6-9-1994

Minimum Average Distance (MAD) Partitioning for Grid Graphs

Debra L. Meiers

Follow this and additional works at: https://louis.uah.edu/honors-capstones

Recommended Citation
https://louis.uah.edu/honors-capstones/128

This Thesis is brought to you for free and open access by the Honors College at LOUIS. It has been accepted for inclusion in Honors Capstone Projects and Theses by an authorized administrator of LOUIS.
Minimum Average Distance (MAD) Partitioning for Grid Graphs

Debra L. Meiers

9 June 1994

Dr. Peter J. Slater (Advisor)
Mathematical Sciences Department
University of Alabama in Huntsville
SECTION 1: INTRODUCTION

Due to its wide application, the placement and arrangement of facilities in locations has long been a topic of great interest. Historically, for example, the selection of capital cities within countries has demanded much study and consideration due to the political, economic, and social implications. The importance of facility planning and positioning today is quite evident when one considers the placement of hospitals, fire stations, police stations, etc. within a city. Usually it is in the best interests of all residents of the city for the facilities to be placed in such a way that the distance from residence buildings to service structures is minimal. This desire to construct minimized configurations of buildings and structures has increased considerably in recent years and has formed the foundation for the blossoming field of facilities location. An increasing number of mathematical models and studies are being developed in support of this field. This paper will describe several types of facility location problems that have already been studied and then focus on developing guidelines for the placement of facilities for a special class of problems that can be modeled by grids.

SECTION 2: BACKGROUND

There are several types of facility location problems based on various minimization criteria. For example, in placing an emergency facility such as a hospital, we would want to minimize the response time from the emergency facility to the farthest possible location of an emergency. As such, the minimization criterion for this situation is the longest distance from a facility to a residence building. In placing a post office or grocery store, however, we would be more interested in minimizing the average travel distance for all the residents in the area. In both cases, although we are interested in placing the facilities centrally, the notion of centrality varies due to the minimization criteria. For a treatment of various distance concepts in addition to facility location problems, one can see Distance in Graphs by Buckley and Harary [1].

As introduced by Jordan [5] in 1869, problems centered around minimizing the longest distance from a facility to a non-facility involve finding a "center" node. Consider a connected graph G, and let v be a node of G. The eccentricity of node v, denoted e(v), is defined to be the distance to the node farthest from v. Each node in G has an eccentricity. A node v with the minimum eccentricity among all the nodes in G is
defined to be a center node. The center $C(G)$ of graph $G$ is the set of all center nodes and, hence, consists of the nodes of minimum eccentricity. This concept is illustrated in Figure 1(a). The eccentricity of each node in graph $G$ is labeled; since node $g$ has the lowest eccentricity, it is defined as the center node. Hence, if we were placing a hospital on a site which could be modeled by graph $G$, we would place the hospital at the central node $g$. This central node best meets our minimization criterion. Problems of this type are called 1-Center facility problems since concern lies with finding a single center node for an emergency facility.

As Harary describes in [3], in contrast to 1-Center facility problems, problems which focus on minimizing the average (or, equivalently, the total) distance from a facility to the non-facilities involve finding a median node. Again, consider a connected graph $G$, and let $v$ be a node of $G$. The status of node $v$, denoted $s(v)$, is defined to be the sum of the distances from $v$ to every other node in $G$. Each node in $G$ has a status, and a node $v$ with the minimum status of all the nodes is determined to be a median node. The median $M(G)$ of graph $G$ is the set of all median nodes and, hence, consists of the nodes with minimum status. This idea is illustrated in Figure 1(b). The status of each node in graph $G$ is labeled; since node $I$ has the minimum status, it is defined as the median. Hence, if we were placing a grocery store or bank on a site which could be modeled by graph $G$, we would place the facility at the median $I$. Problems of this type are called 1-Median facility problems since concern lies with finding a single median node for a facility in which the minimization criterion is average distance.

Figure 1: Illustration of 1-Center and 1-Median Problems
To this point, we have examined facility problems in which only one facility is to be placed. There are numerous problems, however, that require locating multiple facilities; many of these are summarized by Tansel, Francis and Lowe in [9]. One such problem, introduced by Hakimi [2], is the p-Center problem for similar facilities. Like the 1-Center problem, the goal in this problem is to minimize the farthest distance from a facility to a non-facility building. It is assumed that the p facilities to be located are indistinguishable; as such, in minimizing the farthest distance, we need only consider the facility closest to a non-facility. For example, if we are to place p hospitals in a city, since the hospitals all serve the same function, we need only minimize the distance from a residence building to a hospital instead of from a residence building to a particular hospital. Figure 2 displays several configurations for a 3-Center problem, with facilities displayed as shaded nodes. The distance from each non-facility node to the nearest facility is labeled. Since the maximum distance from a non-facility node to a facility node is lowest in Figure 2(a), it is apparent that the configuration in Figure 2(a) is the best of the three configurations shown in terms of our minimization criterion. Unlike the 1-Center problem, there is no way to determine the locations for the p facilities by just computing the eccentricities of the nodes. Instead, the multiple facilities to be placed must be evaluated as a group in various trial configurations. Nevertheless, general guidelines have been developed which often rule out many of the most inappropriate candidate configurations.

(a) \[\text{Figure 2: Illustration of 3-Center Problem}\]

Presented by Hakimi in [2], there are also p-Median problems for similar facilities which focus on minimizing the average distance from a non-facility building to any one of the p facilities. Since these
problems too are concerned with similar facilities, the p facilities are assumed to be indistinguishable. Figure 3 displays several configurations for a 3-Median problem, with the shaded nodes representing facilities. By adding up the statuses for the non-facility nodes, we obtain an overall status which allows us to compare different configurations. Since the overall status for Figure 3(a) is lower than those for Figures 3(b) or 3(c), we realize that the configuration in Figure 3(a) is the best of the three configurations shown in terms of our minimization criterion of average distance. As was the case in the p-Center problem, there is no specific way to determine the locations for the p facilities; rather, different trial configurations must be tested and compared. Again, previous research has produced general guidelines for ruling out the most unsuitable configurations.

\[
\begin{align*}
\text{(a)} & \quad \begin{array}{cccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array} & \text{Total} = 6 \\
\text{(b)} & \quad \begin{array}{cccccccc}
3 & 2 & 1 & 1 & 2 & 3 & 1 & 1 \\
\end{array} & \text{Total} = 12 \\
\text{(c)} & \quad \begin{array}{cccccccc}
2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 \\
\end{array} & \text{Total} = 6
\end{align*}
\]

**Figure 3: Illustration of 3-Median Problem**

In addition to the problems mentioned above, a problem that requires locating multiple different facilities in a region was introduced in Hulme and Slater [4] and Slater [8]. Since the facilities are different, a solution cannot be obtained by simply minimizing the distance from a non-facility to a nearest facility; all the different facilities must be taken into account. Such problems, in which p different facilities are to be placed, are referred to as p-Mean Median problems. Figure 4 displays three configurations for a 3-Mean Median problem, with the shaded nodes denoting facilities. Each non-facility node is labeled with its total distance to all of the p different facilities. By adding up the total distances for all of the non-facility nodes, we get an overall total distance which will enable us to compare configurations. Since the total distance for Figure 4(a) and 4(b) is 54 while the total distance for Figure 4(c) is 52, it is apparent that the configuration in Figure 4(c)
is the best of the three configurations shown in terms of the minimization criterion of total distance. Equivalently, we are minimizing the average distance from a facility to a non-facility. This particular type of facility problem with average/total distance as the minimization criteria will be the focus of the body of this paper for the special case where the sites are nodes of a grid.

![Illustration of 3-Mean Median Problem](image)

In obtaining minimized solutions to facility location problems, it is often convenient to begin with a starting configuration and to then move facilities about the graph in a manner which better suits the minimization criterion. One way of accomplishing this moving is by switching an adjacent facility and non-facility node. If facilities are moved about until no adjacent switch will improve the minimization, we say that a locally minimal solution has been reached. Any adjacent switch will only deflect from the optimization goal; for example, if we are considering a locally optimal solution for a p-Median problem, any adjacent switch will cause the average distance to either remain the same or increase. There may be numerous locally optimal solutions which produce different values. Nevertheless, there is only one globally optimum solution value. To reach the globally optimum solution from a locally optimal solution, we might have to make adjacent switches that would at first go against the optimization goal and would appear to be counterintuitive. As such, it is often difficult to discern the globally optimum solution for a graph.

A switching lemma for graphs can be used as a tool to determine when an adjacent switch between a facility and a non-facility node will produce a configuration closer to a locally optimal solution. For multi-facility problems where the facilities are different and the minimization criterion is average distance, we would want to make an adjacent switch if the facility would move closer to more non-facility nodes than it
would move away from. This switching lemma for graphs is illustrated in Figure 5. Graph 5(b) was obtained from Graph 5(a) by switching the highlighted facility and non-facility. Each non-facility node is labeled with its distance to the three facility nodes. As shown, the total sum of the node distances is lower for Graph 5(b) than for Graph 5(a). As such, we see that the switch was a favorable one which brought the configuration closer to a locally optimal solution.

