Annals of Operations Research 121, 33–55, 2003

A Simulated Annealing Genetic Algorithm for the Electrical Power Districting Problem *

PAUL K. BERGEY ** Paul_Bergey@ncsu.edu
Department of Business Management, North Carolina State University, Raleigh, NC 27695, USA

CLIFF T. RAGSDALE
Department of Business Information Technology, Virginia Tech, Blacksburg, VA 24060, USA

MANGESH HOSKOTE
The World Bank, 1818 H Street NW, Room No. J9-027, Washington, DC 20433, USA

Abstract. Due to a variety of political, economic, and technological factors, many national electricity industries around the globe are transforming from non-competitive monopolies with centralized systems to decentralized operations with competitive business units. A key challenge faced by energy restructuring specialists at the World Bank is trying to simultaneously optimize the various criteria one can use to judge the fairness and commercial viability of a particular power districting plan. This research introduces and tests a new algorithm for solving the electrical power districting problem in the context of the Republic of Ghana and using a random test problem generator. We show that our mimetic algorithm, the Simulated Annealing Genetic Algorithm, outperforms a well-known Parallel Simulated Annealing heuristic on this new and interesting problem manifested by the deregulation of electricity markets.

Keywords: electricity deregulation, genetic algorithms, simulated annealing, multi-criteria decision making

Energy restructuring specialists at the World Bank regularly face the challenge of helping developing countries move from state owned, monopolistic electric utilities to a more competitive environment with multiple electricity service providers. A necessary first step in the electricity sector restructuring process is the division of the physical assets associated with a country's power grid into operational and functional units consistent with an appropriate market design. In theory, an "optimal" segmentation of the monopoly state-owned company would give each functional business unit an equal opportunity to become profitable and attract investors and customers.

The process of partitioning a physical power grid into economically viable districts (distribution companies) is referred to as the electrical power districting problem (EPDP) [3]. The motivation for this study is to evaluate the performance of two stochastic search algorithms designed to serve as an optimization engine for a decision support system (DSS) that allows decision makers (DMs) at the World Bank to visually explore

* This research was supported, in part, by the Edwin Gill Foundation at North Carolina State University.
** Corresponding author.

and quantify the trade-offs associated with various segmentation plans. To this end, we establish performance testing benchmarks for the EPDP in the context of a case study of the Republic of Ghana using data provided by the World Bank. In addition, we introduce a random test problem generator (RTPG) for the EPDP and present rationale for its design. Finally, we demonstrate the utility of the RPTG via performance testing of our algorithm.

1. The Republic of Ghana

The Government of Ghana is undertaking an industry reform program for their emerging competitive electricity market, with the objective of developing an economically stable, customer oriented industrial culture in Ghana. The national electricity transmission grid of Ghana consists of 28 interconnected Bulk Supply Points (BSPs). See figure 1.

BSPs represent junction points where high voltage electricity is reduced for local services. Table 1 provides the input data for the BSPs included in this study. The in-

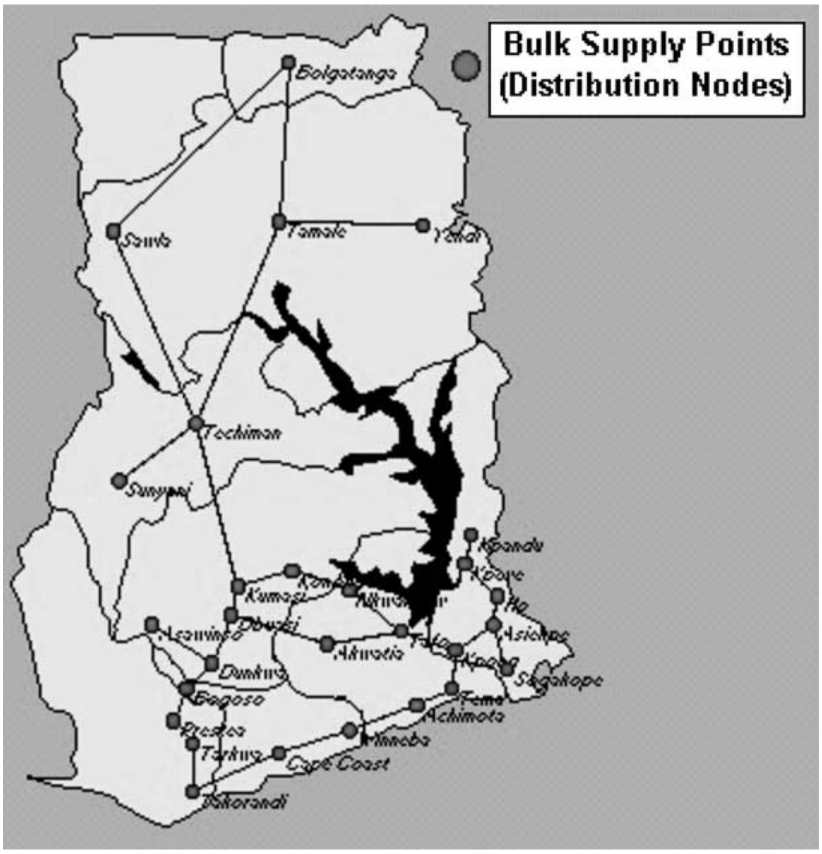

Figure 1. The national electricity power grid of Ghana.

Table 1
Bulk supply point information.

Index	BSP	Expected revenue ($M)	X	Y
1	Achimota	30.518	21	8
2	Sawla	0.006	6	34.5
3	Tamale	1.943	17	35
4	Yendi	0.007	23	35
5	Bolgatanga	0.873	17	45
6	Sunyani	1.978	6	20.5
7	Techiman	1.127	8.5	23
8	Tema	12.369	23	9
9	Kpong	2.511	23	12
10	Sogakope	0.294	27	11.5
11	Ho	0.403	26	15
12	Asiekpe	0.010	26.5	13
13	Kpandu	0.616	25	18
14	Kpeve	0.208	25	16
15	Nkwankaw	1.585	7.5	20
16	Akwatia	0.571	17	11
17	Tafo	0.985	19.3	13
18	Cape Coast	1.654	14	5
19	Winneba	1.053	18	6.5
20	Tarkwa	3.104	9	6
21	Takorandi	5.112	10	3
22	Prestea	1.610	8	7
23	Bogoso	1.479	8.5	8
24	Asawinso	0.623	6.5	13
25	Kumasi	11.284	11	16
26	Konongo	0.434	14	15
27	Obuasi	1.111	11	12.5
28	Dunkwa	0.163	10	11.5

[a] Data provided by the World Bank.

formation consists of the index number and name corresponding to each BSP shown in figure 1, the expected annual revenue potential (provided in millions of dollars) and the standardized physical coordinates for each BSP on a two dimensional map.

