Optimization of Production and Distribution

Forecasting Capacity for Projected Growth

Mark Bacon, Michael Breed
Management Summary

Brinker International currently owns numerous concepts including Corner Bakery, Maggiano’s, and Macaroni Grill. The various stores suffer from a mom-and-pop mentality, relying on experience and intuition rather than quantitative data analysis. There is an opportunity to optimize process in an attempt to meet projected growth.

Given the three-year growth projections for these three concepts, we have created a production and distribution model to analyze existing capacity and determine a need for future production capability. We created our model using an integer program. This allowed us to determine that Brinker cannot meet their projected growth under the current conditions. Additionally, running the models under various scenarios using maximum capacity constraints, Brinker is still unable to meet the projected growth. However, optimizing distribution and allowing plants to ship to other regions makes it possible to fill some capacity gaps and accommodate some growth.
Optimization of Production and Distribution for

Brinker International

Mark Bacon

Michael Breed

EMIS 4395 – Senior Design

May 7, 2002
Current Situation

Brinker International is a multi-billion dollar company, owning several concepts and hundreds of stores. Because of the process of acquiring concepts, the business operations of the various concepts tend to vary. The distributions, supply chain management, and production facilities all need to be linked and in-line to maximize economies and profits. The individual stores tend to operate in a mom-and-pop store mentality relying on experience and intuition to estimate inventory needs and distribution routes. The three concepts we analyzed, Corner Bakery, Maggiano’s, and Macaroni Grill, ship their goods from three major production plants located in Santa Ana, California, Dallas, and Chicago. Currently the stores are serviced by the plant closest to them. Although the processes may not be optimal at the most basic level, the company and its many concepts continue to thrive even in the weakened economy. Brinker has plans to continue growth into other regions of the country. Before this can happen though, the current situation and capacity for growth must be assessed to gain a better idea of the kind of investment it will take and how they can facilitate some growth by optimizing processes before the new facilities become ready for use.

Analysis of the Situation

The current system is simple and allows for the many stores to be serviced without much thought or planning. However, because of the varying production costs of the three plants as well as the varying costs of the distribution lines, there may be opportunity to optimize the distribution and production process and save money. The situation that is of the most concern to Brinker is the question of capacity for growth. The production plants are operating comfortably within their limits, and are having no
problem meeting the needs of the stores they service. However, with growth on the horizon, these capacities will be tested and it is of vital importance that we know the capacity capabilities before stores can be added. It is likely that some amount of investment will be involved to support the projected growth, but optimizing the process and distribution will allow them to support growth incrementally while the investment of equipment and facilities is being established. Looking at the possibility of shipping from any plant to satisfy capacity in any region will allow Brinker to use the current capacity to the fullest and make it easy to facilitate steady growth until increased production capacity is available.

**Objectives**

The first objective of the problem was to optimize the distribution routes to the various plants. This needs to be done in an attempt to utilize maximum existing capacities if the projected growth is to be met. From these results we will need to analyze the potential scenarios to see what is successful in increasing capacities. The results from the various scenarios will lend itself to cost benefit analysis to see what is the most cost effective way of attempting to meet the capacity demands of the projected growth. Finally we will be able to recommend any action that may need to be taken as well as where an investment needs to be made.

**Problem Formulation**

The formulation of the problem and gathering of the data was by far the most difficult and time-consuming part of the project. The pieces of data were often hard to find, needed to be estimated, or were a result of several other pieces of the data. An
understanding of the problem was gained through several meetings, and network sketches.

The problem was divided into two basic parts, distribution and production. We had to establish arc costs between each feasible distribution route. Much of the complexity of the project was a result of the numerous distribution routes and their varying costs. To account for this we established a single arc cost between facilities by estimating the most probable distribution cost. Another part of the problem was the varying capacities, costs, and production capabilities of each plant. From the data we had gathered we were able to establish production capacities, shipping requirements, and production costs for each of the three concepts from each of the three plants. Production capacities were measured in terms of available resources to service the greatest number of concepts for one year. Shipping capacities were measured in total number of cases required to service one concept for one year. Production costs were easily established from previously gathered data that we were given. As the problem was being formulated, it became very clear that the problem lend itself to a basic integer programming model. AMPL was chosen because it was the most familiar to us. After receiving the three year projections for the growth of the three concepts we were able to start modeling and running the program.

Technical Description of the Model & Scenarios

We ran four scenarios with the same basic model that we created in AMPL. All the scenarios kept the three-year growth projections for the number of stores as constant. The first was with the current conditions and restrictions on production capabilities and current hours the plants were running. (All results for the four scenarios can be found in
Appendix 1) The second scenario was run with the maximum increase in production hours that the plant could operate under. There were two plants running under their production capabilities. We increased their workweek form 5, 20-hour days to 6, 20-hour days. This scenario also required an investment in additional freezers in the Santa Ana and Chicago plants. The third scenario added a minor equipment investment to the existing plants with the current production schedule. We calculated the maximum plant capacity by determining the maximum pounds per square foot that could be produce. The final scenario we ran with our AMPL model took into account both the increased production schedule and the maximum equipment investment in the existing plants.

