

APPENDIX A

Simulated Annealing Procedure for the EPDP

SIMULATED ANNEALING OPERATORS

The performance of any simulated annealing application is dependent upon the suitability and design of the operators for the problem of interest. The SA procedure does not specifically require a particular operator to be used during the optimization process but recognizes that each must be tailored for its specific application. This section will describe the operators used in our SA procedure to produce solutions to the EPDP.

The Swap Operator

The "swap" operator is a neighborhood operator that is designed to inspect solutions very nearby the current solution in the decision variable space. It is a random neighborhood operator that swaps a single node from a randomly selected district with a neighboring district. More specifically, we define a swap operation as a district relinquishing a bulk supply point (BSP) rather than acquiring a BSP from another district. In doing so, we are able to clearly define the sufficiency conditions for which a swap may take place. Solutions produced using the swap operator are therefore guaranteed to be exactly one step (one degree of separation) away from the current solution. While the swap operator is simple to describe, the constrained graph modeling environment of the EPDP poses considerable overhead in order to maintain contiguity of nodes assigned to districts and ensure that each district remains non-empty. The following are the sufficiency conditions that we have identified for performing a swap operation:

1. The selected district must contain two or more BSPs in order to perform a swap.
2. The BSPs assigned to the selected district that are eligible for swapping must have a transmission line connecting them to a BSP assigned to some other district. We refer to such BSPs as boundary nodes.
3. The selected district must remain contiguous after relinquishing the BSP. Thus, when a BSP is removed from a district, it must be possible to start at any existing BSP in the selected district and reach all other BSPs still assigned to the district via connecting transmission lines. Enforcing this requirement can invoke considerable overhead for some district configurations.

By performing sequential swaps, it is possible to investigate a great number of districting plans in a relatively efficient manner. However, because most EPDPs can be characterized as a sparse network, the swap operator is not sufficient to produce all possible districting plans. Some network configurations can be riddled with traps that will render the swap operator completely ineffective resulting in stagnation of the optimization run. We refer to such configurations as stranded districts. We digress briefly to explain this important issue. Consider the following network configuration for the Republic of Ghana:

FIGURE A.1
Stranded District Example

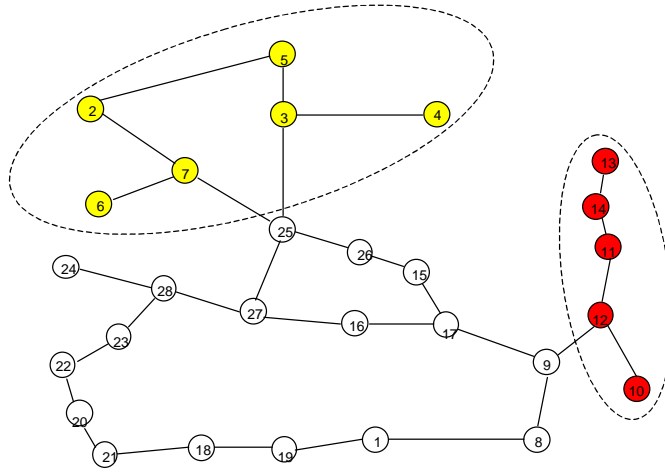


FIGURE A.2
Stranded District Example

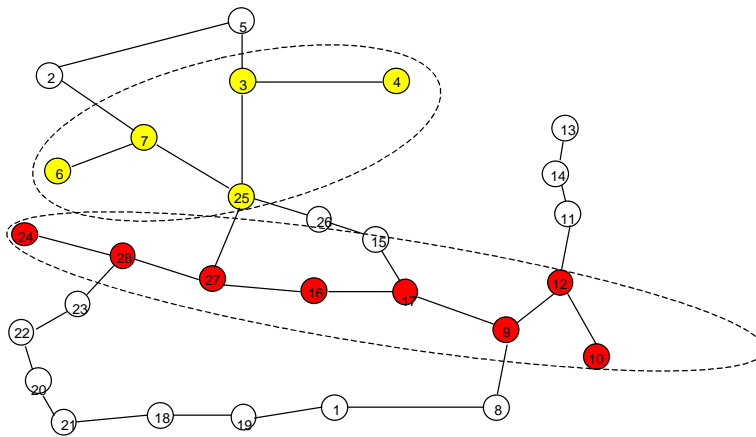


FIGURE A.3
Stranded District Example

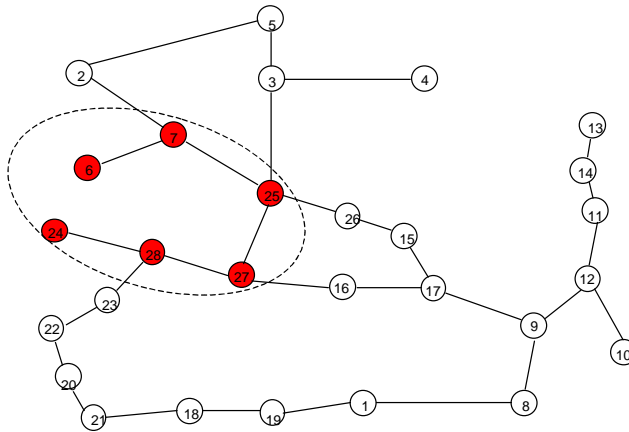
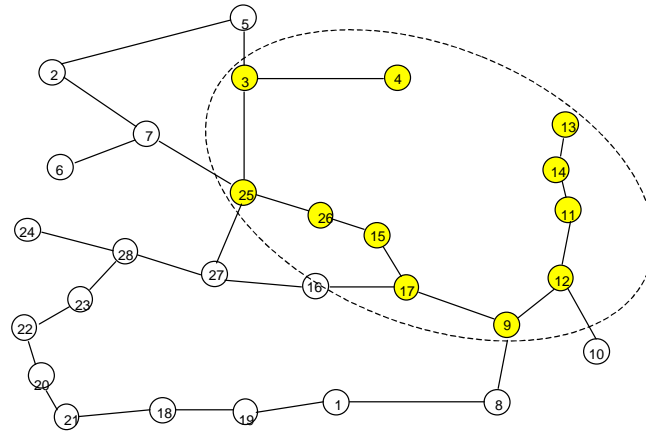