2. The electrical power districting problem

The following sections provide a mathematical framework for the EPDP. A districting plan is an assignment of units (BSPs) to non-overlapping districts (groups) that are contiguous (adjacent). The EPDP is primarily concerned with creating groups with approximately equal earning potential that are also geographically compact. The motivation for having equal earning potential is to provide an environment that will foster competition. The rationale for a compact district design is that districts which are compact over a geographic region (rather than disbursed) will be easier to manage, more economical to maintain and thus more profitable. There are a number of alternatives that can be added to this formulation to increase the complexity of the problem. However, we submit that balanced revenue and geographic compactness represent two fundamental characteristics of EPDP. Indeed, we expect alternative formulations for the EPDP to emerge as researchers explore this exiting new opportunity spawned by the deregulation of electricity markets.

2.1. Background on districting problems

The research literature contains a variety of mathematical characterizations for the generalized districting problem as well as several suitable application areas. The application areas include, but are not limited to, political redistricting, sales territory alignment, school redistricting, and turf allocation models for the forestry and telecommunications industries. We posit that the EPDP has similar constructs to the above applications and represent a huge open opportunity for researchers interested in districting problems to create substantial social and economic benefit. This section provides a review of the various applications of districting problems covered in the research literature, the manner in which the problem has been characterized, and the various solution techniques applied.

The most prevalent redistricting applications in the management science literature are political redistricting and sales territory alignment. The primary goals of political redistricting problems are to provide geographical districts that are compact and contiguous, yet respectful of existing political units to the maximum extent possible. In addition, the districts must have populations with approximately equal voting potential for the subsequent period. The motivation for using a computerized solution method is to reduce the effects of "Gerrymandering", which may occur when political incumbents bias the redistricting solution to accommodate their political agenda.

Automated political redistricting has been of great interest to politicians and researchers for the last four decades [1,2,4,9,15,26,35,36]. The first mathematical characterization of the political districting problem was proposed by [36]. The redistricting problem has been characterized in the recent literature as a set partitioning problem [1], a graph partitioning problem [4,7], a polygon dissection problem [12], and an integer programming problem for redrawing congressional districts in the state of South Carolina [24].

Designing sales territories is another application area that can be viewed as a districting problem. The sales territory alignment problem is concerned with grouping a number of smaller geographic regions into clusters forming non-overlapping sales territories that span a larger geographic region. Sales territories may need to be realigned whenever changing market conditions warrant, such as the introduction of a new product or variation in sales force size. Sales managers are motivated to carefully design an equitable districting plan, otherwise they risk low morale, poor performance, high turnover rate, and ultimately low productivity within the sales force. Furthermore, a balanced territorial design provides a more consistent and effective means of evaluating individual performance.

There are a number of criteria that can be used to assign geographic regions to districts or sales people to customers. Some single alignment criteria methods seek to balance income or revenue potential [23], while other methods attempt to balance workload or effort [6]. Another common objective is to maximize overall expected profitability. The consideration of multiple objectives was first proposed by Zoltners [37].

Sales territory redistricting has a considerable history in the research literature. Similar to the political redistricting problem, a variety of solution techniques have been applied to this problem. A set-partitioning approach was investigated by Shanker et al. [32]. Various assignment methods have been employed by Hess and Samuels [14], Segal and Weinbergey [33], Zoltners [37]. For a thorough review of integer programming approaches, see [38]. Heuristic approaches that utilize incremental improvements were proposed by Easingwood [6], Lodish [23], Heschel [13].

The relationship between a political district and a sales territory is fairly obvious. The political district is driven by the "one person – one vote" principle, which seeks to balance legislative power among districts. When assigning geographic regions to sales territories or salespeople, most equitable solutions seek to balance income potential or effort. Both scenarios can be treated as a bin-packing problem, which seeks to minimize the total deviation of some criteria in each bin (district) from the ideal (global) bin mean. Typically, there are other non-trivial considerations distinguishing a bin-packing problem from a districting problem, such as the compactness of a districting plan or contiguity among the geographic regions allocated to a particular district.

2.2. *Balanced revenue objective for the EPDP*

A districting plan where n nodes (bulk supply points) are assigned to k districts can be described by the solution vector $x = (x_1, x_2, \ldots, x_n)$, where $x_i = j$ if node i is assigned to district $j \in \{1, \ldots, k\}$. To obtain k transmission districts of approximately equal earning potential we can minimize the total deviation of the revenue in each district from a target value. This can be modeled as follows:

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

$$\text{subject to} \quad \overline{r} = \frac{1}{k} \sum_{i=1}^{n} r_i, \tag{2}$$

$$R_j = \sum_{i \in D_j} r_i \quad (j = 1, \ldots, k), \tag{3}$$

$$\left| R_j - \overline{r} \right| \leqslant \delta \overline{r} \quad (j = 1, \ldots, k), \tag{4}$$

where:

$D_j = \{i \mid x_i = j\}$;

r_i = the amount of revenue potential contained in node i;

R_j = the sum total of revenue potential in district j;
$100\delta,\ 0 \leqslant \delta \leqslant 1$, is the maximum allowable percentage
deviation of the actual revenue in a district from the target (optional).

Note that the above formulation does not explicitly model the difficult task of ensuring contiguity within each district. The above model assumes that nodes assigned to a particular district are contiguous. Section 5.2 addresses how our solution methodology maintains contiguity in each district.

