to the non-dominated solution set (Pareto rank = 1) assuming equal population sizes. By this we mean that overall, the final population of solution vectors are located closer to the efficient frontier.

Figure 3.7 shows a single run of SAGA with the final population of solution vectors resulting in 4 equivalence class layers. Also shown is the current configuration of Ghana's transmission network with a specific districting plan (k=5 districts) rendered in the network. Each BSP in the network is assigned a color-coding to identify its district assignment. Each equivalence class layer corresponds to a series shown in the legend of the scatter plot. Notice that one solution vector in the non-dominated set (first series) is highlighted with a large red dot. This solution vector corresponds to the districting plan that is rendered in the network. Relative to all other solutions, it has the minimum total Euclidean distance (f_1=64.48, f_2 =600.56).

FIGURE 3.7
Final Population produced by SAGA after a Typical Run

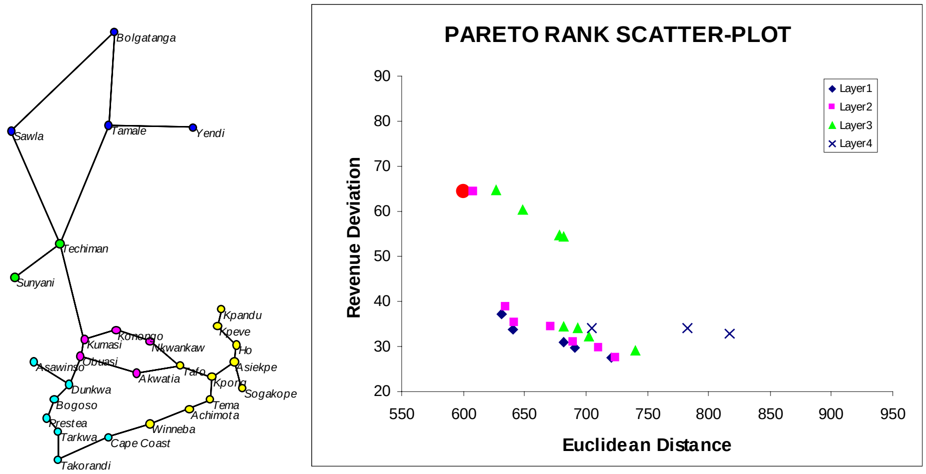


Figure 3.8 shows a single run of PSA with the final population of solution vectors resulting in 7 equivalence class layers. The highlighted solution vector (red dot) corresponding to the districting plan shown in Ghana's transmission network has the minimum total revenue deviation (f_1=27.58, f_2 =720.69) relative to all other solutions.

FIGURE 3.8
Final Population produced by PSA after a Typical Run

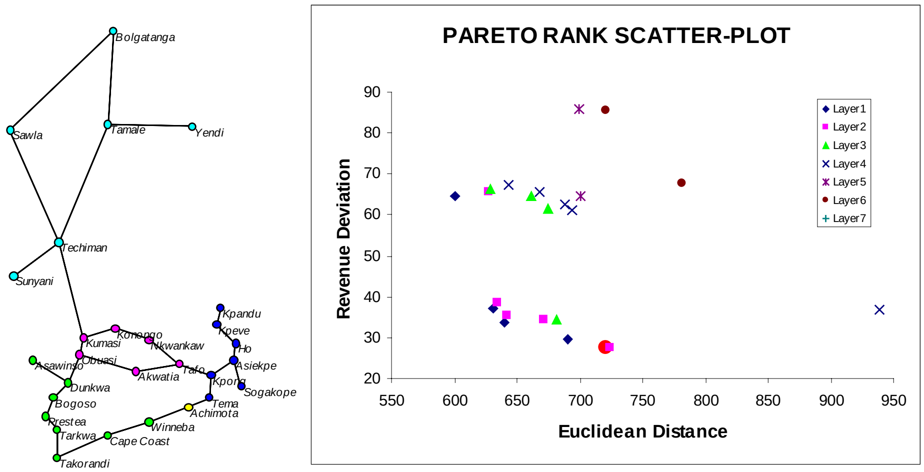


Table 3.3 shows the number of equivalence class layers produced for each of the 30 optimization runs.

TABLE 3.3
Number of Equivalence Class Layers per Run for SAGA and PSA

Number of Equivalence Class Layers						
Districts	4		5		6	
Run	SAGA	PSA	SAGA	PSA	SAGA	PSA
1	4	4	4	7	3	7
2	5	6	3	5	3	5

3	3	6	4	5	3	6
4	3	5	4	5	4	7
5	3	6	4	6	3	5
6	4	6	4	5	3	6
7	4	5	4	6	3	5
8	4	5	4	6	4	8
9	4	7	4	6	3	6
10	3	7	4	7	3	7
11	4	7	4	6	3	7
12	4	5	4	6	4	7
13	3	5	3	5	3	6
14	3	6	3	7	3	4
15	3	5	3	5	3	7
16	3	5	4	8	3	7
17	4	6	4	6	3	5
18	3	8	3	6	4	7
19	3	5	4	5	3	7
20	4	5	3	6	3	7
21	3	7	4	4	3	6
22	3	6	3	5	3	7
23	4	6	4	6	3	6
24	4	7	4	6	4	6
25	3	6	4	6	3	4
26	5	6	3	6	4	6
27	4	5	3	6	3	8
28	3	7	3	5	4	5
29	4	5	4	5	3	6
30	4	5	4	6	3	7
Average	**3.6**	**5.8**	**3.667**	**5.767**	**3.233**	**6.233**

Table 3.4 shows the results of the test for statistically significant differences in the average number of equivalence class layers produced by SAGA and PSA. We point out that the average performance difference of SAGA and PSA is statistically significant in all cases.