![Figure 5: Illustration of Switching Lemma for Graphs](image)

**SECTION 3: GRIDS**

Several facility location problems have been studied for a restricted class of graphs; examples of such problems can be found in Mitchell [7] and Knor, Niepel, and Soltes [6]. This section will focus on a special class of facility location problems in which the structure upon which the facilities are to be positioned can be modeled by a grid. In this class of problems, the facilities to be placed are different and the minimization criterion is average distance. As such, if we consider the gridlike structure in this class to be a city, interest lies in minimizing the overall distance from each residence building in the city to each of the \( p \) different service structures. Lemmas specific to this problem type will be developed and then used to find globally optimum solutions for the special class of equipartitioned grid problems.
Among other entities, cities often follow a gridlike construct. For example, in Manhattan, the streets run from north to south and from east to west. As such, the layout of Manhattan reveals that it can be easily modeled by a grid. Grids consist of a series of 1x1 square blocks that are organized into rows and columns. A grid $G_{m \times n}$ refers to a group of $m \times n$ blocks that are organized into $m$ rows and $n$ columns. For example, Figure 6(a) displays a grid $G_{2 \times 3}$, which has 2 rows and 3 columns for a total of $2 \times 3 = 6$ blocks. In this model, vertices, as defined earlier in the paper, correspond to the 1x1 square blocks. For our study, we will be interested in the placement of black and white vertices on the grid. The black and white vertices will represent the service structures and residence buildings, respectively. $B$ will refer to the set of black vertices on the grid and $W$ will refer to the set of white vertices. When referring to a specific black vertex, $b$ will be used while $w$ will denote a specific white vertex. A vertex can be referenced by its row and column; for example, a vertex $(2, 3)$ refers to the vertex in row 2 and column 3 of a grid. In Figure 6(a), the set of black vertices $B$ is $\{ (1,1), (1,3), (2,2) \}$ while the set of white vertices $W$ is $\{ (1,2), (2,1), (2,3) \}$.

A secondary method of representing grids will be used in addition to the form shown in Figure 6(a). In this alternate arrangement, a black block is denoted by 1 while a white block is denoted by 0. In addition, grid lines are not used. Figure 6(b) depicts the same grid $G_{2 \times 3}$ as in Figure 6(a), only in the alternative representation. The additional row, denoted the summary row, represents the number of black vertices in each column of the grid. Likewise, the additional right column, denoted the summary column, reflects the number of black vertices in each row of the grid. In general, for an $m \times n$ grid, $b_i$ will represent the number of black vertices in row $i$, $1 \leq i \leq m$, and $a_j$ will represent the number of black vertices in column $j$, $1 \leq j \leq n$. As such, the summary row will hold all the $a_j$ values for $1 \leq i \leq m$, and the summary column will hold all the $b_i$ values for $1 \leq j \leq n$. Figure 7 illustrates this concept.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>w</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w</td>
<td>b</td>
<td>w</td>
<td></td>
</tr>
</tbody>
</table>

(a)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

(b)

Figure 6: Grid Representations
Grids with corresponding numbers of rows and columns can differ due to the placement of black and white vertices. A specific arrangement of black and white vertices within a grid will be defined to be a configuration. The distribution for a grid configuration will be defined by the number of black and white vertices in each column and row of the grid. As an example, Figure 8 depicts two 5x8 grids having the same distribution but different configurations. Since the summary rows and summary columns for the two grids are the same, the grids are said to have the same distribution; they have the same number of black and white vertices in their corresponding rows and columns. Although the grids have the same distribution, their configurations are different since the black and white vertices are arranged differently within the rows and columns.

```
0 1 1 0 0 1 0  4  
0 0 0 1 0 0 1 0  2  
1 1 0 1 0 0 1 0  4  
1 0 1 0 0 0 1 1  3  
1 0 0 1 1 0 1 1  5  
3 2 2 4 1 0 4 2  
```

```
1 0 1 1 0 0 0  4  
0 0 0 0 0 0 1 1  2  
1 1 0 1 0 0 1 0  4  
0 1 0 1 0 0 1 0  3  
1 0 1 1 0 0 1 1  5  
3 2 2 4 1 0 4 2  0  
```

Figure 8: Illustration of Grids With the Same Distribution But Different Configurations

Since we are attempting to minimize the overall distance from each residence building in the city to each of the p different service structures, we will be concerned with the total distance from each residence building, or white vertex, to all of the different service structures, or black vertices. Assume that B-W distance denotes the total distance between all the black vertices and white vertices in the grid. The B-W
distance for a particular grid configuration will also be denoted by TD(B,W), where B and W again represent
the set of black and white vertices in the grid. The distance from a black vertex to a white vertex can be
calculated by adding the vertical distance between the two vertices' rows to the horizontal distance between
the two vertices' columns. As such, TD(B,W) corresponds to adding the horizontal and vertical differences
between all the black and white vertices in the grid. A globally optimum solution for a grid will have the
lowest possible TD(B,W). We will denote TD(B,W) for a globally optimum solution on a grid G to be the
Minimum Total Distance MTD(G, j, k), where j represents the number of black vertices and k the number of
white vertices in grid G.

In pursuing an optimal solution, we will often want to reposition black and white vertices on a grid as
a means of decreasing the B-W distance. As an example, Figure 9 shows two 5x6 grids, in which Grid 2 is
obtained from Grid 1 by switching the adjacent black and white vertices (3,4) and (3,5). The set of black
vertices in Grid 2, B₂, consists of the same set of black vertices as in Grid 1, B₁, only with the addition of the
new black vertex (3,5) and with the removal of the old black vertex (3,4); thus, B₂ = B₁ - (3,4) + (3,5). In the
same way, W₂ = W₁ - (3,5) + (3,4), where W₁ and W₂ represent the set of white vertices in Grid 1 and Grid
2, respectively. As shown in the figure, TD(B₂,W₂) < TD(B₁,W₁), revealing that the adjacent switch of
vertices lowered the B-W distance. The following Switching Lemma for grids tells us precisely when an
adjacent switch between black and white vertices will yield a decrease in the B-W distance.

<table>
<thead>
<tr>
<th>Grid 1</th>
<th>Grid 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD(B₁,W₁) = 764</td>
<td>TD(B₂,W₂) = 760</td>
</tr>
<tr>
<td>1 0 0 1 1 1 4</td>
<td>1 0 0 1 1 1 4</td>
</tr>
<tr>
<td>0 0 1 1 0 1 3</td>
<td>0 0 1 1 0 1 3</td>
</tr>
<tr>
<td>1 0 0 1 0 1 3</td>
<td>1 0 0 0 1 1 3</td>
</tr>
<tr>
<td>1 1 1 0 0 0 3</td>
<td>1 1 1 0 0 0 3</td>
</tr>
<tr>
<td>1 1 1 1 0 1 5</td>
<td>1 1 1 1 0 1 5</td>
</tr>
<tr>
<td>4 2 3 4 1 4 0</td>
<td>4 2 3 3 2 4 0</td>
</tr>
</tbody>
</table>

Figure 9: Illustration of the Switching Lemma for Grids
**Switching Lemma:** Suppose there is a grid $G_{mxn}$ with a B-W configuration producing row distribution $(b_1, b_2, ..., b_m)$ and column distribution $(a_1, a_2, ..., a_n)$ and that a black block at $(j,k)$ is switched with a white block at $(j,k+1)$. The change in TD(B,W) will be:

$$2km-mn-2(a_1+a_2+...+a_k)+2(a_{k+1}+a_{k+2}+...+a_n)+2.$$ 

![Figure 10: Grid $G_{mxn}$ To Be Used in the Switching Lemma Proof](image)

**Proof:**

Consider the black block that is switched. (See Figure 10.) Since $a_i$ represents the number of black blocks in column $i$ and there are $m$ blocks in a column, $(m-a_i)$ represents the number of white blocks in column $i$. Since the black block is moving from column $k$ to column $(k+1)$, the black block is getting closer to all the white blocks in the columns from column $(k+1)$ to column $n$ except for the white block with which it is switching (where the distance remains the same). Hence, the black block is getting closer to $(m-a_{k+1})+(m-a_{k+2})+...+(m-a_n)-1$ white blocks, which can be written alternatively as $m(n-k)-(a_{k+1}+a_{k+2}+...+a_n)-1$ white blocks. It gets farther from all the white blocks in the first $k$ columns. As such, it gets farther from $(m-a_1)+(m-a_2)+...+(m-a_k)$ white blocks, or, alternatively, $km-(a_1+a_2+...+a_k)$ white blocks. Thus, in summary, the distance from the black block increases by one to $km-(a_1+a_2+...+a_k)$ white blocks but also decreases by one to $m(n-k)-(a_{k+1}+a_{k+2}+...+a_n)-1$ white blocks.

Consider the white block that is switched. It gets farther from the black blocks in the columns from column $k+1$ to column $n$. Thus, the white block gets farther from $(a_{k+1}+a_{k+2}+...+a_n)$ black blocks. Since
the white block is getting closer to all the black blocks in the first $k$ columns except for the black block with which it is switching (where the distance remains the same), the white block gets closer to $(a_1+a_2+\ldots+a_k)-1$ black blocks. Thus, the distance from the white block increases by one to $a_{k+1}+a_{k+2}+\ldots+a_n$ black blocks but also decreases by one to $a_1+a_2+\ldots+a_k-1$ black blocks.