A fully annotated set of variables, constraints, and objectives can be found in Appendix C.

Results

In each of the scenarios of projected growth there were a number of restaurants that could not be serviced. Under each scenario plants operate at near %100 capacity. In order to maximize capacities, restaurants were often serviced by plants outside their region. This of significance because such a solution to maximize use of capacity is not currently being utilized.

Under the current conditions with projected growth 104 restaurants were left unserviced. The total cost of this scenario is $3.76 million and services 142 stores.

The scenario of current capacities with an increased production schedule leaves 92 unserviced restaurants at a cost of $4.6 million. $240,000 of this cost is due to additional freezers in two of the plants. The freezers must be added to account the increased production.
The third scenario was run under maximum additional equipment investment in the current plants. This scenario serviced all but 56 restaurants, at a cost of $5.7 million plus an additional undisclosed equipment investment.

Implementing both the increased production schedule and the additional equipment investment left only 37 restaurants unserviced. The total cost amounted to $6.8 million plus the undisclosed equipment investment.

Detailed quantitative results can be found in Appendix A.

**Conclusions/Recommendations**

It is clear from the results that an investment in new production facilities will be needed to facilitate the kind of growth that is desired. However, the results show that varying distribution routes were used to maximize use of current capacity, which will allow for limited growth without a major investment. The first recommendation is to increase the production schedule of each plant. It is the most cost/effective, and is usually free. However, in this case it will require the minor investment in the additional freezers. It is always a good idea to stretch your fixed assets before you invest in new ones. There may also be opportunity to optimize the current process by streamlining the current production process in the plants and reduce waste. The second and most encouraged recommendation is to begin cost benefit analysis of building new production facilities. We see this as a necessity to accommodate the projected and desired growth.
Appendix A
Brinker Capacity and Growth Capabilities
Model Results

Current Conditions

<table>
<thead>
<tr>
<th>Cost (thousands)</th>
<th>3,758</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity Used %</td>
<td></td>
</tr>
<tr>
<td>Santa Ana</td>
<td>98.7</td>
</tr>
<tr>
<td>Dallas</td>
<td>99.8</td>
</tr>
<tr>
<td>Chicago</td>
<td>99.8</td>
</tr>
<tr>
<td>Unserviced Restaurant</td>
<td>104</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CB</th>
<th>Mg</th>
<th>Mac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Santa Ana</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Dallas</td>
<td>27</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Chicago</td>
<td>53</td>
<td>12</td>
<td>0</td>
</tr>
</tbody>
</table>
Brinker Capacity and Growth Capabilities
Model Results

Increase to 6/20 hr days

<table>
<thead>
<tr>
<th>Cost (thousands)</th>
<th>4,342</th>
</tr>
</thead>
<tbody>
<tr>
<td>120 freezer in Santa Ana</td>
<td></td>
</tr>
<tr>
<td>120 freezer in Chicago</td>
<td></td>
</tr>
<tr>
<td>4,582</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capacity Used %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Santa Ana</td>
</tr>
<tr>
<td>Dallas</td>
</tr>
<tr>
<td>Chicago</td>
</tr>
</tbody>
</table>

Unserved Restaurants 92

<table>
<thead>
<tr>
<th></th>
<th>CB</th>
<th>Mg</th>
<th>Mac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Santa Ana</td>
<td>7</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Dallas</td>
<td>28</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Chicago</td>
<td>41</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>
## Brinker Capacity and Growth Capabilities
### Model Results

### Additional Equipment

<table>
<thead>
<tr>
<th>Cost (thousands)</th>
<th>5,719</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equipment Investment</td>
<td>1,800</td>
</tr>
<tr>
<td>Total Cost</td>
<td>7,519</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capacity Used %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Santa Ana</td>
</tr>
<tr>
<td>Dallas</td>
</tr>
<tr>
<td>Chicago</td>
</tr>
</tbody>
</table>

| Unserved Restaurants | 56 |

<table>
<thead>
<tr>
<th>CB</th>
<th>Mg</th>
<th>Mac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Santa Ana</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Dallas</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>Chicago</td>
<td>13</td>
<td>12</td>
</tr>
</tbody>
</table>
Brinker Capacity and Growth Capabilities
Model Results

Additional Equipment and Hours

<table>
<thead>
<tr>
<th>Cost (thousands)</th>
<th>6,603</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equipment Investment</td>
<td>1,800</td>
</tr>
<tr>
<td>freezer in Santa Ana</td>
<td>120</td>
</tr>
<tr>
<td>freezer in Chicago</td>
<td>120</td>
</tr>
<tr>
<td>Total</td>
<td>8,643</td>
</tr>
</tbody>
</table>

Capacity Used %

<table>
<thead>
<tr>
<th>Location</th>
<th>Used %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Santa Ana</td>
<td>99.7</td>
</tr>
<tr>
<td>Dallas</td>
<td>99.8</td>
</tr>
<tr>
<td>Chicago</td>
<td>99.2</td>
</tr>
</tbody>
</table>