TABLE 3.4

SAGA vs. PSA Test Results

Districts	t-value
4	-10.815*
5	-12.139*
6	-14.599*

* t-values significant at α=.005

Reliability

A most interesting and unexpected result that we observed was an increase in the frequency in which the extremum of each (independent) function was located during the empirical studies in this section. Recall from chapter 2, we reported in Tables 2.9 and 2.10 the number of times the best solution was repeated by the GA and SA for f_1 and f_2 optimized independently. Tables 3.5 and 3.6 show this data for f_1 and f_2, respectively, optimized simultaneously. The improvement on f_2 with respect to this performance measure is quite remarkable.

One possible explanation for this result would be that the most reliable path to the best solution for f_2 is not necessarily the most direct path (i.e., weight of f_1 =0, weight of f_2 =1). The data supports this position in that on many occasions the best solution was located by a solution vector not weighted as shown above. Another possible explanation is that by using an aggregate weighted objective function, the flat spots that occur in f_1 may balance out the gradient of f_2. We believe that this interesting observation is worthy of future investigation. It is also noteworthy to mention that a new best solution was located for f_2 when k=6 districts as shown in Table 3.6.

TABLE 3.5

Revenue Deviation ($ Millions)

Revenue Deviation	Best Solution	Frequency of Best Solution	
Districts	Value	SAGA	PSA
4	19.220	30	19
5	27.583	30	23
6	33.158	2	4

TABLE 3.6

Compactness (Euclidean Distance)

Compactness	Best Solution	Frequency of Best Solution	
Districts	Value	SAGA	PSA
4	691.903	30	23
5	600.561	29	23
6	534.04*	3	8

* New best solution value found

Summary - Part I

To summarize the results in Part 1of this empirical study, we return to our central story. It is our objective throughout our empirical studies to match the relevant performance characteristics of the algorithms (SAGA and PSA) to the needs of our DSS. The story began with an investigation of two well-known general-purpose stochastic algorithms (GA and SA) in chapter 2. We found that both had redeeming qualities that warranted further investigation.

Population based search techniques provide a significant advantage over point based solution methods when confronted with a multi-criteria problem. The EPDP is best posed as a multi-criteria problem. Thus, we developed a composite algorithm that would exploit the features in each procedure (SAGA), and we modified the SA algorithm into a population based version (PSA) to determine which would be most suitable for use as the search engine in our DSS. A statistical analysis of the test results with respect to the performance measures that we believe are most important has revealed that SAGA is better suited for our needs in the DSS (for the case of Ghana's EPDP). However, we also believe that to make a more general conclusion about the relative merits of SAGA and PSA, further analysis that is independent of Ghana's power grid is necessary.

EMPIRICAL STUDY WITH MULTIPLE CRITERIA – PART II

There has been significant interest in the research community in the use of "test-problem generators" as a basis of comparison to improve the generalizeability of the results of an empirical study (Dejong et al. 1997, Spears 1998). Many researchers have argued that simply adding additional test problems to well known test suites is perhaps not the best solution to providing a better understanding of the performance and behavior of algorithms over a general class of problems. Rather, a well designed random test problem generator (RTPG) strengthens an empirical study of an algorithm’s performance by allowing the relevant characteristics for a well specified class of problems to be systematically studied in a controlled fashion. To this end, we have developed a RTPG for the EPDP based upon an analysis of the Ghana model and some rational assumptions described in the ensuing discussion.

Random Test Problem Generator

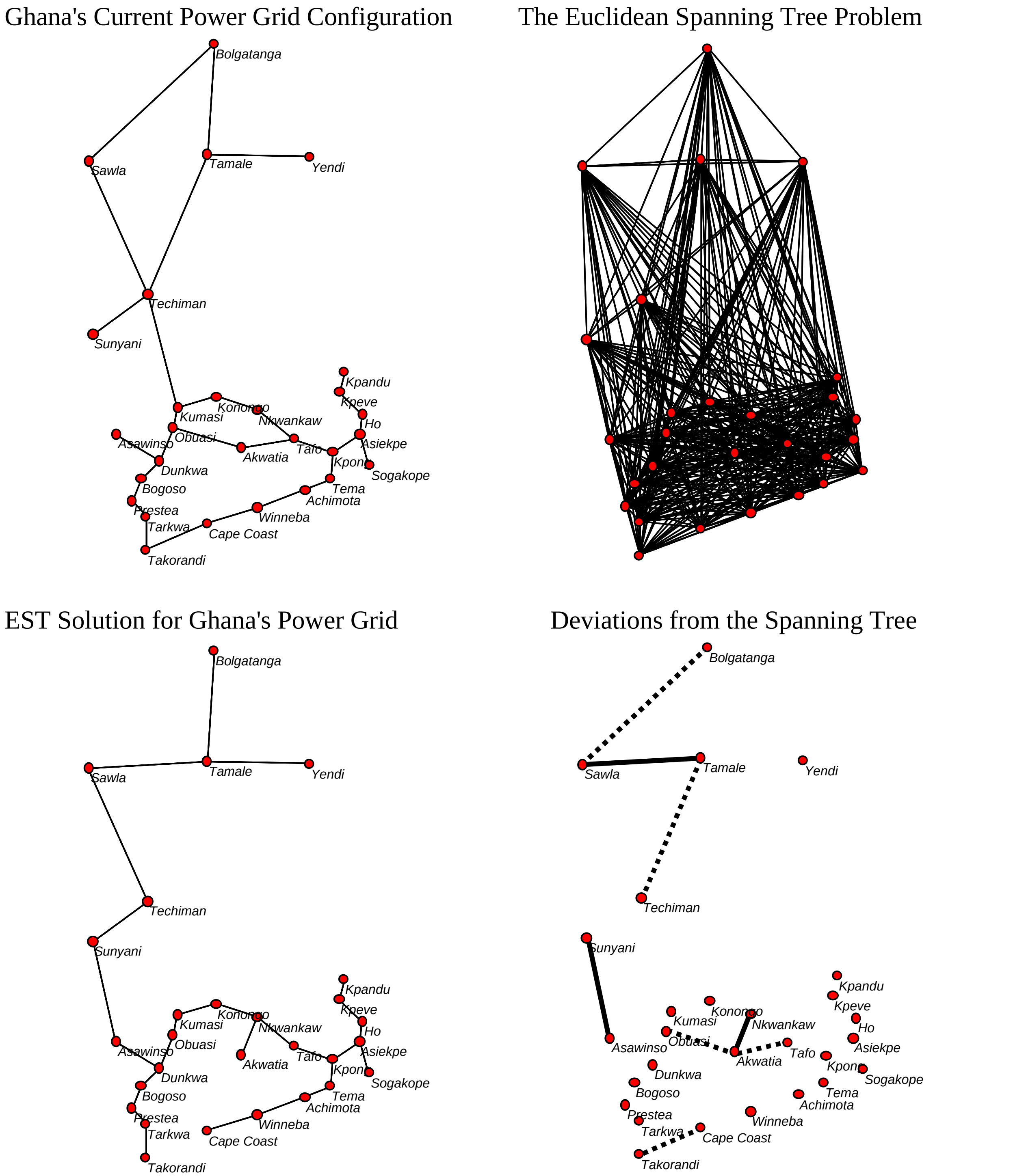
A very noticeable and intuitive characteristic of a large electrical power transmission network is that it tends to be sparse with regard to the density of the connection matrix. The rationale for this characteristic is two-fold. First, the raw material cost of power cable and associated equipment as well as labor costs required to lay high power transmission lines is extremely large. Second, the transmission losses over lines spanning great geographic distances is a substantial operating cost of electricity production. The resulting power grid for most national monopolies have naturally evolved as a sparse network of BSPs and transmission lines.