Now, adding the distance increases and decreases for the black and white block, it is apparent that the change in $TD(B,W)$ is:

$$km-(a_1+a_2+\ldots+a_k)+a_{k+1}+a_{k+2}+\ldots+a_n = [m(n-k)-(a_{k+1}+a_{k+2}+\ldots+a_n)+1+2(a_1+a_2+\ldots+a_k)]+2$$

$$= 2km-2(a_1+a_2+\ldots+a_k)-2(b_{j+1}+b_{j+2}+\ldots+b_m)+2.$$

The following corollary follows in a similar manner.

**Corollary:**

1. If a white block at $(j,k)$ is switched with a black block at $(j,k+1)$, the change in $TD(B,W)$ will be $mn-2km+2m-2(a_k+a_{k+1}+\ldots+a_n)+2(a_1+a_2+\ldots+a_{k-1})+2$.

2. If a black block at $(j,k)$ is switched vertically with a white block at $(j+1,k)$, the change in $TD(B,W)$ will be $2jn-nm-2(b_1+b_2+\ldots+b_{j-1})+2(b_{j+1}+b_{j+2}+\ldots+b_m)+2$.

3. If a black block at $(j,k)$ is switched vertically with a white block at $(j-1,k)$, the change in $TD(B,W)$ will be $mn-2jn+2n-2(b_j+b_{j+1}+\ldots+b_m)+2(b_1+b_2+\ldots+b_{j-2})+2$.

By applying the Switching Lemma to a pair of adjacent black and white vertices, we can tell whether switching the adjacent vertices will lower $TD(B,W)$. Since the Switching Lemma tells us what the change in $TD(B,W)$ will be if the switch is performed, we know that if the calculated change is negative, we should make the switch since $TD(B,W)$ will be decreasing. Similarly, if the change is positive, we should not make the switch since $TD(B,W)$ will increase. If the change is zero, the switch will not affect $TD(B,W)$.

The following lemma establishes the relationship between the distribution of a grid configuration and $TD(B,W)$. 
**Distribution Lemma:** Suppose there is a grid $G_{m \times n}$ with a B-W configuration producing row distribution $(b_1, b_2, ..., b_m)$ and column distribution $(a_1, a_2, ..., a_n)$. $TD(B,W)$ is given by:

$$\sum_{1 \leq i < j \leq m} [(n - b_i)b_j | j - i|] + \sum_{1 \leq i < j \leq n} [(m - a_i)a_j | j - i|].$$

In particular, $TD(B,W)$ will be the same for any two bicolored grid configurations having the same number of black and white blocks in their corresponding rows and columns.

**Proof:**

$TD(B,W)$ can be found by calculating the distance from all of the white blocks in the grid to all of the black blocks in the grid.

Suppose that a white block "travels" to a black block by first moving in its column from its row to the row of the black block and then moving from its column to the column of the black block. As such, the distance from a white block to a black block is the sum of the vertical distance between the two blocks' rows and the horizontal distance between the two blocks' columns. Thus, the distance from all the white blocks in the grid to all the black blocks in the grid is simply the sum of the vertical distances from the white blocks to the black blocks and the horizontal distances from the white blocks to the black blocks.

Consider a white block in row $i$. (See Figure 11.) To reach a black block in row $j$, the white block must travel a vertical distance of $|j-i|$. However, since there are $b_j$ black blocks in row $j$, the white block must travel a vertical distance of $b_j|j-i|$ to reach all the black blocks in row $j$. Now, since one white block in row $i$
Meiers 13

travels a distance of $b_j l - i l$ to reach all the $b_j$ black blocks in row $j$ and there are $(n-b_j)$ white blocks in row $i$, the vertical distance from all the white blocks in row $i$ to all the black blocks in row $j$ is $(n-b_j)b_j l - i l$. In a similar manner, since there are $(n-b_j)$ white blocks in row $j$ and $b_i$ black blocks in row $i$, the distance from all the white blocks in row $j$ to all the black blocks in row $i$ is $(n-b_j)b_j l - i l = (n-b_j)b_j l - i l$. Thus, the total vertical distance between the white blocks and black blocks in rows $i$ and $j$ is $(n-b_j)b_j l - i l + (n-b_j)b_j l - i l$. As such, to get the total vertical distance from all the white blocks to all the black blocks in the grid, one can calculate the summation of $(n-b_j)b_j l - i l + (n-b_j)b_j l - i l$ for $1 \leq i < j \leq m$.

The vertical distance from all the white blocks to all the black blocks is

$$\sum_{1 \leq i < j \leq m} [(n-b_i)b_j l - i l + (n-b_j)b_i l - i l].$$

Similarly, the horizontal distance from all the white blocks to all the black blocks is

$$\sum_{1 \leq i < j \leq n} [(m-a_i)a_j l - i l + (m-a_j)a_i l - i l].$$

Thus, the total distance from the white blocks to the black blocks is

$$\sum_{1 \leq i < j \leq m} [(n-b_i)b_j l - i l + (n-b_j)b_i l - i l] + \sum_{1 \leq i < j \leq n} [(m-a_i)a_j l - i l + (m-a_j)a_i l - i l].$$

The Switching Lemma and Distribution Lemma can be used in conjunction with each other to find locally minimal solutions to grid problems. A heuristic computer program, based on the two lemmas, has been implemented to help identify locally minimal solutions. Unlike many programs, heuristic programs do not necessarily yield optimum answers; rather, they follow certain guidelines to produce a possibly suboptimum answer. Given the size specifications for a grid, this program implementation generates a random grid configuration and then switches adjacent facility and non-facility nodes when the Switching Lemma reveals that the switch will reduce the total/average distance. The program produces locally optimal solutions in which no switch between an adjacent facility and non-facility node for the final grid configuration will decrease the total distance. By applying the Distribution Lemma to the final solution grid, the total distance for the grid can be calculated. Given grid specifications, the program gives the user the option of
generating a number of random grids, finding locally optimal solutions for them all, and keeping track of both the best and worst configurations found. Figures 12(a) and 12(b) show the best and worst locally optimal configurations found for a 10x10 grid with 16 facilities in which 500 solutions were generated. Although Figure 12(a) shows the best locally optimal configuration found from among the 500 solutions, this solution is not necessarily a globally optimum solution. By running the program again and generating 500 more locally optimal solutions, it is possible that a better locally optimal solution will be found. In fact, Figure 13 shows a 10x10 grid with 16 facilities which has a lower total average distance than the configuration shown in Figure 12(a). In the next section, we will examine a special class of grids for which locally optimal solutions will actually be globally optimum solutions.

<table>
<thead>
<tr>
<th>a) BEST CASE: TD(B, W) = 7696</th>
<th>b) WORST CASE: TD(B, W) = 7902</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 1 1 1 0 0 0 0 3</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 1 1 0 0 0 0 0 5</td>
<td>0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1</td>
</tr>
<tr>
<td>0 0 1 1 1 0 0 0 0 4</td>
<td>0 0 0 0 1 1 0 0 0 2</td>
</tr>
<tr>
<td>0 0 0 0 1 1 0 0 0 0 3</td>
<td>0 0 0 1 1 1 0 0 0 5</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0 0 3</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 2 5 4 4 0 0 0 0</td>
<td>1 1 1 2 3 4 2 1 1 0 0</td>
</tr>
</tbody>
</table>

Figure 12: Best and Worst Locally Minimal Solutions Found Among 500 Solutions

TD(B, W) = 7680

| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 1 1 1 0 0 0 0 4           | 0 0 0 1 1 1 0 0 0 4           |
| 0 0 0 1 1 1 0 0 0 0 4           | 0 0 0 1 1 1 0 0 0 4           |
| 0 0 0 1 1 1 0 0 0 0 4           | 0 0 0 1 1 1 0 0 0 4           |
| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 4 4 4 4 0 0 0 0 0        |

Figure 13: Locally Minimal Solution Which Is Better Than the Best Solution from Figure 12(a)
SECTION 4: EQUIPARTITIONED GRIDS

With the Switching Lemma and Distribution Lemma as tools, we can determine minimized solutions for the special class of equipartitioned grids. The following four theorems present global minimized solutions for equipartitioned grids, that is, where the number of black and white vertices is equal or as close as possible to equal.

Theorem 1: Suppose there is a grid $G_{m\times 2k}$ and that there are $km$ black blocks and $km$ white blocks in the grid, where $m$ can be either even or odd. $TD(B,W)$ will be a minimum only if there are an equal number of black and white blocks in each row of the grid. (Likewise, for a grid $G_{2j\times m}$ with $jm$ black blocks and $jm$ white blocks, $TD(B,W)$ will be a minimum only if there are an equal number of black and white blocks in each column of the grid.)

![Figure 14(a)](image1.png) Figure 14(a): Illustration of the Globally Optimal Distribution for a Grid $G_{mx2k}$

(Figure 14(a) illustrates the globally optimal distribution for a grid $G_{mx2k}$.)