FIGURE A.4
Stranded District Example



For any of the district configurations shown in Figures A.1-A.4 the swap operator will fail to perform a modification to the highlighted district because it would result in the district becoming non-contiguous. The Figures A.1-A.4 above represent only a sample of the stranded district configurations that exist in the Ghana model. There are indeed many more. A noticeable characteristic of stranded districts is that they begin and end with nodes that are connected to the network with only one edge. We refer to these types of nodes as terminal nodes. Indeed, most EPDPs will consist of several terminal nodes because they tend to be sparse networks and thus, any solution algorithm that is to be effective on solving EPDPs must be able to break apart a stranded district in order to perform a thorough search of all possible district combinations. To this end, we introduce a second operator to complement the swap operator.

The Split Operator

While the formation of stranded districts is the exception rather than the rule, we found that such formations do occur and can affect the reliability of the SA procedure in consistently producing high quality solutions. To address this shortcoming, we introduce another random operator that is specifically designed to escape this known pitfall. We refer to it as the "split" operator.

The split operator is designed to split a stranded district such that the resulting district is non-empty and contiguous. It differs from the swap operator in that it will swap as many nodes as necessary to accomplish the task. Thus, unlike the swap operator, which produces only neighboring solutions in the decision variable space, the split operator produces solutions that are several moves away from the current solution in the decision variable space. Such a move can be counterproductive to the annealing process because it can potentially represent a radical change to the current assignment of BSPs. This makes recovery from an "uphill" move more difficult if the uphill move is accepted. However, given the rationale provided above and our experiences with stranded district configurations, one can understand the necessity for this operator.

Stranded districts will occur when all boundary nodes in a district cannot be swapped because such a move would result in a non-contiguous district. We use this as the condition to detect a stranded district and invoke the split operation. The split operation takes place as follows:

1. Randomly select a boundary node from the stranded district and mark the adjacent district. At least one boundary node must exist for each district since the entire network is connected.
2. Remove the boundary node and add it to a list as the target node.
3. Find a neighboring node to the target node that is also assigned to the same district. Remove the neighboring node from the stranded district and add the neighboring node to the list.
4. Check to see if the district is now contiguous. If so, go to 6, else, go to 5.
5. Randomly select another neighboring node to the target node that is also assigned to the same district. Remove the neighboring node from the stranded district and add the neighboring node to the list. Go to 4. If no other neighboring nodes (to the target node) exist in the stranded district, then increment the target node to the next node on the list and go to 3.
6. Add all nodes on the checklist to the adjacent district marked in 1.

Implementing the split procedure guarantees that a stranded district will be partitioned into a non-empty district. Recall that a single node represents a feasible district and thus is the extreme partition that will be invoked by the split operation if the configuration of the stranded district is such that a smaller segment cannot be split apart. A fair question to ask is, "Why not just use the split operation?" The answer to this question is, the swap operator is more constructive than the split operator because it is based upon small incremental improvements. Thus, it is successful much more frequently than the split operator at finding improving solutions. However, if the stranded district is undesirable to the overall districting plan, then the split operation provides a mechanism to break it apart. In addition, the split operator is more sophisticated than the swap operator and thus, requires substantially more computational effort to perform. The split operator can be used as the sole search operator. However, we have found that using both of the operators in conjunction with each other produces better results than using either in isolation. We have found the following heuristic to be very effective on EPDPs:

1. Randomly select a district to be modified.
2. If the selected district is stranded, apply the split operation deterministically.
3. If the selected district is not stranded, apply the split operation with probability of 0.10, else apply the swap operation.

Simulated Annealing Pseudo-Code for the EPDP

The following is pseudo code describes our SA procedure for optimizing the total absolute revenue deviation function provided in (2.1).

SA Procedure for EPDP:

Set $T_c = T_1$ (Initial Temperature)
Set $d = 0.95$ (Cooling Factor)
Set $W = 1/p_a$ (Moving Average Window)

Create initial solution of k instances of $G' (N', E') \subseteq G (N, E)$ in state s

$$\text{evaluate } E(s) = f(x) = \sum_{j=1}^k |R_j - \bar{r}|$$

Do

Do

$a = 0$ ('accepted trial' counter)
 $b = 0$ ('do loop' counter)

Create neighbor in state t of k instances of $G' (N', E') \subseteq G (N, E)$

$$\text{evaluate } E(t) = f(x) = \sum_{j=1}^k |R_j - \bar{r}|$$

$$\Delta E = E(t) - E(s)$$

If $\Delta E < 0$ then

Set $s = t$

$a = a + 1$

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

Set $s = t$

End if

$b = b + 1$

Loop Until $\text{MovingAverage}_{w,a,b} < p_a$

*$T_c = d * T_c$*

Loop Until $T_c \leq T_z$

End Procedure