If a national power grid is to facilitate "open access" it must be a spanning network of transmission lines where each BSP has at least one connection to the network allowing any generator (supplier) to deliver electricity to any BSP (buyer). If cost were the sole criteria for the design, then the trivial solution to the problem would be a minimal spanning tree (MST) of transmission lines connecting the BSPs. However, this is not exactly the case with real world power grids where reliability, buffer capacity, political boundaries, and natural geographic barriers are design issues that cause a real power grid to deviate from the MST solution.

An Analytical Foundation for the RPTG

The current configuration of Ghana's power grid consists of 28 BSPs and 30 interconnecting transmission lines. For any network of n nodes there are exactly n-1 edges required to span the network such that all nodes are connected (minimal distance or not). Ghana's power grid requires at least n-1 = 27 transmission lines to connect all nodes and also contains 3 additional transmission lines. An interesting question is, "To what extent does Ghana's power grid resemble a MST?" To answer this question we must solve the Euclidean spanning tree (EST) problem using the x and y coordinates corresponding to each BSP. In the EST problem the network is fully connected. Solving the EST provides an ideal network design if Euclidean distance is used as the only proxy for cost. Figure 3.9 shows the remarkable similarity between the MST solution to the EST problem for Ghana's actual power grid.

FIGURE 3.9



Of the 30 actual transmission lines connecting the 28 BSPs in Ghana, 24 of the actual transmission lines are included in the MST solution to the EST problem while only 5 are not included. The actual configuration of Ghana's power grid contains only 3 connections more than the MST solution to the EST problem for an incremental increase of 11.1%. Note that the deviations of the actual configuration from the EST solution consistently connect to neighboring nodes that are "very nearby." This observation has strong implications in the design of our test-problem generator's random operator.

The MST, driven by the factors that reduce real costs experienced in the electric utility industry, serves as an excellent model to develop a RTPG for the general class of EPDP problems. While a MST is an ideal design for a transmission network, it is likely unachievable in a real world scenario. Therefore, it is reasonable to expect a limited number of random deviations away from the ideal configuration to be an approximation for a real system.

Our RTPG is based upon a greedy random operator that adds the nearest neighbor to the tree "most" of the time. Specifically, given an ordered list of available edges (increasing costs) used to connect unconnected nodes to the MST, the RPTG runs a series of Bernoulli trials to select the next node to be added to the tree, starting with the first item (edge) on the list. We introduce a control parameter gamma (γ: $0<\gamma\leq 1$) which determines the greediness of the spanning tree algorithm. When $\gamma=1$ the MST is guaranteed to be included in the problem, while $\gamma<1$ offers no such guarantee. If the first edge in the list is not selected, another Bernoulli trial is run for the next item on the list until the trial is successful. For example, consider the case where $\gamma=0.9$ and we have selected the first item from the ordered list of edges. The probability that the nearest neighbor (i.e., MST solution) is added to the tree is $P=\gamma=0.9$. The probability that the Z^{th} nearest neighbor is added to the tree is defined by the geometric distribution $P(Z=z)= (1-p)^{z-1}p$.

Once the greedy random spanning tree (GRST) is constructed it can then be perturbed by adding edges to the tree for the purpose of increasing the density of the connection matrix to the desired level. The RPTG achieves this with the control parameter alpha (α: $\alpha=1,2,\ldots$), which is the number of additional edges added to the GRST. The number of additional edges placed in the GRSP is a crucial problem characteristic because the density of the connection matrix determines, to a large extent, the number of possible solutions to the problem. We digress to explain this very important issue.

Lower Complexity Bound for the EPDP

The spanning tree solution (not necessarily minimal) provides valuable insight into the complexity of an EPDP. A characteristic of a spanning tree mentioned previously is that, for any network of n nodes there are exactly n-1 edges required to span the network, such that all n nodes are connected to the tree. Another noteworthy characteristic of a spanning tree that is relevant to solving the EPDP (or graph partitioning problems in general), is that for any spanning tree exactly t cuts (removal of an edge) will result in exactly t+1 contiguous districts regardless of the configuration of the network. Thus, the following sub-problem to any EPDP can be used to calculate a lower bound for the number of possible solutions to the EPDP at hand.