Proof:

Assume a minimum solution and suppose that there are not an equal number of white and black blocks in each row of the grid. It will be shown that switching a black and white block that are adjacent will lessen $TD(B,W)$, contradicting the minimality.
Suppose that there are an equal number of black and white blocks in each row of the grid for the first r rows, but not in row r+1. (See Figure 14(b).) As such, in the (r+1) row, there are either more white blocks than black blocks or more black blocks than white blocks. Suppose that the (r+1) row has more white blocks than black blocks. Now, since in total there are an equal number of white and black blocks in the grid, there must be a row below the (r+1) row that has more black blocks than white blocks. As such, there must be a row below the (r+1) row where there is a black block in the same column as one of the white blocks in the (r+1) row.

Now, this row may or may not be directly below the (r+1) row. Suppose the first row below the (r+1) row which has a black block in the same column as one of the white blocks in the (r+1) row is in the (r+s+1) row. This implies that, in rows (r+1) through (r+s), there was a white block in the same column as each of the white blocks in the (r+1) row. Thus, in rows (r+1) through (r+s), there were more white blocks than black blocks. Suppose that in the s rows from row (r+1) to row (r+s) there are ks+c white blocks, where c>=s>=1. Now, since there are an equal number of black and white blocks in the grid and in the first r rows, there must be an equal number of black and white blocks in the final m-r rows of the grid. As such, since there are ks+c white blocks in the s rows from row (r+1) to row (r+s), there must be kt+c black blocks in the final t=m-(r+s) rows of the grid. Thus, in these s rows from row (r+1) to row (r+s), there are ks - c black blocks and ks + c white blocks. In the final t rows, there are kt + c black blocks and kt - c white blocks.

Suppose that the adjacent black and white block in the same column where the white block is in the (r+s) row and the black block is in the (r+s+1) row are switched.

Consider the first r rows, where there are an equal number of black and white blocks in each row. Since there are kr black blocks in this region and the white block is moving one square away from these black blocks, the distance from the white block to each of the kr black blocks will increase by 1. But, since there are kr white blocks and the black block is moving one square closer to them, the distance to the black block will decrease by 1 for each of the kr white blocks. Since the distance from the black block to the kr white blocks decreases by 1 and the distance from the white block to the kr black blocks increases by 1, the net change in the distance from the white blocks to the black blocks in this region is zero and indicates that, for the blocks in the first r rows, the distance between black and white blocks is not affected by the switch.
Now, consider how the distance changes for the blocks in the remaining \( m-r \) rows. Consider the white block. It gets farther from the \( (ks - c) \) black blocks in the \( s \) rows. There are a total of \( (kt + c) \) black blocks in the bottom \( t \) rows. Since the white block is getting closer to all those blocks except for the black block with which it is switching (where the distance remains the same), the white block gets closer to \( (kt + c - 1) \) black blocks. Thus, in summary, the distance from the white block increases by 1 to \( (ks - c) \) black blocks but decreases by 1 to \( (kt + c - 1) \) black blocks.

Consider the black block. It gets closer to the \( (ks + c) \) white blocks in the \( s \) rows except for the white block with which it is switching (where the distance remains the same). As such, the black block gets closer to \( (ks + c - 1) \) white blocks. Since there are \( (kt - c) \) white blocks in the bottom \( t \) rows, the distance from the black block to each of those \( (kt - c) \) white blocks increases by one. Thus, the distance from the black block increases by 1 to \( (kt - c) \) white blocks but decreases by 1 to \( (ks + c - 1) \) white blocks.

Now, adding the distance increases and decreases for the black and white block,

\[
ks - c + kt - c - [(kt + c - 1) + ks + c - 1] = 2 - 4c < 0 \quad (\text{since } c \geq s \geq 1).
\]

Thus, the overall distance between the black and white blocks in the grid decreases due to the switch. But, since we assumed a minimum solution, we have reached a contradiction.

(The proof for the \( 2j \times m \) grid follows in a similar manner.)

Theorem 2: Suppose there is a grid \( G_{2j \times 2k} \) and that there are \( 2jk \) black blocks and \( 2jk \) white blocks in the grid. \( TD(B,W) \) will be a minimum if and only if there are \( k \) black and \( k \) white blocks in each row and \( j \) black and \( j \) white blocks in each column of the grid.

![Figure 15: Illustration of the Globally Optimal Distribution for a Grid \( G_{2j \times 2k} \)](image-url)
(Figure 15 illustrates the globally optimal distribution for an even x even grid \( G_{2j \times 2k} \).)

**Proof:**

Theorem 1 guarantees that TD(B,W) in a \( m \times 2k \) grid will be a minimum only if there are an equal number of black and white blocks in each row of the grid. Similarly, it also ensures that TD(B,W) in a \( 2j \times m \) grid will be a minimum only if there are an equal number of black and white blocks in each column of the grid. Thus, applying Theorem 1 to the grid \( G_{2j \times 2k} \), we see that TD(B,W) in this grid will be a minimum only if there are an equal number of black and white blocks in each row and in each column of the grid.

The Distribution Lemma guarantees that TD(B,W) will be the same for any \( 2j \times 2k \) grid with this distribution, ensuring that TD(B,W) will be minimum if and only if there are an equal number of black and white blocks in each row and in each column of the grid. Applying the Distribution Lemma to grid \( G_{2j \times 2j} \), we see that in particular

\[
\text{MTD}(G_{2j \times 2j}, 2j^2, 2j^2) = 2 \sum_{1 \leq a < b \leq 2j} (b - a).
\]

**Theorem 3:** Suppose there is a \( 2j+1 \times 2k+1 \) grid and that there are \( 2jk + j + k \) black blocks and \( 2jk + j + k + 1 \) white blocks in the grid. The distance between the black blocks and white blocks will be a minimum if and only if there are \( k \) black blocks and \( k+1 \) white blocks in each odd row, \( k+1 \) black blocks and \( k \) white blocks in each even row, \( j \) black blocks and \( j+1 \) white blocks in each odd column, and \( j+1 \) black blocks and \( j \) white blocks in each even column of the grid. Let this distribution be denoted the "alternating distribution."

![Figure 16(a): Illustration of the Globally Optimal Distribution For a Grid \( G_{2j+1 \times 2k+1} \)](image)

![Figure 16(b): Grid \( G_{2j+1 \times 2k+1} \) To Be Used In the Proof](image)
(Figure 16(a) illustrates the globally optimal distribution for an odd x odd grid $G_{2j+1 \times 2k+1}$.)

**Proof:**

Assume a minimum solution and suppose that the solution does not follow the alternating distribution. It will be shown that switching a black and white block that are adjacent will lessen the total distance between the black blocks and white blocks.

Suppose that the grid follows the alternating distribution in each row of the grid for the first $r$ rows, but not in row $r+1$. (See Figure 16(b).) Since there are an odd number of blocks in each row, there are either more black blocks or more white blocks in row $r+1$.

A. Suppose that the $(r+1)$ row has more white blocks than black blocks. There are two cases depending on whether $r$ is even or odd.

Suppose that $r$ is odd. As such, in row $r$, there are $k$ black blocks and $k+1$ white blocks. Since there is only one more white block than black block in the grid and since an extra white block is already accounted for in the first $r$ odd rows, there must be an equal number of black and white blocks in the grid from row $(r+1)$ to row $(2j+1)$. As such, since row $r+1$ has more white blocks than black blocks, there must be a row below the $(r+1)$ row where there is a black block in the same column as one of the white blocks in the $(r+1)$ row.

Suppose that $r$ is even. As such, in row $r$, there are $k+1$ black blocks and $k$ white blocks. Since row $r+1$ breaks the alternating distribution, there are not $k$ black blocks and $k+1$ white blocks in row $r+1$. Since row $r+1$ breaks with the alternating distribution and since there are more white blocks than black blocks in the row, there must be at least 2 more white blocks than black blocks in row $r+1$; if there were only one extra white block, then row $r+1$ would be reflecting the alternating distribution. Since there is only one more white block than black block in the grid and since there are an equal number of black and white blocks in the first $r$ even rows, there must be one extra white block in the rows from row $r+1$ to row $2j+1$. Since this extra white block is already accounted for in row $r+1$ along with at least one other extra white block, there must be more black blocks than white blocks in row $r+2$ to row $2j+1$. As such, there must be a row below the $(r+1)$ row where there is a black block in the same column as one of the white blocks in the $(r+1)$ row.
We have established that, regardless of whether \( r \) is even or odd, there is a row below the \((r+1)\) row which has a black block in the same column as one of the white blocks in the \((r+1)\) row. Now, this row may or may not be directly below the \((r+1)\) row. Suppose the first row below the \((r+1)\) row which has a black block in the same column as one of the white blocks in the \((r+1)\) row is in the \((r+s+1)\) row. This implies that, in the \( s \) rows from row \((r+1)\) to row \((r+s+1)\), there was a white block in the same column as each of the white blocks in the \((r+1)\) row. As such, we will switch the white block in row \((r+s)\) with the black block in row \((r+s+l)\). Let \( t \) denote the remaining \(2j+1-r-s\) rows in the grid, from row \((r+s+1)\) to row \((2j+1)\).