2.3. Compactness objective for the EPDP

To measure the compactness for any district j, we use the total Euclidean distance from the centroid of district j to each node assigned to district j. The districting plan $x = (x_1, x_2, \ldots, x_n)$ that minimizes the total compactness is then found by solving the following problem:

$$\text{minimize } f_2(x) = \sum_{j=1}^{k} K_j \tag{5}$$

$$\text{subject to} \quad K_j = \sum_{i \in D_j} \sqrt{\big(C_x(j) - L_x(i)\big)^2 + \big(C_y(j) - L_y(i)\big)^2} \quad (j = 1, \ldots, k), \tag{6}$$

$$C_x(j) = \frac{1}{N(D_j)} \sum_{i \in D_j} L_x(i) \quad (j = 1, \ldots, k), \tag{7}$$

$$C_y(j) = \frac{1}{N(D_j)} \sum_{i \in D_j} L_y(i) \quad (j = 1, \ldots, k), \tag{8}$$

where:

K_j = compactness measure of district j;
$D_j = \{i \mid x_i = j\}$;
$N(D_j)$ = cardinality of (or number of nodes in) district j;
$C_x(j) = x -$ coordinate for the centroid of district j;
$C_y(j) = y -$ coordinate for the centroid of district j;
$L_x(i) = x -$ coordinate for the location of node i;
$L_y(i) = y -$ coordinate for the location node I.

2.4. The multi-criteria EPDP model

The following section provides a Multi-Criteria Decision Making (MCDM) formulation for solving the EPDP. Recall that (1) is defined as the total revenue deviation of a districting plan and (5) is defined as the total Euclidean distance (compactness) of a districting plan. We use the following model to simultaneously minimize the components of criterion vector $F(x)$:

$$\text{minimize } F(x) = \big(f_1(x), f_2(x)\big) \tag{9}$$

$$\text{subject to} \quad \text{equations (2)–(4) and (6)–(8).} \tag{10}$$

When solving a MCDM problem, it is generally assumed that DMs prefer or desire to obtain Pareto optimal (or non-dominated) solutions. For the criterion vector $F(x) = (f_1(x), f_2(x), \dots, f_q(x))$, a solution vector $x^* \in \mathbb{R}^n$ is said to *dominate* $x \in \mathbb{R}^n$ if $f_i(x^*)$ is at least as good as $f_i(x)$ for all i and $f_i(x^*)$ is better than $f_i(x)$ for at least one i. A solution vector $x^* \in \mathbb{R}^n$ is *Pareto optimal* if there exists no other solution vector $x \in \mathbb{R}^n$ that dominates x^*.

3. Simulated annealing

In this section, we provide a brief overview of the Simulated Annealing (SA) algorithm. The SA algorithm is a general purpose optimization procedure based upon the thermodynamic process of annealing metals by slow cooling [18]. At high temperatures, molecules in metal move rapidly with respect to one another. If the metal is cooled sufficiently slow then thermal mobility is lost. The resulting configurations of atoms are aligned to form a pure crystal that is completely ordered. This ordered state occurs when the system has achieved minimum energy.

To achieve low energy configurations, the annealing process must be cooled sufficiently slow to reach thermal equilibrium. The compelling reason for this is that the energy of a system (in thermal equilibrium) is probabilistically distributed according to the Boltzmann distribution. At high temperatures, the algorithm visits a very large neighborhood of the current state. As cooling takes place, the system accepts high and low energy states, but the lower the temperature, the lower the likelihood of accepting an uphill move (higher energy states). Therefore, the transitions to higher energy states become less frequent, and the solution stabilizes.

The SA algorithm starts at some high temperature T_c and in some state s. A sequence of points is then generated using a neighborhood operator to find some other configuration t. The probability of moving from state s to a neighboring state t is a function of the level of energy, $E(s)$ and $E(t)$, of each state. The state transition can be characterized by two conditions. First, if $E(t) - E(s) < 0$ then the new configuration is accepted deterministically because it represents a desirable reduction in energy level. Second, if $E(t) - E(s) \geqslant 0$, indicating an uphill move to a higher energy state, the probability that state t is accepted is provided in (11):

$$P(\text{accept state } t) = \mathrm{e}^{-(E(t)-E(s))/T_c}. \tag{11}$$

A cooling schedule for T_c is also necessary. The cooling schedule consists of the sequence of temperature changes and the amount of time spent at each temperature. If the annealing process reaches thermal equilibrium at each temperature, it can be proven that it produces the global minimum energy at absolute zero T_z [25].

4. Evolutionary algorithms

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) [16], Evolution Strategies (ES) [27] and Genetic Programming [20].

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 [31] and using an ES by Kursawe [21]. For an excellent general overview of evolutionary approaches to the MCDM problem see [8]. A Pareto ranking technique was proposed in the Pareto Genetic Algorithm (PGA) [11]. 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.

5. A simulated annealing genetic algorithm for the EPDP

In this section, we introduce a hybrid algorithm for solving the multi-criteria formulation of the EPDP provided in (9), (10). 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 shortcoming of traditional Genetic Algorithms (GAs) 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 [30]. This is the motivation for combining GAs with local search operators [28,29]. In order to solve partitioning problems effectively, it is necessary to design operators specifically for the task. The following section is intended to describe in detail the genetic operators that have been tailored specifically for solving the EPDP.