Consider the case of the MST solution to Ghana's EST problem. If we desire k districts in our solution to the EPDP in this configuration, making k-1 cuts in the MST will produce this result. The question, "How many different ways can we make k-1 cuts in a MST with n nodes?"

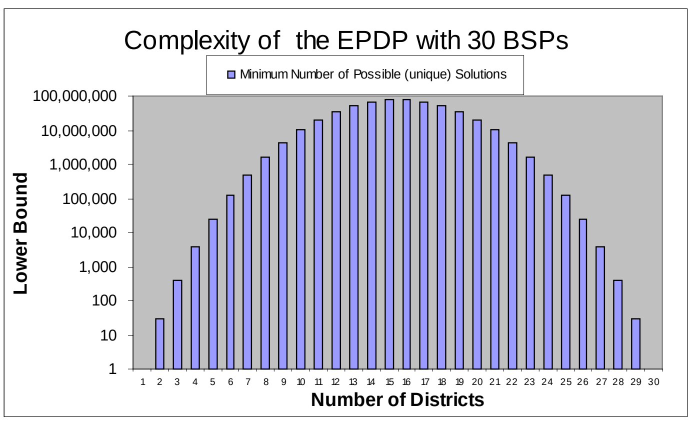
is equivalent to asking the question, "How many ways can we choose *k*-1 edges out of *n*-1 edges?" In this case, order is not important as cutting edge "Sawla-Tamale" and then cutting edge "Dunkwa-Obuasi" will result in the same districting plan (where *k*=3) as cutting them in reverse order. Clearly, as the additional edges are added to the spanning tree sub-problem the number of possible solutions to the problem will increase. So, the obvious lower bound for the number of solutions to any EPDP with *n* nodes and *k* districts can be calculated using the expression given in (3.14). Thus, the number of possible districting plans grows exponentially with the number of nodes in the network for any number of desired districts where 1<*k*<*n*-1.

$$\binom{n-1}{k-1} \tag{3.14}$$

It is important to realize that the analysis above is limited to the special case of the spanning tree sub-problem of an EPDP. However, it is equally important to make note of some problem characteristics that are manifested in this special case as they may have a profound impact on the development of solution algorithms designed to address this problem. For example, consider the distribution of the number of possible solutions with respect to the number of districts.

The bar chart provided in Figure 3.10 shows the lower bound for an EPDP with *n*=30 nodes for any number of desired districts in logarithmic scale. It is interesting to note that the complexity characteristic is symmetric (for a specified *n*) and centered at exactly *k*=*n*/2 when *n* is odd, and *k*=(*n*+1)/2 when *n* is even.

FIGURE 3.10



An important issue to keep in mind when designing a solution algorithm for solving this problem is the range of *k* that is typical for the problem. In particular, when the typical range of *k*<*n*/2, the representations used in an effective solution algorithm are likely to be quite different than if the typical range of *k*>*n*/2. As the typical range of *k* deviates further away from *n*/2 this concern is exacerbated. Recall that for EPDP problems, the typical range of *k* is expected to be 4≤*k*≤6. It is crucial to keep this information in mind as we make tactical and strategic decisions for developing effective solution algorithms for the general class of EPDPs.

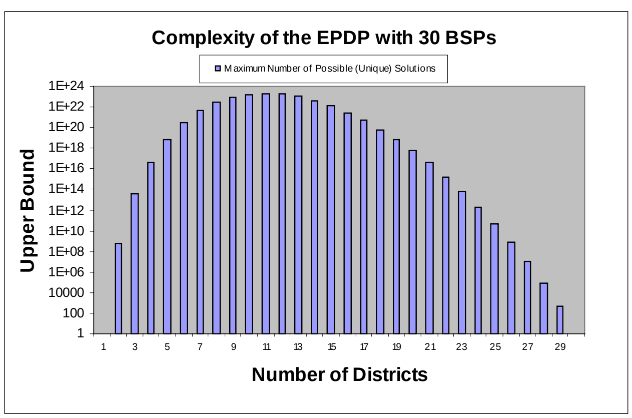
Upper Complexity Bound for the EPDP

Another way to gain some insight into the complexity of EPDPs is to consider the number of possible solutions at the opposite extreme of a spanning tree - the fully connected tree. In this extreme case, the problem is equivalent to a set partition problem where any BSP can be placed in the same district as any other BSP in the network. The number of possible districting plans (or unique solutions) to this problem is given by the Stirling number of the Second Kind denoted as $S(n,k)$ (Abramowitz 1995) . Simply described, it is the number of ways of partitioning a set of n elements into k non-empty sets. For example, Set {1,2,3} can be partitioned into k=1 subset in only one way {{1, 2, 3,}}, k=2 subsets in three ways {{1, 2}, {3}}, {{1},{2, 3}}, {{1, 3},{2}}, and k=3 subsets in only one way {{1},{2},{3}}. The formal expression for computing the Stirling number of the Second Kind for any n and k is:

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k-1} -1^{i}\binom{k}{i}(k-i)^{n} \tag{3.15}$$

Clearly, this would provide an upper bound for the number of possible solutions to an EPDP. However, EPDPs are more appropriately characterized as sparse networks and will therefore be closer to the lower bound in its complexity. Figure 3.11shows the upper bound for an EPDP with n=30 nodes and any number of desired districts k, where $1<k<n-1$. Note that it has a similar shape characteristic to the lower bound complexity bar graph in Figure 3.10. However, it is not perfectly symmetric. Starting with a small number of districts, the number of possible solutions to the problem increases up to a point and then begins to decrease again. But, it differs in that the maximum number of possible solutions does not peak at $n/2$ or $(n+1)/2$. Rather, the peak occurs at some point slightly before then as the distribution of possible solutions is noticeably skewed to the right.