Now, since \( r + s + t = 2j+1 \), there appear to be four cases: (1) \( r \) is odd, \( s \) is odd, \( t \) is odd; (2) \( r \) is odd, \( s \) is even, \( t \) is even; (3) \( r \) is even, \( s \) is odd, \( t \) is even; (4) \( r \) is even, \( s \) is even, \( t \) is odd.

Let's consider Case 1. Let \( r = 2g+1 \), \( s = 2h+1 \), and \( t = 2i+1 \). Now, since the first \( r \) rows follow the alternating distribution, there are \( k+1 \) white blocks in each odd row and \( k+1 \) black blocks in each even row. Thus, in the \( r = 2g+1 \) rows, there are \((g+1)(k+1)\) white blocks and \((g+1)k + g(k+1)\) black blocks. If the \( s = 2h+1 \) rows were following the alternating distribution, since row \( r+1 \) is an even row, there would be \((h+1)k + h(k+1)\) white blocks and \((h+1)(k+1) + hk\) black blocks. However, since there are extra white blocks in the \( s \) rows, there are \((h+1)k + h(k+1) + c\) white blocks and \((h+1)(k+1) + hk - c\) black blocks in the \( s \) rows.

Similarly, if the \( t = 2i+1 \) rows were following the alternating distribution, since \( r \) and \( s \) are odd and row \( s+1 \) is an odd row, there would be \((i+1)(k+1) + ik\) white blocks and \((i+1)k + i(k+1)\) black blocks in the final \( t \) rows. But, since there are extra black blocks in the \( t \) rows to account for the extra white blocks in the \( s \) rows, there are \((i+1)(k+1) + ik - c\) white blocks and \((i+1)k + i(k+1) + c\) black blocks. These results, along with those of the other cases, are summarized in Table 1.

Considering Case 2, let \( r = 2g+1; s = 2h; \) and \( t = 2i \). Since \( r \) is odd, the number of black and white blocks in the first \( r \) rows is the same as the number in Case 1. Now, if the \( s \) rows were following the alternating distribution, since \( s = 2h \) is even, there would be \( h(k+1) + hk \) white and black blocks. However, since there are extra white blocks in the \( r \) rows, there are \( h(k+1) + hk + c \) white blocks and \( h(k+1) + hk - c \) black blocks. Similarly, since \( t \) is even and there are extra black blocks in the \( t \) rows to account for the extra white blocks in the \( s \) rows, there are \( i(k+1) + ik - c \) white blocks and \( i(k+1) + ik + c \) black blocks in the final \( t \) rows.
For Case 3, let $r=2g$; $s=2h+1$; and $t=2i$. Since the first $r$ rows follow the alternating distribution, there are $k+1$ white blocks in each odd row and $k+1$ black blocks in each even row. Thus, there are $g(k+1) + gk$ white blocks and $g(k+1) + gk$ black blocks in the first $r$ rows. If the $s=2h+1$ rows were following the alternating distribution, since row $r+1$ is odd, there would be $(h+1)(k+1) + hk$ white blocks and $(h+1)k + h(k+1)$ black blocks in the $s$ rows. However, since there are extra white blocks in the $s$ rows, there are actually $(h+1)(k+1) + hk + c$ white blocks and $(h+1)k + h(k+1) - c$ black blocks in the $s$ rows. In a similar manner, since $t$ is even and there are extra black blocks than white blocks in the $t$ rows, there are $i(k+1) + ik - c$ white blocks and $i(k+1) + ik + c$ black blocks in the $t$ rows.

Finally, for Case 4, we have $r=2g$; $s=2h$; and $t=2i+1$. Since $r$ is even, the number of black and white blocks in the first $r$ rows is the same as the number in Case 3. Now, if the $s$ rows were following the alternating distribution, since $s=2h$ is even, there would be $h(k+1) + hk$ white and black blocks. However, since there are extra white blocks in the $s$ rows, there are $h(k+1) + hk + c$ white blocks and $h(k+1) + hk - c$ black blocks. Since $t$ is odd and row $s+1$ is odd, if the $t$ rows were following the alternating distribution, there would be $(i+1)(k+1) + ik$ white blocks and $(i+1)k + i(k+1)$ black blocks. Since there are extra black blocks in the $t$ rows, there are $(i+1)(k+1) + ik - c$ white blocks and $(i+1)k + i(k+1) + c$ black blocks in the final $t$ rows.

Since each row in the $s$ rows has at least one more white block than black block, $c \geq s$.

Suppose that the adjacent black and white block in the same column where the white block is in the $(r+s)$ row and the black block is in the $(r+s+1)$ row are switched. We will consider how the distance changes for the blocks in the rows.

Consider the white block. With the switch, the white block is moving one square away from all the black blocks in the $r$ and $s$ rows. As such, the distance from the white block to each of the black blocks in the $r$ and $s$ rows increases by one. However, it also gets closer to all the black blocks in the $t$ rows except for the black block with which it is switching (where the distance remains the same). Thus, the distance from the white block to each of the black blocks in the $t$ rows except for the one with which it is switching decreases by one. Similarly, the black block gets closer to and, hence, decreases its distance by one to each of the white blocks in the $r$ and $s$ rows except for the one with which it is switching (where the distance
remains the same). It also gets farther from and increases its distance to all of the white blocks in the final t rows by one. To determine the net change in distance caused by the switch, we must simply add the distance increases and decreases for the black and white block being switched.

Since the number of blocks in the r, s, and t rows differs depending on the case being studied, we will have to address the distance change for each case separately.

For Case 1:

The white block gets closer to all of the black blocks in the final t rows except for the one with which it is switching and gets farther from all the black blocks in the r and s rows. Thus, consulting Table 1, we see that for the white block:

White Block: (Distance increases): \( g(k+1) + (g+1)k + (h+1)(k+1) + hk - c \)

(Distance decreases): \( i(k+1) + (i+1)k + c - 1 \)

The black block gets closer to the each of the white blocks in the r and s rows except for the white block with which it is switching (where the distance remains the same). It gets farther from each of the white blocks in the final t rows. Thus, we see that for the black block:

Black Block: (Distance increases): \( (i+1)(k+1) + ik - c \)

(Distance decreases): \( (g+1)(k+1) + gk + (h+1)k + h(k+1) + c - 1 \)

Now, adding the distance increases and decreases for the black and white block,

\[
g(k+1) + (g+1)k + (h+1)(k+1) + hk - c + (i+1)(k+1) + ik - c
\]

\[
- [i(k+1) + (i+1)k + c - 1 + (g+1)(k+1) + gk + (h+1)k + h(k+1) + c - 1]
\]

\[
= gk + g + gk + k + hk + k + h + 1 + hk + c + ik + k + i + 1 + ik - c
\]

\[
- [ik + i + ik + k + c - 1 + gk + k + g + 1 + gk + hk + k + hk + h + c - 1]
\]

\[
= 2gk + 2hk + 2ik + g + 3k + h + 2 + i - 2c
\]

\[
- [2ik + 2gk + 2hk + i + 3k + 2c - 1 + g + h]
\]

\[
= 2 - 2c - [2c - 1] = 3 - 4C < 0 \quad \text{(since } C >= S >= 1)\).

Following the same manner in applying the Switching Lemma for the other three cases, we see that the switch will reduce the B-W distance in all four cases. But, since we assumed a minimum solution, we have reached a contradiction.
B. Suppose that the (r+1) row has more black blocks than white blocks. The proof for this case follows in the same way as in Part A. Although row (r+1) has more black blocks than white blocks, eventually there must be a row below row (r+1) that has more white blocks than black blocks. Suppose that the first row that has a white block in the same column as one of the black blocks in row (r+1) occurs in row (r+s+1). This implies that there is a black block in each column with a black block in row (r+1) for the s rows following row r. As such, we will consider switching the black block in row (r+s) with the white block in row (r+s+1). Again, let t denote the 2j+1-r-s rows from row (r+s+1) to row (2j+1). Just as in Part A, since r + s + t = 2j+1, there appear to be 4 cases: (1) r is odd, s is odd, t is odd; (2) r is odd, s is even, t is even; (3) r is even, s is odd, t is even; (4) r is even, s is even, t is odd. The number of black and white blocks in each section is summarized in Table 2 for all four cases.

Just as in Part A, applying the Switching Lemma to the black block in row (r+s) and to the white block in row (r+s+1) reveals that the switch in all four cases will result in a decrease in B-W distance for the grid. But, since we assumed a minimum solution, we have reached a contradiction, as in Part A.

(Since the grid is 2j+1 X 2k+1, a switch between a black and a white block in the same row but in different columns under the same circumstances will yield the same result in terms of the net TD(B,W) change.)

Thus, any optimum configuration satisfies the alternating distribution for an odd x odd grid, and the Distribution Lemma implies that any two such configurations with the alternating distribution have the same total B-W distance.