5.1. The crossover operator

In most GA implementations, constraints are enforced via penalty functions. When the size of the constraint set is large, the penalty function method often results in the

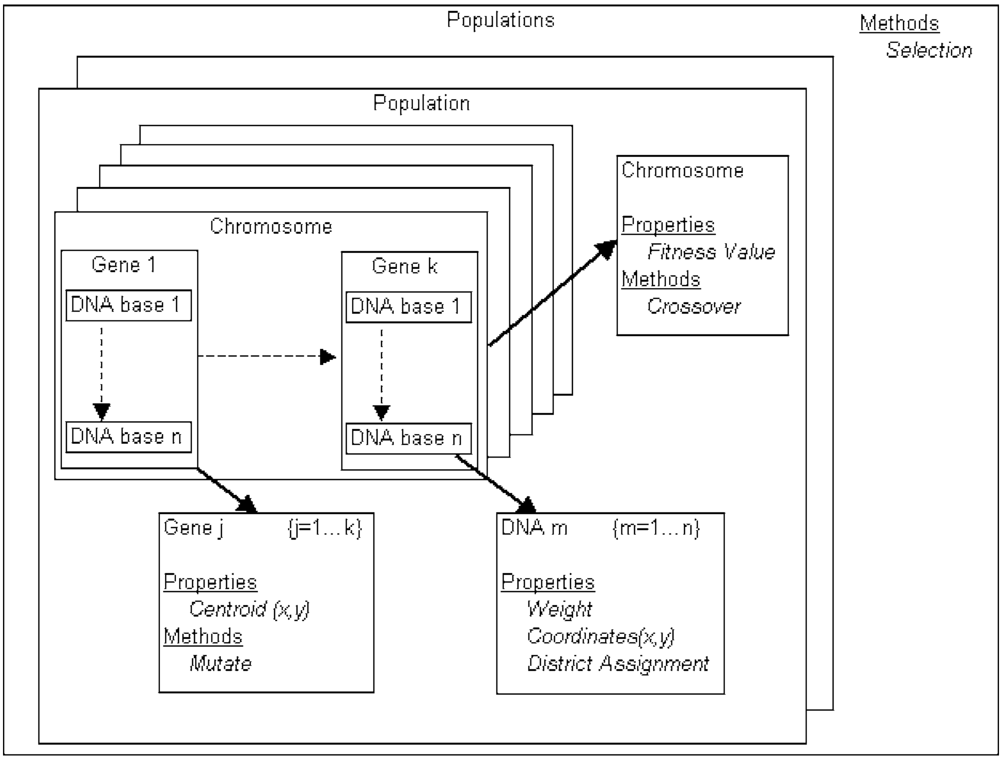

Figure 2. DNA object model.

production of a substantial number of infeasible chromosomes. Thus, a great deal of computational effort is wasted in the process. Enforcing contiguity in a districting plan requires an exponential number of constraints in the model with respect to the number of nodes in the graph [24]. An alternative to including the constraint set in the model formulation would be to enforce contiguity strictly in the search algorithm's problem representation.

In our DNA object model we create an additional layer of abstraction beyond the traditional GA model where a collection of DNA bases reside within a gene object. In the model, a DNA base is analogous to a specific node in the graph, while a gene is analogous to a node induced subgraph – a district. It is the unique collection of DNA bases that give the gene (district) a set of unique properties or characteristics. A base pair is analogous to the set of links belonging to a node in the graph. The links connecting the nodes in the graph are used as a surrogate for the rules which allow DNA bases to bond and sequence into a strand. In our algorithm, DNA bases are limited to randomly bonding with their neighbors during the replication process, which results in only feasible solutions to highly constrained graphical network problems without the use of inefficient penalty functions. Figure 2 depicts the object model hierarchical relationships.

Figure 3 provides a visual representation of the DNA replication process in our GA. The process begins with two parents contributing complete DNA strands (chromo-

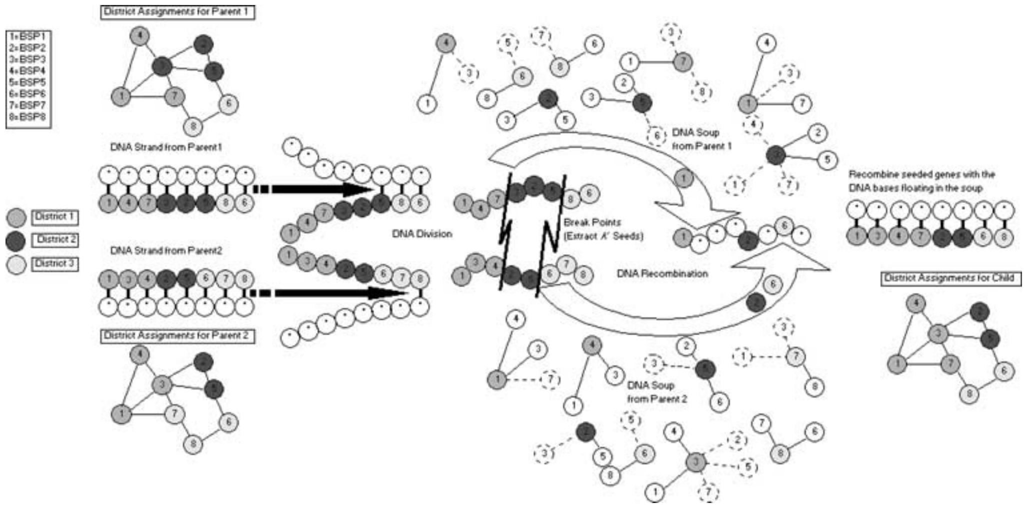

Figure 3. Enforcing contiguity during DNA replication.

somes). Each DNA strand represents an assignment of DNA bases to a gene, which corresponds to an assignment of BSPs to a district. Parent 1 has the following assignments: District_1 = {BSP1, BSP4, BSP7}, District_2 = {BSP2, BSP3, BSP5}, District_3 = {BSP6, BSP8}. Parent 2 has the following assignments: District_1 = {BSP1, BSP3, BSP4}, District_2 = {BSP2, BSP5}, District_3 = {BSP6, BSP7, BSP8}. DNA division in our algorithm is represented by the instantiation of new DNA bases in the DNA soup containing identical information to their parent strands. Thus, each DNA base is aware of its current district assignment, it's connected neighbors (solid line) and its unconnected neighbors (dotted line).

During recombination, a new child chromosome containing K gene objects is instantiated. Each gene in the child strand is randomly seeded with a DNA base from the available DNA objects in the DNA soup. The assignment of the DNA base to a gene in the child strand corresponds to its district assignment in the parent gene. Once K unique seeds have been randomly selected and assigned, the recombination process allows each seed to "attempt" to reconstruct its original parent gene. When a DNA base is assigned, it is removed from the DNA soup of both parent strands. Since the process is random, the complete set of DNA bases may not be available to reconstruct the parent gene entirely. DNA bases that become "stranded" due to the randomness of assignments are placed in a temporary location until all possible DNA nucleotides have been assigned based upon the seed selection. The remaining nodes in the temporary location are then randomly assigned back to a feasible gene (not necessarily the original parent gene).