FIGURE 3.11

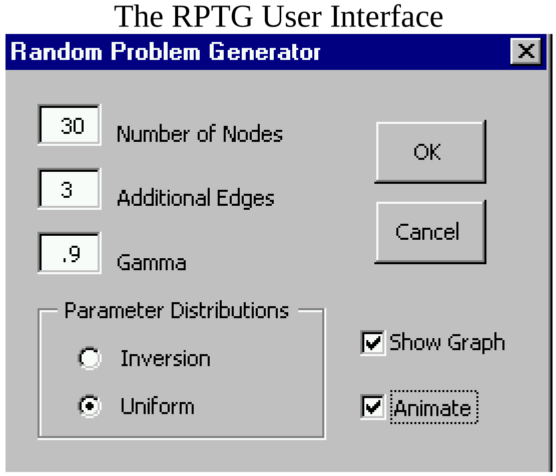


We are unaware of a formal expression for the number of possible solutions to a districting problem characterized in terms of the connection matrix. We believe that this would be an interesting and challenging path for future research. It has been rigorously proven that the (political) districting problem, having similar constructs to the EPDP, is in the family of NP-hard problems (Altman 1995). Specifically, Altman's dissertation included four proofs. First, creating equal population districts is NP-hard. Second, creating a maximally compact districting plan is NP-hard. Third, creating a plan with maximally competitive districts is NP-hard, where a maximally competitive plan was defined as one that minimized the overall expected difference between Republican and Democratic registration in each district. Fourth, creating a contiguous equal-population districting plan is NP-hard.

Distribution of Data

Another factor that must be considered while developing a RTPG for an EPDP is the distribution of randomly generated data to be used as location information and revenue information for the problems generated. Our RTPG provides two alternatives for generating such data. The first option is based upon the inverse transformation technique for random number generation, which uses the distribution of the data in the Ghana case as a model for creating randomly generated information for problems. The second option provided by the RPTG is a Uniform distribution over the same standardized range as the Ghana data. While both approaches have merit in their use for empirical testing, we chose to use the Uniform distribution of data in our investigation. We determined experimentally that the Uniform distribution provided EPDP configurations that appeared to approximate a real system more accurately than the inverse transformation technique. In addition, we believe that the Uniform distribution will provide more valuable test results than an inverse transformation technique as it is independent of the actual case of Ghana and thus, more generalizable. The interface for our RPTG is shown in Figure 3.12.

FIGURE 3.12
The RPTG User Interface



Generating a Suite of Random Problems

We have used the RPTG described in the previous section to generate a test suite of 30 random problems. Recall that gamma (γ: $0<\gamma\leq1$) is the control parameter for the greediness of the spanning tree algorithm used to connect the nodes to the network. We used a setting of γ=0.9 for generating the entire test suite. The number of nodes (n) for the problems in the test suite is held constant at n=30. Recall that the control parameter alpha (α) is used to adjust the density of the connection matrix by adding additional edges. We used a setting of α=3 or (10% of n) to perturb the GRSP for all of our randomly generated problems. The choice of settings for generating the test suite was based upon our analysis of Ghana's power grid. Indeed, additional settings of our RTPG will produce EPDP problems with differing characteristics in a controlled fashion. We readily admit that such practices would increase the robustness of an empirical study and strengthen the generalizability of the results. We believe that this would be an interesting path to explore in future research.

Random Search

In order to provide a foundation for comparing the performance of the SAGA and PSA, it is reasonable to also include the results of a totally random search procedure. The rationale for such inclusion is that any new procedure, in particular stochastic search algorithms, must be shown to outperform the "naive method." The random search procedure that we implement is identical to the random starting heuristic used to initialize the populations of solution vectors in SAGA and PSA. In the random search, we used the same framework of a population based search having a parent population and child population. However, rather than using a heuristic mechanism to produce new solutions we regenerated new random solutions in each iteration until reaching the maximum number of function evaluations. In the Random search, the Pareto ranking and selection procedure was used to maintain the best solutions found (currently non-dominated) during each optimization run. To this end, we report and compare the results of SAGA, PSA, and Random search on the randomly generated problems in our test suite. Similar to the empirical study in Part I, a population size of 25 solution vectors was used and each procedure was allowed to run for exactly 10,000 function evaluations.

Comparing SAGA, PSA, and Random Search with Multiple Problems

In contrast to the manner in which the previous empirical studies were performed, where multiple optimization runs are performed on a single problem (Ghana), we are now challenged with understanding the behavior of the same solution techniques for multiple problems. Our approach to performance testing in Part II is to complete a single optimization run using each of the three search techniques and examine the differences between techniques as matched pairs. Thus, unlike the previous statistical tests, where the variable of interest was the difference in average performance across 30 optimization runs, this approach creates three matched pairs for each problem (e.g., SAGA vs. PSA, SAGA vs. Random, PSA vs. Random). This method is recommended for comparing solution methods with an RTPG (Dejong et al. 1997, Spears 1998).

The performance measures that we use in our empirical study in Part II are the same as those described in Part I. First, we compare the number of non-dominated solutions in the final population of solution vectors for each technique. Recall that this is our measure for coverage of the efficient frontier. Second, we compare the number of equivalence class layers in the final population of solution vectors. We reiterate that the purpose of this performance measure is to examine the quality of the "soft frontier" for use in our DSS. In Part II of our empirical study, we have limited our investigation to k=5 districts, which was the actual scenario requested by the

DMs at the World Bank for the Ghana case. Thus, we believe that k=5 districts is the most important single scenario to investigate in a practical amount of time.