**Table 1: Summarization of Block Locations for Part A**

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td># of BB in r</td>
<td>g(k+1) + (g+1)k</td>
<td>g(k+1) + (g+1)k</td>
<td>g(k+1) + gk</td>
<td>g(k+1) + gk</td>
</tr>
<tr>
<td># of WB in r</td>
<td>(g+1)(k+1) + gk</td>
<td>(g+1)(k+1) + gk</td>
<td>g(k+1) + gk</td>
<td>g(k+1) + gk</td>
</tr>
<tr>
<td># of BB in s</td>
<td>(h+1)(k+1) + hk - c</td>
<td>h(k+1) + hk - c</td>
<td>(h+1)(k+1) + h(k+1) - c</td>
<td>h(k+1) + hk - c</td>
</tr>
<tr>
<td># of WB in s</td>
<td>h(k+1)+(h+1)k + c</td>
<td>h(k+1) + hk + c</td>
<td>(h+1)(k+1)+hk + c</td>
<td>h(k+1) + hk + c</td>
</tr>
<tr>
<td># of BB in t</td>
<td>i(k+1)+(i+1)k + c</td>
<td>i(k+1) + ik + c</td>
<td>i(k+1) + ik + c</td>
<td>(i+1)k+(i+1)k + c</td>
</tr>
<tr>
<td># of WB in t</td>
<td>(i+1)(k+1) i + k - c</td>
<td>(i+1)(k+1) i + k - c</td>
<td>(i+1)(k+1) i + k - c</td>
<td>(i+1)(k+1) i + k - c</td>
</tr>
</tbody>
</table>
Table 2: Summarization of Block Locations for Part B

<table>
<thead>
<tr>
<th># of BB in r</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>g(k+1) + (g+1)k</td>
<td>g(k+1) + (g+1)k</td>
<td>g(k+1) + gk</td>
<td>g(k+1) + gk</td>
<td></td>
</tr>
<tr>
<td>(g+1)(k+1) + gk</td>
<td>(g+1)(k+1) + gk</td>
<td>(g+1)(k+1) + gk</td>
<td>(g+1)(k+1) + gk</td>
<td></td>
</tr>
<tr>
<td># of BB in s</td>
<td>(h+1)(k+1) + h + c</td>
<td>h(k+1) + h + c</td>
<td>(h+1)(h+1) + h + c</td>
<td>h(k+1) + h + c</td>
</tr>
<tr>
<td># of WB in s</td>
<td>h(k+1) + h + c</td>
<td>h(k+1) + h + c</td>
<td>(h+1)(h+1) + h + c</td>
<td>h(k+1) + h + c</td>
</tr>
<tr>
<td># of BB in t</td>
<td>i(k+1) + i + c</td>
<td>i(k+1) + i + c</td>
<td>(i+1)(k+1) + i + c</td>
<td>(i+1)(k+1) + i + c</td>
</tr>
<tr>
<td># of WB in t</td>
<td>(i+1)(k+1) + i + c</td>
<td>(i+1)(k+1) + i + c</td>
<td>(i+1)(k+1) + i + c</td>
<td>(i+1)(k+1) + i + c</td>
</tr>
</tbody>
</table>

**Theorem 4:** Suppose there is a $2j+1 \times 2k$ grid $G_{2j+1} \times 2k$ and that there are $k(2j+1)$ black blocks and $k(2j+1)$ white blocks in the grid. $TD(B,W)$ will be a minimum if and only if: (1) there are an equal number of black and white blocks in each row of the grid and (2) there are $2j+1$ black blocks and $2j+1$ white blocks in each pair of columns $2i-1$ and $2i$, for $i = 1$ to $k$ and (3) for consecutive columns $2i-1$ and $2i$, for $i = 1$ to $k$, there are either $j+1$ black blocks and $j$ white blocks in column $2i-1$ and, hence, $j$ black blocks and $j+1$ white blocks in column $2i$, or there are $j$ black blocks and $j+1$ white blocks in column $2i$ or there are $j$ black blocks and $j+1$ white blocks in column $2i-1$ and $j+1$ black blocks and $j$ white blocks in column $2i$.

Figure 17(a): Illustration of the Globally Optimal Distribution For a Grid $G_{2j+1} \times 2k$

(Figure 17(a) illustrates the globally optimal distribution for an odd x odd grid $G_{2j+1 \times 2k}$.)

Figure 17(b): Grid $G_{2j+1} \times 2k$ To Be Used In the Proof
Proof:

Assume we have a configuration whose distribution provides the minimum TD(B,W). By Theorem 1, since there are an even number of columns, we know that the distance between the black and white blocks will be a minimum only if there are an equal number of black and white blocks in each row of the grid. Thus, condition 1 is satisfied and, in considering a minimum solution, we must only focus our attention on the configuration of blocks within the columns.

For condition 2, suppose that there are not 2\(j+1\) black blocks and \(2j+1\) white blocks in each pair of columns \(2i-1\) and \(2i\), for \(i = 1\) to \(k\). It will be shown that switching a black and white block that are adjacent will lessen the total distance between the black blocks and white blocks, a contradiction.

Suppose that there are \(2j+1\) black blocks and \(2j+1\) white blocks in each consecutive pair of columns \(2i-1\) and \(2i\), for \(i = 1\) to \(r\), but not for \(i=r+1\). Without loss of generality, suppose that there are more white blocks than black blocks in the \(2(2j+1)\) blocks in columns \(2r+1\) and \(2r+2\). As such, there are at least \(2j+2\) white blocks in columns \(2r+1\) and \(2r+2\) combined. Suppose that there are more black blocks than white blocks in column \(2r+2\) and, as such, that there are at most \(j\) white blocks in column \(2r+2\). This implies that there are at least \(j+2\) white blocks in column \(2r+1\). But applying the switching lemma to a white block in column \(2r+1\) with an adjacent black block in column \(2r+2\) would imply a decrease in TD(B,W), a contradiction. Hence, we see that there are at least \(j+1\) white blocks in column \(2r+2\). Since in total there are an equal number of white and black blocks in the grid, there must be a column from column \(2r+3\) to column \(2k\) that has more black blocks than white blocks. As such, there must be a column from column \(2r+3\) to column \(2k\) in which there is a black block in the same row as one of the white blocks in the column \(2r+2\). The first such column may or may not be column \(2r+3\).

Suppose the first column after column \(2r+2\) which has a black block in the same row as one of the white blocks in column \(2r+2\) is in column \(2r+s\), where \(s \geq 3\). This implies that, in the \(s-3\) columns strictly between column \(2r+2\) and column \(2r+s\), there was a white block in the same row as each of the white blocks in column \(2r+2\). Thus, since there are at least \(j+1\) white blocks in column \(2r+2\), in each column between column \(2r+2\) and column \(2r+s\), there are more white blocks than black blocks. Applying the switching lemma
to the white block in column 2r+s-1 and the black block in column 2r+s, we see that TD(B,W) will decrease if the blocks are switched, a contradiction. Hence, condition 2 holds.

Assume our minimum solution satisfies conditions 1 and 2 but not condition 3, so that there are not either j+1 black blocks in column 2i-1 and j black blocks in column 2i or j black blocks in column 2i-1 and j+1 black blocks in column 2i, for some i in \{1,2,\ldots,k\}. (See Figure 17(b).) Since condition 2 is satisfied, we do have 2j+1 black blocks and 2j+1 white blocks in each consecutive pair of columns 2i-1 and 2i, for i = 1 to k. Without loss of generality, suppose that column 2d+1 has at least j+2 white blocks, implying that column 2d+2 has at most j-1 white blocks. Since there are a total of 2j+1 white blocks and 2j+1 black blocks in columns 2d+1 and 2d+2 and column 2d+1 has more white blocks than black blocks, there is at least one white block in column 2d+1 which is adjacent to a black block in column 2d+2.

Since there are 2j+1 black blocks and 2j+1 white blocks in consecutive columns 2i-1 and 2i, for i = 1 to k, there are d(2j+1) black blocks and d(2j+1) white blocks in the first 2d columns. Let [2k-(2d+2)] = 2h. As such, there are h(2j+1) white blocks and h(2j+1) black blocks in the final 2h columns. Let a2d+1 represent the number of black blocks in column 2d+1 and let a2d+2 represent the number of black blocks in column 2d+2. Since column 2d+1 has at least j+2 white blocks and column 2d+2 has at most j-1 white blocks, a2d+1 < a2d+2 and a2d+2 - a2d+1 >=3.

Suppose that adjacent black and white blocks in the same row where the white block is in column 2d+1 and the black block is in column 2d+2 are switched. We will consider how the distance changes for the blocks.

The white block gets closer to all of the black blocks in the final 2h+1 columns except for the one with which it is switching and gets farther from all the black blocks in the first 2d+1 columns. Thus, for the white block, the distance increases by one to \(d(2j+1) + a_{2d+1}\) black blocks and the distance decreases by one to \(h(2j+1) + a_{2d+2} - 1\) black blocks.