Note that the "break points" for each parent strand in figure 3 are located between distinct genes, where the genes can contain a variable number of DNA bases. The recombination step in this example takes place as follows: BSP1 is selected from parent 1 as the seed for district 1. BSP2 is selected from parent 2 as the seed for district 2. BSP6 is selected from parent 2 as the seed for district 3. Each gene in the child strand is now ready to “attempt” to reconstruct back to its original configuration. BSP1 in gene 1 examines the soup from parent 1 and adds the available neighbors BSP4 and BSP7. BSP2 in gene 2 examines the soup from parent 2 and adds the available neighbor BSP5. BSP6 in gene 3 examines the soup from parent 2 and adds the available neighbor BSP8. The random process has stranded BSP3 in the DNA soup because all available neighbors have been assigned to completely reconstructed districts. Recall that when a DNA base is assigned, it is removed from the DNA soup of both parents. BSP3 is placed in a temporary location and then randomly assigned to a feasible district – in this case district 1. The new child strand of DNA consists of the following assignments: District_1 = {BSP1, BSP3, BSP4, BSP7}, District_2 = {BSP2, BSP5}, District_3 = {BSP6, BSP8}. Note that it strongly resembles each parent, yet differs slightly due to random recombination.

5.2. The mutation operator

The mutation operator uses a series of neighborhood moves which exchange a single node with a neighboring district. Thus, the mutation operator perturbs nodes that exist on the boundary between two partitions and “anneals” the chromosome toward improving solutions. We refer to the annealing process as “rapid cooling” because it borrows from the search methodology of simulated annealing. Thus, the mutation operator is a greedy-random hill-climbing operator that is designed to optimize the DNA strand of the child chromosome by the simulated annealing process with an accelerated cooling schedule.

6. Parallel Simulated Annealing (PSA)

Parallel Simulated Annealing (PSA) is a parallel implementation of the simulated annealing algorithm described in section 4. It was chosen as a basis for comparison because of its proven effectiveness at solving difficult problems [22]. Similar to EAs designed for MCDM, PSA includes a population of solution vectors that span the solution space. For a detailed description of PSA, the reader is referred to [19].

7. Empirical study of Ghana

The purpose of this empirical study is to solve the EPDP using data from Ghana’s current power transmission configuration and provide a benchmark for performance testing of EPDPs. To this end, we examine and compare the performance of SAGA and PSA in their ability to minimize the functions provided in (9), (10). Through the assistance of specialists at The World Bank, we have determined a reasonable range for the number of districts to study to be $4 \leqslant k \leqslant 6$.

Following a precedent from similar empirical studies of hybrid GAs by Ishibuchi and Murata [17], a population size of 25 solution vectors was used and each procedure was allowed to run for exactly 10,000 function evaluations during each run of the performance test. The control parameters unique to each algorithm were optimized in preliminary investigations prior to the completion of performance testing.

To measure the quality of the final solutions produced by each algorithm, we analyze the difference between SAGA and PSA over 30 independent optimization runs. We use the average number of non-dominated solutions found during the performance tests as a proxy for the reliability of each algorithm. A larger number of non-dominated solutions would represent better coverage of the efficient frontier. 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 $=1$). By this we mean that overall, the final population of solution vectors are located closer to the efficient frontier.

7.1. Non-dominated solution set

Table 2 shows the number of non-dominated solutions produced in each of the 30 runs and the average number of non-dominated solutions produced across all runs. The results show 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.

7.2. Equivalence class layers

Table 3 shows the number of equivalence class layers produced for each of the 30 optimization runs and the average number of equivalence class layers produced across all runs. The results show a statistically significant difference in the average number of equivalence class layers produced by SAGA and PSA in all cases, with SAGA producing fewer layers. Again, fewer equivalence class layers would be preferred to more as this represents a tighter or closer overall fit closer to the efficient frontier.

8. Empirical study using a random test problem generator

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 [5,34]. 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.

Table 2
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
t-value	13.169[a]		6.15[a]		−1.361	

[a]t-values significant at $\alpha = 0.005$.

8.1. 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

Table 3
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
t-value	-10.815^{a}		-12.139^{a}		-14.599^{a}	

[a]t-values significant at $\alpha = 0.005$.

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.

8.2. 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 4 shows the remarkable similarity between the MST solution to the EST problem for Ghana's actual power grid.

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 ($\gamma\colon\ 0 < \gamma \leqslant 1$) 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 Zth nearest neighbor is added to the tree is defined by the geometric distribution $P(Z = z) = (1 - p)^{z-1}p$.