The Non-dominated Solution Set

Figure 3.13 shows the first equivalence class layer in the final solution set for test problem number 2 (labeled Problem Number-n-α) resulting from SAGA, PSA and Random search. Clearly, both SAGA and PSA produce a superior set of solutions than the Random search method. This result is consistent throughout all of the performance testing in Part II. We also note that for the specific test problem shown in Figure 3.13, the number of non-dominated solutions produced by SAGA is substantially greater that those found by PSA. The coverage of the efficient frontier by SAGA in this particular problem is impressive. However, we stress that these are the results of a single optimization run and must admit that Problem 2 showed the most impressive difference between SAGA and PSA for any of the problems in our randomly generated test suite. However, there are several occasions where PSA produces more non-dominated solutions than SAGA. For example, Problems 5, 19 and 23 shown in Table 3.7. For the curious reader, diagrams similar to 3.13 for the entire test suite are provided in Appendix C.

FIGURE 3.13

Equivalence Class Layer 1 for Random Problem 2, with n=30 and α=3

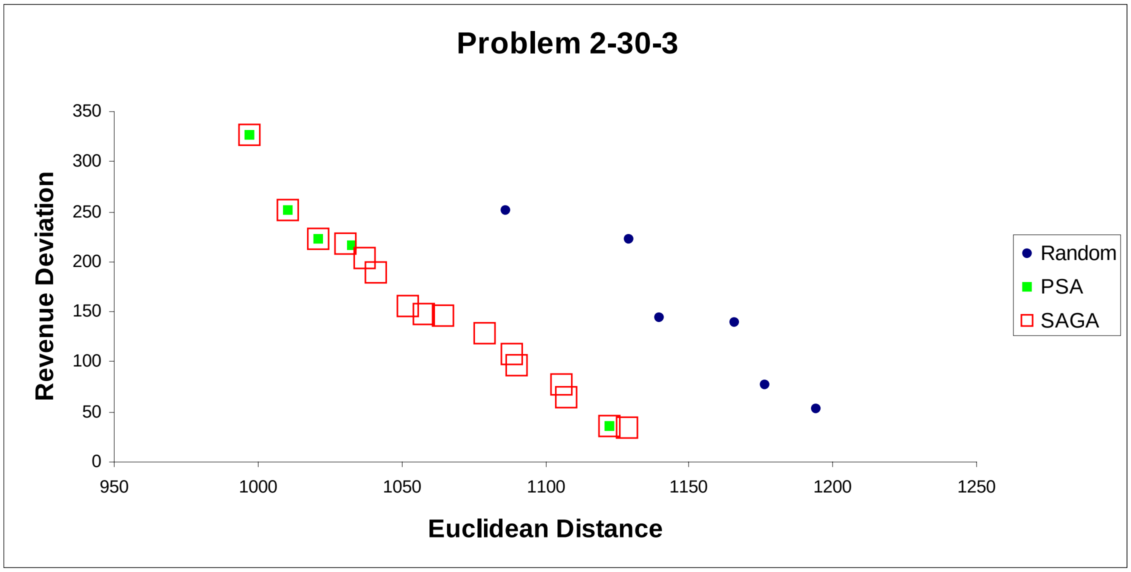


Table 3.7 provides the number of non-dominated solutions generated by each of the three solution techniques for each of the problems in our test suite. In Table 3.7, non-dominated solutions were determined by considering all of the solutions produced by each of the three techniques after the completion of all optimization runs. Thus, some solutions in the first equivalence class layer from one solution technique were later dominated by solutions in the first equivalence class layer from one of the other techniques. In addition, Table 3.7 also gives the variable of interest for our statistical test - the difference in the number of non-dominated solutions for paired observations forming matched samples.

TABLE 3.7
Number of Non-dominated Solutions for Random Test Suite

Number of Non-dominated Solutions				Differences In Matched Samples		
Problem	SAGA	PSA	Random	SAGA-PSA	PSA-Random	SAGA-Random
1	6	5	0	1	5	6
2	16	5	0	11	5	16
3	4	3	0	1	3	4
4	3	2	0	1	2	3
5	3	4	0	-1	4	3
6	4	3	0	1	3	4
7	4	4	0	0	4	4
8	6	5	0	1	5	6
9	6	3	0	3	3	6
10	5	4	0	1	4	5
11	8	6	0	2	6	8
12	3	2	0	1	2	3
13	4	3	0	1	3	4
14	4	4	0	0	4	4
15	2	2	0	0	2	2
16	6	2	0	4	2	6
17	4	1	1	3	0	3
18	4	3	0	1	3	4
19	5	6	0	-1	6	5
20	4	4	1	0	3	3
21	5	4	0	1	4	5
22	5	2	0	3	2	5
23	3	4	0	-1	4	3
24	6	3	0	3	3	6
25	8	3	0	5	3	8
26	5	3	0	2	3	5
27	5	3	0	2	3	5
28	6	5	0	1	5	6
29	2	2	1	0	1	1
30	7	3	0	4	3	7
Average	**5.100**	**3.433**	**0.100**	**1.667**	**3.333**	**5.000**

Table 3.8 indicates via pairwise comparison of each procedure that a statistically significant number of non-dominated solutions is produced across the entire test suite. It can by gleaned by inspection of Appendix C that both SAGA and PSA are clearly superior to Random search.