The black block gets closer to the each of the white blocks in the first 2d+1 columns except for the white block with which it is switching (where the distance remains the same). It gets farther from each of the white blocks in the final 2h+1 columns. Thus, for the black block, the distance increases by one to \(h(2j+1) + (2j+1 - a_{2d+2})\) white blocks and the distance decreases by one to \(d(2j+1) + (2j+1 - a_{2d+1}) - 1\) white blocks.
Now, adding the distance increases and decreases for the black and white block,
\[
d(2j+1) + a_{2d+1} + h(2j+1) + (2j+1 - a_{2d+2}) \\
- [h(2j-1) + a_{2d+2} - 1 + d(2j+1) + (2j+1 - a_{2d+1}) - 1] \\
= 2a_{2d+1} - 2a_{2d+2} + 2 \\
\leq -6 + 2 < 0 \quad \text{(since } a_{2d+2} - a_{2d+1} \geq 3).\]

As such, it is seen that the overall distance between the black and white blocks in the grid decreases due to the switch. But, since we assumed a minimum solution, we have reached a contradiction. Therefore, we see that condition 3 holds.

Thus, any optimum configuration satisfies conditions (1), (2), and (3), and the Distribution Lemma implies that any two such configurations with the distribution outlined by the three conditions have the same TD(B,W).

SECTION 5: CONCLUSION

In this paper, useful tools and guidelines have been developed to help find locally optimal as well as globally optimum solutions to certain gridlike facility location problems. The Switching Lemma and Distribution Lemma can be used for all grid facility problems in identifying locally optimal solutions. The Switching Lemma identifies when a switch of adjacent black and white vertices will yield an improvement in the overall B-W distance and, hence, bring the configuration closer to a locally optimal solution, in which no adjacent switch between any two black and white vertices will lower the B-W distance. The Distribution Lemma assures us that grids with the same distribution but different configurations have the same B-W distance.

With the use of these tools, it has been shown that locally optimal solutions for the equipartitioned grid problems are actually globally optimum solutions. As such, guidelines for optimally placing facilities on a gridlike structure in which the number of facility and residence buildings are equal (or near equal for the odd by odd case) have been developed.

Despite these successful results, there are still a number of problems left to be considered even within the special class of grid location problems alone. Foremost in this area, guidelines for producing
globally optimum solutions for non-equipartitioned grids are desired. It is hoped that such guidelines will be developed in the future and that they can be extended to wider classes of graphs.
References


APPENDIX
This program, written in Turbo Pascal, implements the Switching and Distribution Lemmas. The user is prompted for grid specifications, which include the number of rows and columns in the grid as well as the number of black blocks to be placed. The user may enter the desired coordinates of the black blocks; if the user elects instead to accept default values, the program generates a random grid with the given specifications. The user is also given the option to run the program multiple times, recording the best and worst locally optimal solutions found. The program uses the Switching Lemma to determine when a switch between an adjacent black and white block will yield an improvement in B-W distance. Adjacent black blocks and white blocks are switched until a locally optimal solution is reached, in which a switch between any adjacent black and white block will only increase TD(B,W) or keep it the same. Once a locally optimal solution is reached, the Distribution Lemma is used to calculate TD(B,W). Once the final solution(s) have been found, the results are written to an output file.

PROGRAM Minimize_Grid

VAR
  Grid : Grid_Type;
  Best : Grid_Type;
  Worst : Grid_Type;
  Coordinates : Coord_Ray;
  OrigCoord : Coord_Ray;
  Rows, Columns : Integer;
  Num_Blacks : Integer;
  Num_Times : Integer;

TYPE
  Grid_Type = ARRAY[0..Nx,0..Nx] OF Integer;
  Ray = ARRAY[1..2500] OF Integer;
  Coord_Ray = ARRAY[1..2500,1..2] OF Integer;

CONST
  Nx = 50;
  Black = 1; (* Represents a facility node or black block *)
  White = 0; (* Represents a non-facility node or white block *)
  Max_Times = 500; (* Holds the maximum number of grids to be processed. *)
  Max_Int = 2147483647;

PROGRAM Minimize_Grid (Input,Output);

i : Integer;
Highest : LongInt;
Lowest : LongInt;
Current : LongInt;
Answer : Char;
Default : Char;
Outfile : Text;

(*************************************************************************************************)
PROCEDURE Initialize_Grid (M : Integer;
   N : Integer;
   VAR Grid : Grid_Type);

VAR
   i, j : Integer;

BEGIN (* Initialize_Grid *)
   FOR i := 1 TO M+1 DO
      BEGIN
         FOR j := 1 TO N+1 DO
            Grid[i,j] := White
      END;
   END; (* Initialize_Grid *)

(*************************************************************************************************)
PROCEDURE ArrayCopy ( VAR OldGrid : Grid_Type;
   VAR NewGrid : Grid_Type;
   M : Integer;
   N : Integer);

VAR
   i, j: Integer;

BEGIN (* ArrayCopy *)
   FOR i := 1 TO M DO
      BEGIN
         FOR j := 1 TO N DO
            NewGrid[i,j] := OldGrid[i,j];
      END;
   END; (* ArrayCopy *)

(*************************************************************************************************)
PROCEDURE CoordCopy ( VAR OldGrid : Coord_Ray;
   VAR NewGrid : Coord_Ray;
   M : Integer;
   N : Integer);

VAR
   i, j: Integer;

BEGIN (* CoordCopy *)
   FOR i := 1 TO M DO
      BEGIN
         FOR j := 1 TO N DO
            NewGrid[i,j] := OldGrid[i,j];
      END;
   END; (* CoordCopy *)
BEGIN (* CoordCopy *)
FOR i := 1 TO M DO
BEGIN
FOR j := 1 TO N DO
   NewGrid[i,j] := OldGrid[i,j];
END;
END; (* CoordCopy *)

FUNCTION Check_Coord ( VAR Coord : Coord-Ray;
                        Num_Coord: Integer;
                        M, N : Integer): Boolean;
VAR
   Done: Boolean;
   i : Integer;
BEGIN (* Check_Coord *)
   Check_Coord := True;
   IF Num_Coord <> 0 THEN
      BEGIN
         i := 1;
         Done := False;
         WHILE (i <= Num_Coord) AND (NOT Done) DO
            BEGIN
               IF (Coord[i,1] = M) AND (Coord[i,2] = N) THEN
                  BEGIN
                     Check_Coord := False;
                     Done := True;
                  END
               ELSE
                  i := i + 1
               END;
         END;
   END;
END; (* Check_Coord *)

PROCEDURE Get_Coord ( VAR Coord : Coord-Ray;
                       M, N : Integer;
                       Num_Blks : Integer);
VAR
   Coord1,
   Coord2,
   Num_Entered : Integer;
BEGIN (* Get_Coord *)
   Num_Entered := 0;
   Writeln('Please enter each coordinate corresponding to a black block');
   Writeln('on a separate line. Enter the row first, followed by a space.');
WHILE NumEntered < NumBlks DO
    BEGIN
        Read(Coord1);
        Readln(Coord2);
        IF (Coord1 <= M) AND (Coord2 <= N) THEN
            BEGIN
                IF Check_Coord(Coord,NumEntered,Coord1,Coord2) = True THEN
                    BEGIN
                        NumEntered := NumEntered + 1;
                        Coord[NumEntered,1] := Coord1;
                        Coord[NumEntered,2] := Coord2;
                    END
                ELSE
                    BEGIN
                        Write('This coordinate has already been entered. Enter ');
                        Writeln(' another coordinate.);
                    END;
            END
        ELSE
            Writeln('This coordinate is invalid. Enter another coordinate.');
        END;
    END; (* Get_Coord *)

PROCEDURE Permu ( VAR PMR: Ray; 
P : Integer);
    VAR
        I, L, T, K: Integer;
    BEGIN (* Permu *)
        Randomize;
        FOR I := 1 TO P DO
            PMR[I] := I;
        FOR I := 1 TO P - 1 DO
            BEGIN
                L := P - I + 1;
                K := RANDOM(L) + I;
                T := PMR[I];
                PMR[I] := PMR[K];
                PMR[K] := T;
            END;
    END; (* Permu *)

PROCEDURE Store_Coordinate (VAR PEM : Coord_Ray; 
    VAR PMR : Ray; 
P : Integer; 
    N : Integer);
VAR
  K, I, J, T, H: Integer;
BEGIN (* Coordinate *)
  FOR I := 1 TO P DO
  BEGIN
    T := PMR[I];
    K := T DIV N;
    J := T - (K * N);
    IF J = 0 THEN
      BEGIN
        J := N;
        H := K;
      END
    ELSE
      H := K + 1;
      PEM[I,1] := H;
      PEM[I,2] := J;
    END;
  END; (* Coordinate *)

-----------------------------------------------------------------------

PROCEDURE Place_Blacks ( VAR Coord : Coord-Ray;
  Number : Integer;
  M     : Integer;
  N     : Integer);
VAR
  PMR : Ray;
  i   : Integer
BEGIN (* Place_Blacks *)
  Permu(PMR, M*N);
  Store_Coordinate(Coord, PMR, Number, N);
END; (* Place_Blacks *)