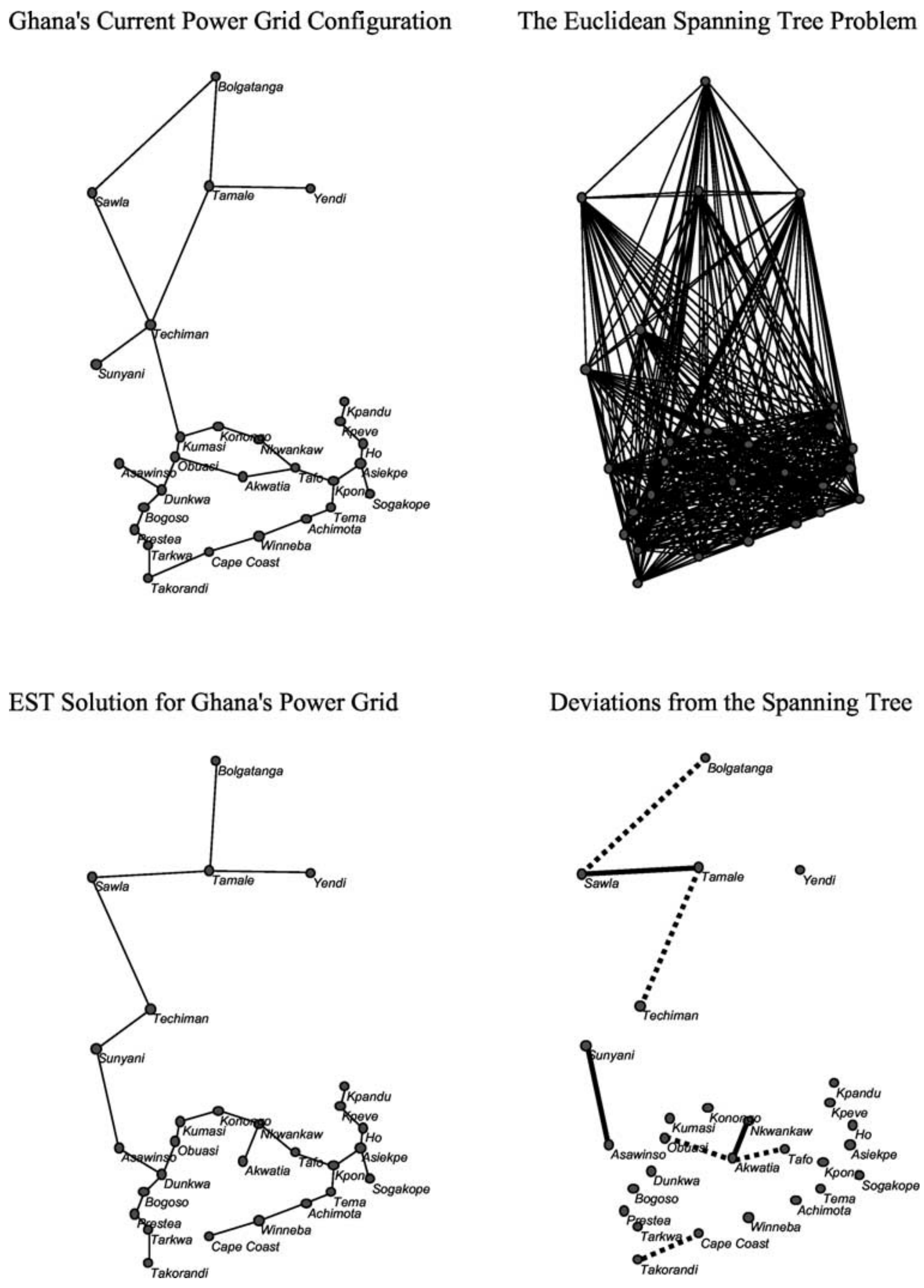

Figure 4. Ghana’s grid vs. the minimal spanning tree.

Once the greedy random spanning tree (GRST) is constructed it is then 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.

8.3. 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.

8.4. Generating a suite of random problems

We used the RPTG to generate a test suite of 30 random problems. We used a setting of $\gamma = 0.9$ 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 $\alpha = 3$ 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.

8.5. 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 this section is to complete a single optimization run using each of the 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 a matched pair for each problem (SAGA vs. PSA). This method is recommended for comparing solution methods with an RTPG [5,34].

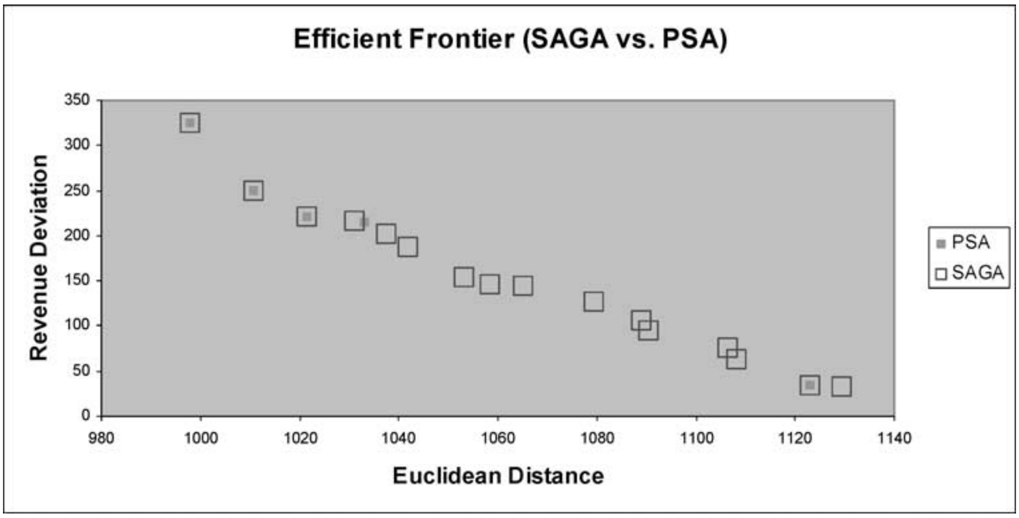

Figure 5. Efficient frontier produced by SAGA and PSA.

The performance measures that we use in this part of the empirical study are the same as those previously described. 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 this part of our empirical study, we have limited our investigation to $k = 5$ districts, because it was the actual scenario requested by the DMs at the World Bank for the Ghana case. Our intent is to provide an example of how our RPTG could be used by other researchers to investigate algorithms designed for solving EPDPs in a controlled manner.

8.6. The non-dominated solution set

Figure 5 shows the first equivalence class layer in the final solution set from the randomly generated set of test problems (problem 2). We note that for the specific test problem shown in figure 5, the number of non-dominated solutions produced by SAGA is substantially greater that those found by PSA. In addition, the coverage of the efficient frontier by SAGA can be characterized as reasonably uniform. 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 the problems in our randomly generated test suite.