TABLE 3.8
Non-dominated Solutions Test Results

Paired Procedures	t-value
SAGA vs. PSA	3.928*
PSA vs. Random	13.298*
SAGA vs. Random	10.326*

* t-values significant at α=.005

Equivalence Class Layers

Figure 3.14 shows the results of a single optimization run of SAGA on Problem number 2, with the final population of solution vectors organized into 2 equivalence class layers. Also shown is the configuration of the randomly generated transmission network in Problem 2. Note that a particular districting plan is also shown in the network where each BSP is assigned to a district with a color-coding. Each equivalence class layers corresponds to a series shown in the legend of the scatter plot as described previously for the case of Ghana. Recall that one solution vector in the first series is highlighted with a large red dot. This is the solution vector corresponding to the districting plan that is rendered in the power grid. The districting plan shown is non-dominated and has the minimum total Revenue Deviation (f_1=32.23, f_2 =1129.43) relative to all other plans. For the sake of completeness, similar diagrams for the entire test suite are provided in Appendix D. Again we point out that the difference between SAGA and PSA on Problem 2 is impressive.

FIGURE 3.14

Min Rev Dev Plan found by SAGA — Equivalence Class Layers in Final Population of SAGA

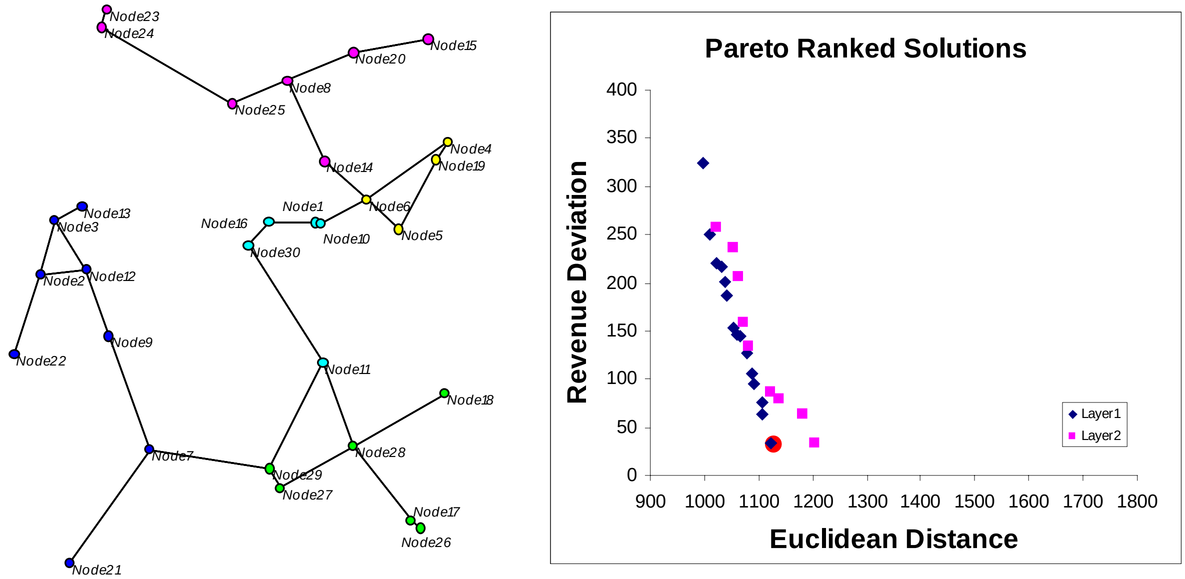
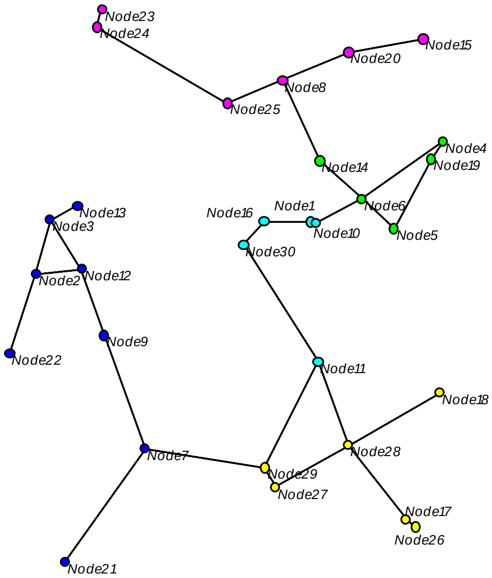


Figure 3.15 shows the final population of solution vectors generated using PSA organized into 5 equivalence class layers. The highlighted solution vector corresponding to the districting plan shown in the randomly generated transmission network is non-dominated and has the minimum total revenue deviation (f_1=33.07, f_2 =1123.17) from its respective Pareto set. It is interesting to note that the solutions illustrated in Figures 3.14 and 3.15 differ only in the assignment of Node 14 to an alternative district. We also point out that PSA failed to locate the solution vector highlighted in Figure 3.14, which is less compact but has a slightly lower total revenue deviation. The solution highlighted in Figure 3.15 corresponds to the solution that is adjacent to the highlighted solution in Figure 3.14. Clearly, the final set of solution vectors in

Figure 3.14 represent a tighter fit with the non-dominated set than those in Figure 3.15 and thus, is a higher quality sample of alternatives to present to DMs.

FIGURE 3.15

Min Rev Dev Plan found by PSA | Equivalence Class Layers in Final Population of PSA



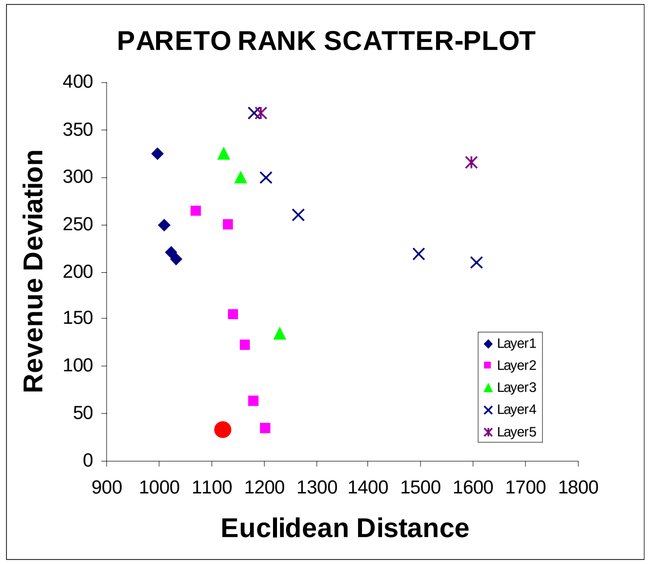


Table 3.9 provides the number of equivalence class layers generated by each technique for each of the problems in our test suite. In addition, Table 3.10 also gives the variable of interest in the right column (the difference between SAGA and PSA forming a matched sample).