-----------------------------------------------------------------------

PROCEDURE Update_Count ( VAR Grid : Grid_Type;
  VAR Coord : Coord-Ray;
  Number: Integer;
  M     : Integer;
  N     : Integer);
VAR
  i, j, k : Integer;
BEGIN (* Update_Count *)
  FOR i := 1 TO Number DO
    Grid[Coord[i,1], Coord[i,2]] := Black;
  FOR i := 1 TO M DO
    BEGIN
      k := 0;
      FOR j := 1 TO N DO
        Grid[i,j] := Black;
      END;
    END;
END; (* Update_Count *)
FOR \( j := 1 \) TO \( N \) DO
\[
\begin{align*}
  & \left. \begin{array}{l}
  k := k + \text{Grid} \[i,j]\;
  \text{Grid} \[i,j+1] := k;
  \end{array} \right. \\
  \end{align*}
\]
END;
FOR \( i := 1 \) TO \( N \) DO
BEGIN
\[
\begin{align*}
  & \left. \begin{array}{l}
  k := 0;
  \text{FOR} \ j := 1 \ \text{TO} \ M \ \text{DO}
  \ \ \ \begin{array}{l}
  k := k + \text{Grid} \[i,j];
  \text{Grid} \[i,j+1] \ := k;
  \end{array}
  \end{array} \right. \\
  \end{align*}
\]
END;
END; (* Update_Count *)

FUNCTION Total_Count ( VAR Grid : Grid_Type;
M : Integer;
N : Integer); LongInt;
VAR
  i, j: Integer;
  HCount, VCount, Temp1, Temp2: LongInt;
BEGIN (* Total_Count *)
  HCount := 0;
  VCount := 0;
  FOR \( i := 1 \) TO \( (m-1) \) DO
    BEGIN
      FOR \( j := (i+1) \) TO \( m \) DO
        BEGIN
          Temp1 := \text{Grid} \[i,n+1] * (n-\text{Grid} \[j,n+1]) * ABS(j-i);
          Temp2 := \text{Grid} \[j,n+1] * (n-\text{Grid} \[i,n+1]) * ABS(j-i);
          VCount := VCount + Temp1 + Temp2
        END;
    END;
  END;
  FOR \( i := 1 \) TO \( (n-1) \) DO
    BEGIN
      FOR \( j := (i+1) \) TO \( n \) DO
        BEGIN
          Temp1 := \text{Grid} \[m+1,i] * (m-\text{Grid} \[m+1,j]) * ABS(j-i);
          Temp2 := \text{Grid} \[m+1,j] * (m-\text{Grid} \[m+1,i]) * ABS(j-i);
          HCount := HCount + Temp1 + Temp2
        END;
    END;
  Total_Count := HCount + VCount;
END; (* Total_Count *)

PROCEDURE Display_Grid ( VAR Grid : Grid_Type;
M : Integer;
N : Integer;
Count : Integer;
VAR FileOut : Text);
VAR
    i, j: Integer;

BEGIN (* Display_Grid *)
    // Code for displaying the grid

    PROCEDURE Switch ( blocknum : Integer;
                      directn : Integer;
                      VAR Coord : Coord-Ray;
                      VAR Grid : Grid-Type);

    BEGIN (* Switch *)
        // Code for switching
    END; (* Switch *)

    FUNCTION MakeSwitch ( directn : Integer;
                          VAR Grid : Grid_Type;
                          J,K : Integer;
                          M,N : Integer):Boolean;

    VAR
        // Additional variables

BEGIN (* MakeSwitch *)
CASE directn of
  3: BEGIN
    IF (J <> M) AND (Grid[j+1,k] <> 1) THEN
      BEGIN
        T1 := 2*J*N - N*M;
        T2 := 0;
        FOR i := 1 TO J DO
          T2 := T2 + Grid[i,n+1];
        T3 := 0;
        FOR i := (J+1) TO M DO
          T3 := T3 + Grid[i,n+1];
        Change := T1 - 2*T2 + 2*T3 + 2;
      END;
    ELSE
      Change := 1
    END;
  2: BEGIN
    IF (K <> N) AND (Grid[j,k+1] <> 1) THEN
      BEGIN
        T1 := 2*K*M - M*N;
        T2 := 0;
        FOR i := 1 TO K DO
          T2 := T2 + Grid[M+1,i];
        T3 := 0;
        FOR i := (K+1) TO N DO
          T3 := T3 + Grid[M+1,i];
        Change := T1 - 2*T2 + 2*T3 + 2;
      END;
    ELSE
      Change := 1
    END;
  1: BEGIN
    IF (J <> 1) AND (Grid[j-1,k] <> 1) THEN
      BEGIN
        T1 := M*N - 2*J*N + 2*N;
        T2 := 0;
        FOR i := J TO M DO
          T2 := T2 + Grid[i,N+1];
        T3 := 0;
        FOR i := 1 TO (J-1) DO
          T3 := T3 + Grid[i,N+1];
        Change := T1 - 2*T2 + 2*T3 + 2;
      END;
    ELSE
      Change := 1
    END;
  4: BEGIN
    IF (K <> 1) AND (Grid[j,k-1] <> 1) THEN
      BEGIN
        T1 := M*N - 2*K*M +2*M;
        T2 := 0;
        FOR i := K TO N DO
          T2 := T2 + Grid[M+1,i];
T3 := 0;
FOR i := 1 TO (K-1) DO
  T3 := T3 + Grid[M+1,i];
  Change := T1 - 2*T2 + 2*T3 + 2;
END
ELSE
  Change := 1;
END;
IF Change < 0
  THEN
    MakeSwitch := True
  ELSE
    MakeSwitch := False;
END; (* MakeSwitch *)

PROCEDURE Move-Blocks ( VAR Coord : Coord-Ray;
                         VAR Grid : Grid-Type;
                         Tot_Blacks : Integer;
                         M, N : Integer);

VAR
  Num_Checked, Block_No,
  Direction: Integer;

BEGIN (* Move-Blocks *)
  Num_Checked := 0;
  Block_No := 1;
  WHILE Num_Checked <= Tot_Blacks DO
    BEGIN
      direction := 1;
      WHILE direction <= 4 DO
        BEGIN
          IF MakeSwitch(direction,Grid,Coord[block_no,1],
                         Coord[block_no,2],M,N) = True
          THEN
            BEGIN
              Num_Checked := 0;
              Switch(Block_No,direction,Coord,Grid);
              Update_Count(Grid,Coord, Tot_Blacks, M, N);
              direction := 1;
            END
          ELSE
            END
        ELSE
          direction := direction + 1;
        END; (* While *)
      Num_Checked := Num_Checked + 1;
      IF block_no <> Tot_Blacks THEN
        block_no := block_no + 1
      ELSE
        block_no := 1;
      END; (* While *)
  END; (* Move-Blocks *)
BEGIN (* Minimize_Grid *)
  Assign(Outfile, 'c:\tp\schoollma499\output.pas');
  Rewrite(Outfile);
  Writeln('Please enter the number of rows for the grid (<= ',Nx-1,').); Readln(Rows);
  Writeln('Please enter the number of columns for the grid (<= ',Nx-1,').); Readln(Columns);
  Writeln('Please enter the number of black blocks to be placed on the grid.
If you enter 0, the number of black and white blocks will be 
"as equally divided as possible."); Readln(Num_Blacks);
  IF Num_Blacks = 0 THEN Num_Blacks := (Rows * Columns) DIV 2;
  Writeln('Would you like to use default values? [Y/N]); Readln(Default);
  IF UPCASE(Default) = 'N' THEN Get_Coord(OrigCoord, Rows, Columns, Num_Blacks);
  Writeln('Would you like to run the program more than once? [Y/N]); Readln(Answer);
  IF UPCASE(Answer) = 'Y' THEN BEGIN
    Num_Times := Max_Times;
    Highest := 0;
    Lowest := Max_Int;
  END ELSE BEGIN
    Num_Times := 1;
    Highest := Max_Int;
    Lowest := 0;
  END;
  FOR i := 1 TO Num_Times DO BEGIN
    Initialize_Grid(Rows,Columns,Grid);
    IF UPCASE(Default) = 'N' THEN CoordCopy(OrigCoord, Coordinates, Num_Blacks, 2)
    ELSE Place_Blacks(Coordinates,Num_Blacks,Rows,Columns);
    Update_Count(Grid,Coordinates,Num_Blacks,Rows,Columns);
    Move_Blacks(Coordinates,Grid,Num_Blacks,Rows,Columns);
    Current := Total_Count(Grid,Rows,Columns);
    IF Current < Lowest THEN BEGIN
      ArrayCopy(Grid, Best, Rows+1, Columns+1);
      Lowest := Current;
    END;
    IF Current > Highest THEN BEGIN
      ArrayCopy(Grid, Worst, Rows+1, Columns+1);
      Highest := Current;
    END;
  END;
IF Num_Times = 1 THEN
    Display_Grid(Grid, Rows+1, Columns+1, Num_Blacks, Outfile)
ELSE
    BEGIN
        Writeln(Outfile, 'BEST CASE: ');
        Display_Grid(Best, Rows+1, Columns+1, Num_Blacks, Outfile);
        Writeln(Outfile);
        Writeln(Outfile, 'WORST CASE: ');
        Display_Grid(Worst, Rows+1, Columns+1, Num_Blacks, Outfile);
    END;
END.
(* Minimize_Grid *)