Table 4 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 addition, table 4 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 4

Number of non-dominated solutions			Differences in matched samples
Problem	SAGA	PSA	SAGA − PSA
1	6	5	1
2	16	5	11
3	4	3	1
4	3	2	1
5	3	4	−1
6	4	3	1
7	4	4	0
8	6	5	1
9	6	3	3
10	5	4	1
11	8	6	2
12	3	2	1
13	4	3	1
14	4	4	0
15	2	2	0
16	6	2	4
17	4	1	3
18	4	3	1
19	5	6	−1
20	4	4	0
21	5	4	1
22	5	2	3
23	3	4	−1
24	6	3	3
25	8	3	5
26	5	3	2
27	5	3	2
28	6	5	1
29	2	2	0
30	7	3	4
Average	5.100	3.433	1.667
t-value (paired procedure)			3.928[a]

[a]t-values significant at $\alpha = 0.005$.

8.7. Equivalence class layers

Figure 6 shows the results of a single optimization run of SAGA on problem 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. Note that one solution vector in the first series is highlighted with a large red dot.

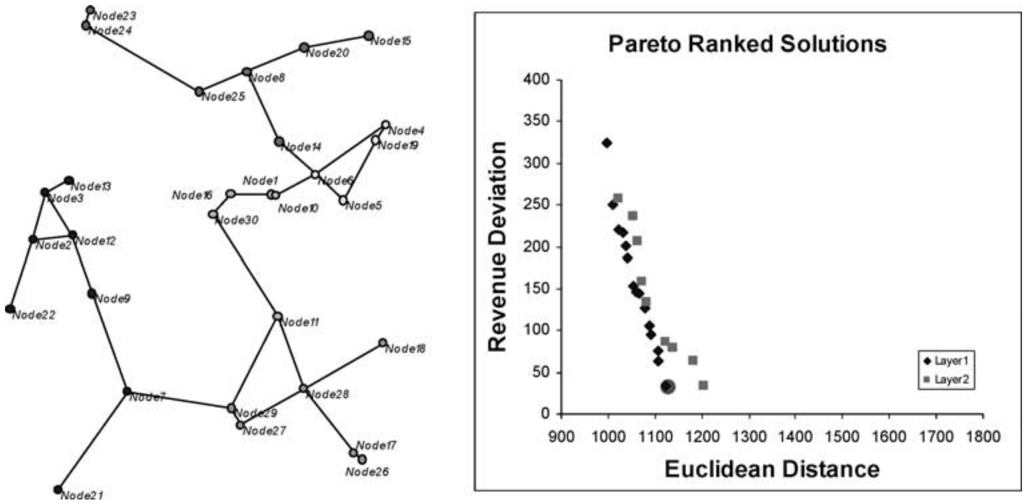

Figure 6. Solution set produced by SAGA (with minimal revenue deviation – left).

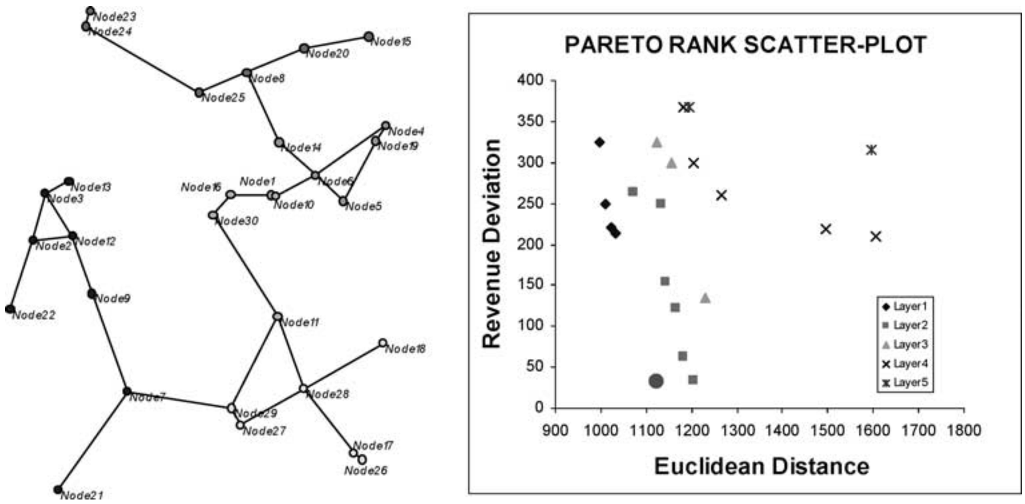

Figure 7. Solution set produced by PSA (with minimal revenue deviation – left).

This is the solution vector which corresponds to the districting plan that is shown in the power grid to the left. The districting plan shown (via the Red Dot) is non-dominated and has the minimum total Revenue Deviation ($f_1 = 32.23$, $f_2 = 1129.43$) relative to all other plans.

Figure 7 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. Clearly, the final set of solution vectors in figure 6 repre-

Table 5

Number of Pareto layers			Difference in matched samples
Problem	SAGA	PSA	(SAGA − PSA)
1	6	5	1
2	2	5	−3
3	6	8	−2
4	6	7	−1
5	6	7	−1
6	6	8	−2
7	6	6	0
8	7	9	−2
9	5	9	−4
10	4	7	−3
11	4	7	−3
12	5	6	−1
13	9	11	−2
14	7	4	3
15	10	11	−1
16	8	10	−2
17	5	9	−4
18	5	5	0
19	6	6	0
20	5	10	−5
21	6	5	1
22	6	8	−2
23	7	6	1
24	5	6	−1
25	6	9	−3
26	9	8	1
27	9	8	1
28	7	7	0
29	5	12	−7
30	6	6	0
Average	6.133	7.5	−1.367
t-value (paired procedure)			−3.549[a]

[a] t-values significant at $\alpha = 0.005$.

sent a tighter fit with the non-dominated set than those in figure 7 and thus, is a higher quality sample of alternatives to present to DMs.

Table 5 provides the number of equivalence class layers generated by each technique for each of the problems in our test suite. In addition, table 5 also gives the variable of interest in the right column (the difference between SAGA and PSA forming a matched sample).