TABLE 3.9

Equivalence Class Layers for Random Test Suite

Number of Pareto Layers				Difference in Matched Samples
Problem	SAGA	PSA	Random	(SAGA-PSA)
1	6	5	5	1
2	2	5	4	-3
3	6	8	6	-2
4	6	7	7	-1
5	6	7	6	-1
6	6	8	6	-2
7	6	6	7	0
8	7	9	3	-2
9	5	9	4	-4
10	4	7	4	-3
11	4	7	4	-3
12	5	6	4	-1
13	9	11	5	-2
14	7	4	6	3
15	10	11	7	-1
16	8	10	5	-2
17	5	9	5	-4

18	5	5	6	0
19	6	6	4	0
20	5	10	5	-5
21	6	5	6	1
22	6	8	5	-2
23	7	6	7	1
24	5	6	6	-1
25	6	9	8	-3
26	9	8	4	1
27	9	8	6	1
28	7	7	6	0
29	5	12	3	-7
30	6	6	5	0
Average	**6.133**	**7.5**	**5.3**	**-1.367**

Table 3.10 indicates that there is a statistically significant difference in the number of equivalence class layers produced across the entire test suite between SAGA and PSA.

TABLE 3.10

Equivalence Class Layers Test Results

Districts	t-value
5	-3.549*

* t-value significant at α=.005

False Domination

Table 3.9 clearly shows that the average number of equivalence class layers produced by Random search is less than the average number produced by SAGA and PSA. We intentionally do not report the t-statistic for the number of equivalence class layers comparing SAGA or PSA to Random search as it would be of no practical value and possibly misleading. The misleading aspect is the fact that the non-dominated set produced by the Random search is comprised nearly entirely of solution vectors that are "false dominants." It is interesting that the equivalence class layers in the final set of solution vectors developed from the Random search are indeed tightly fit about their respective Pareto set. However, the final solution set produced by the Random search is of a much lower quality than those produce by either SAGA or PSA.

This propensity of the Random search to produce false dominants is very noticeable with a visual inspection of Appendix C. The fact that Random search tends to produce a final solution set with fewer equivalence class layers that are tightly fit to the false dominants in its respective first equivalence class layer is also apparent in Appendix D, and thus, we provide no further evidence in the body of this report. However, for the intensely curious reader, we provide a more rigorous analysis of the "false-dominants" that were found in this empirical study in Appendix E. We believe that the tendency of the Random search to exhibit this joint behavior can be attributed to the Pareto ranking and selection system that was used to maintain the non-dominated solutions in the population following the random generation of new solution vectors in each generation. Indeed, the transitivity property of the non-domination relation holds as each set of 25 randomly generated solution vectors is produced and then Pareto ranked.

CONCLUSION

This chapter investigated the EPDP centered on Ghana's actual power grid as a multi-criteria problem. The general class of the EPDP was characterized by our understanding of the issues derived from the case of Ghana and some insights that we developed from our research into related issues. We also learned a great deal about the EPDP from those who are interested in solving this very important real world problem – the DMs at the World Bank.

In chapter 2, we performed a static and a dynamic analysis of two well-known general-purpose stochastic algorithms – the GA and SA. We found that each had redeeming qualities that could be combined to produce more effective solution methods for the EPDP. In chapter 3, we addressed the more realistic multi-criteria version of the EPDP with our modified algorithms – SAGA and PSA. Again, we investigated the specific case of Ghana's actual power grid and also performed a sensitivity analysis of SAGA and PSA with the use of a random problem generator. Both modified solution methods were inspired by the notion that a population based search method is more effective in searching for an optimal solution set, as opposed to a point based method searching for a unique optimum.

PSA enhanced the point based approach of SA with multiple solution vectors organized into a population having multiple (unique and constant) search directions but did not make use of any concepts related to Pareto dominance. This could be a reasonable explanation why the final solution set formed a "soft frontier" that was less dense than SAGA (and Random search) which employed the concept of Pareto dominance directly. However, the use of the Pareto dominance concept alone (Random Search) proved to be insufficient in locating solutions on the efficient frontier relative to search techniques that were guided by weights.

SAGA was intended to combine the redeeming qualities that we observed in both the GA and SA from chapter 2. Thus, SAGA was the only modified solution method which truly exploited both of the features of guided search and the concept of Pareto dominance. In each generation, the surviving solution vectors were selected using the Pareto ranking technique and then allowed to reproduce in a fashion similar to a typical GA. Then, the weights were reassigned to the offspring of the parents based upon the relative location of the children in the multi-criteria solution space. Subsequently, a rapid cooling procedure motivated by SA was applied to allow the children to improve rapidly with a limited amount of computing resources. One may say that this is akin to what we observe in real life, where newborns seem to become acclimated to their environment at a much quicker pace in the earlier stages of their life.

The results that we observed in the empirical studies in chapter 3 show that there is a statistically significant difference between the modified solution methods in nearly all of the scenarios that we studied. For the majority of our study, SAGA outperformed PSA and Random search in its ability to locate non-dominated solutions and produce a high quality (compact) frontier in the vicinity of the actual non-dominated set. Thus, we conclude from the collective of our empirical studies that SAGA is provably the solution technique that is best suited as the primary search engine for our DSS.