9. Conclusions

With competitive electricity markets emerging throughout the world, it is necessary to develop effective solution methods (systems and algorithms) for designing power districts. This empirical study investigated the performance of SAGA and PSA using actual data taken from Ghana's power distribution system. In addition, we introduced a random test problem generator and provided rationale for its design. We demonstrated the use of the RTPG for a randomly generated test bed of problems and compared the performance of SAGA and PSA. The results show in nearly all cases that there is a statistically significant difference in performance between SAGA and PSA. SAGA outperformed PSA in its ability to locate non-dominated solutions and produce a high quality (soft) efficient frontier. Thus, we conclude that SAGA is well-suited for solving the EPDP.

References

[1] M. Altman, Is automation the answer? – The computational complexity of automated redistricting, Rutgers J. Comput. Technol. Law 23(1) (1995) 81–142.

[2] C. Backstrom, Problems of implementing redistricting, in: *Representation and Redistricting Issues* (1982).

[3] P.K. Bergey, C.T. Ragsdale and M. Hoskote, A decision support system for the electrical power districting problem, Decision Support Systems, to appear.

[4] M.H. Browdy, Simulated annealing: An improved computer model for political redistricting, Yale Law & Policy Rev. 8 (1990) 163.

[5] K.A. De Jong, M.A. Potter and W.M. Spears, Using problem generators to explore the problems with epistasis, in: *Proc. of the International Conference on Genetic Algorithms*, ed. T. Baeck (1997) pp. 333–345.

[6] C. Easingwood, Heuristic approach to selecting sales regions and territories, Oper. Res. Quart. 24(4) (1973) 527–534.

[7] E. El-Farzi and G. Mitra, Solution of set-covering and set-partitioning problems using assignment relaxations, J. Oper. Res. Soc. 43 (1992) 483.

[8] C.M. Fonseca and P.J. Flemming, An overview of evolutionary algorithms in multi-objective optimization, Evolut. Comput. 3(1) (1995) 1–16.

[9] R.S. Garfinkel and G.L. Nemhauser, Optimal political districting by implicit enumeration techniques, Management Sci. 16(8) (1970) B495–B508.

[10] D.E. Goldberg, Genetic algorithms with sharing for multi-modal function optimization, in: *Proceedings of the Second International Conference on Genetic Algorithms* (Cambridge, MA, 1987) pp. 41–49.

[11] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, MA, 1989).

[12] J. Hershberger, Finding tailored partitions, J. Algorithms 12 (1991) 431.

[13] M.S. Heschel, Effective sales territory development, J. Marketing 41(2) (1977) 39–43.

[14] S.W. Hess and S.A. Samuels, Experiences with a sales districting model: criteria and implementation, Management Sci. 18(4) (1971) 41–54.

[15] S.W. Hess, J.B. Weaver, H.J. Siegreldt, J.N. Whelan and P.A. Zitlau, A nonpartisan political redistricting by computer, Oper. Res. 13 (1965) 998–1008.

[16] J. H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, MI, 1975), reissued (MIT Press, Cambridge, MA, 1975).

[17] H. Ishibuchi and T. Murata, A multi-objective genetic local search algorithm and its application to flowshop scheduling, IEEE Trans. Systems Man Cybernet. – Part C: Appl. Rev. 28(3) (1998) 392–403.
[18] S. Kirkpatrick, C.D. Gallatt and M.P. Becchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[19] G. Kliewer and S. Tschoke, A general parallel simulated annealing library and its applications in industry, in: *Proceedings of the First Meeting of the EURO-PAREO Working Group on Parallel Processing in Operations Research* (1998).
[20] J.R. Koza, Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems, Technical Report STAN-CS-90-1314, Computer Science Department, Stanford University (1990).
[21] F. Kursawe, A variant of evolution strategies for vector optimization, in: *Parallel Problem Solving from Nature* (Springer, Berlin, 1991) pp. 193–197, 496.
[22] F.H. Lee and G.S. Stiles, *Parallel Simulated Annealing: Several Approaches, Transputer Research and Applications* (IOS Press, Amsterdam, 1989).
[23] L.M. Lodish, Vaguely right approach to sales force allocations, Harvard Business Rev. (January–February, 1974) 119–124.
[24] A. Mehrotra, E.L. Johnson and G.L. Nemhauser, An optimization based heuristic for political districting, Management Sci. 44(8) (1998) 1100–1114.
[25] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth and A.H. Teller, Equations of state calculation by fast computing machines, J. Chem. Phys. 21(6) (1953) 1087–1092.
[26] S.S. Nagel, Simplified bipartisan computer redistricting, Stanford Law Rev. 17 (1965) 863.
[27] I. Rechenberg, *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipen der Biologischen Evolution*, 2nd ed. (Fromman-Holzboog, Stuttgart, 1993).
[28] C. Reeves, Hybrid genetic algorithms for bin-packing and related problems, Ann. Oper. Res. 63 (1996) 371–396.
[29] C. Reeves, Genetic algorithms for the operations researcher, INFORMS J. Comput. 9(3) (1997) 231–251.
[30] P. Ross, What are genetic algorithms good at?" INFORMS J. Comput. 9(3) (1997) 260–262.
[31] J.D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: *Proceedings of an International Conference on Genetic Algorithms and their Applications* (1985) pp. 93–100.
[32] R.J. Shanker, R.E. Turner and A.A. Zoltners, Sales territory design: An integrated approach, Management Sci. 22(3) (1975) 309–320.
[33] M. Segal and D.B. Weinbergey, Turfing, Oper. Res. 25(3) (1977) 367–386.
[34] W.M. Spears, The role of mutation and recombination in evolutionary algorithms, Doctoral Dissertation Thesis, George Mason University, VA (1998).
[35] R.G. Torricelli and J.I. Porter, Toward the 1980 census: The reapportionment of New Jersey's congressional districts, Rutgers J. Comput. Technol. Law 7(135) (1979).
[36] W. Vickrey, On the prevention of gerrymandering, Political Sci. Quart. 76(105) (1961).
[37] A.A. Zoltners, A unified approach to sales territory alignment, in: *Sales Management: New Developments from Behavioral and Decision Model Research* (Marketing Science Institute, Cambridge, MA, 1979) pp. 360–376.
[38] A.A. Zoltners and P. Sinha, Integer programming models for sales resource allocation, Management Sci. 26(3) (1980) 242–260.