Unserviced Restaurants

<table>
<thead>
<tr>
<th>Restaurant</th>
<th>CB</th>
<th>Mg</th>
<th>Mac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Santa Ana</td>
<td>7</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Dallas</td>
<td>13</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Chicago</td>
<td>0</td>
<td>12</td>
<td>0</td>
</tr>
</tbody>
</table>
Brinker Capacity and Growth Capabilities
Model Results

Scenario Comparisons

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Stores Serviced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>220</td>
</tr>
<tr>
<td>2</td>
<td>210</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>190</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Stores Serviced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>220</td>
</tr>
<tr>
<td>2</td>
<td>210</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>190</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Stores Serviced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>220</td>
</tr>
<tr>
<td>2</td>
<td>210</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>190</td>
</tr>
</tbody>
</table>

Cost per Serviced Store

Cost (thousands) 3,758 4,342 7,519 8,643

Unserviced Stores 104 92 56 37

Serviced Stores 142 154 190 209

Cost per Serviced Store 26.46479 28.19481 39.57368 41.35407

Cost per Add'l Store Serviced 48.66667 88.25 59.15789

Cost per Serviced Store

Stores Serviced

Number of Stores

Scenario
Appendix B
option solver cplex;

var ca_to_ca_cb >= 0 integer;
var ca_to_d_cb >= 0 integer;
var ca_to_ca_mg >= 0 integer;
var ca_to_d_mg >= 0 integer;
var ca_to_ca_mac >= 0 integer;
var ca_to_d_mac >= 0 integer;
var d_to_d_cb >= 0 integer;
var d_to_ca_cb >= 0 integer;
var d_to_ch_cb >= 0 integer;
var d_to_d_mg >= 0 integer;
var d_to_ca_mg >= 0 integer;
var d_to_d_mac >= 0 integer;
var d_to_ca_mac >= 0 integer;
var ch_to_ch_cb >= 0 integer;
var ch_to_d_cb >= 0 integer;
var ch_to_d_mg >= 0 integer;
var ch_to_d_mac >= 0 integer;
var diff_ca_cb >= 0 integer;
var diff_ca_mg >= 0 integer;
var diff_ca_mac >= 0 integer;
var diff_d_cb >= 0 integer;
var diff_d_mg >= 0 integer;
var diff_d_mac >= 0 integer;
var diff_ch_cb >= 0 integer;
var diff_ch_mg >= 0 integer;
var diff_ch_mac >= 0 integer;
var diff_ch_mac >= 0 integer;

minimize difference:
\[ \text{diff}_{ca_cb} + \text{diff}_{ca_mg} + \text{diff}_{ca_mac} + \text{diff}_{d_cb} + \text{diff}_{d_mg} + \text{diff}_{d_mac} + \text{diff}_{ch_cb} + \text{diff}_{ch_mg} + \text{diff}_{ch_mac}. \]

minimize cost:
\[ 13959.92 \times \text{ca}_{to_{ca}}_{cb} + 13959.92 \times \text{ca}_{to_{d}}_{cb} + 22100 \times \text{ca}_{to_{d}}_{cb} + 15623.4 \times \text{ca}_{to_{ca}}_{mg} + 15623.4 \times \text{ca}_{to_{d}}_{mg} + 25500 \times \text{ca}_{to_{d}}_{mg} + 9456.2 \times \text{ca}_{to_{ca}}_{mac} + 9456.2 \times \text{ca}_{to_{d}}_{mac} + 11900 \times \text{ca}_{to_{d}}_{mac} + 098.52 \times \text{d}_{to_{d}}_{cb} + 33098.52 \times \text{d}_{to_{ca}}_{cb} + 33098.52 \times \text{d}_{to_{ch}}_{cb} + 10400 \times \text{d}_{to_{ch}}_{cb} + 042.72 \times \text{d}_{to_{d}}_{mg} + 37042.72 \times \text{d}_{to_{ca}}_{mg} + 25500 \times \text{d}_{to_{ca}}_{mg} + 37042.72 \times \text{d}_{to_{ch}}_{mg} + 12000 \times \text{d}_{to_{ch}}_{mg} + 419.8 \times \text{d}_{to_{d}}_{mac} + 22419.8 \times \text{d}_{to_{ca}}_{mac} + 11900 \times \text{d}_{to_{ca}}_{mac} + 22419.8 \times \text{d}_{to_{ch}}_{mac} + 5600 \times \text{d}_{to_{ch}}_{mac} + 1786.8 \times \text{ch}_{to_{ch}}_{cb} + 51786.8 \times \text{ch}_{to_{d}}_{cb} + 10400 \times \text{ch}_{to_{d}}_{cb} + 57958.16 \times \text{ch}_{to_{ch}}_{mg} + 57958.16 \times \text{ch}_{to_{d}}_{mg} + 12000 \times \text{ch}_{to_{d}}_{mg} + 35078.68 \times \text{ch}_{to_{ch}}_{mac} + 35078.68 \times \text{ch}_{to_{d}}_{mac} + 5600 \times \text{ch}_{to_{d}}_{mac}; \]

subject to Capacity_Calif:
\[ 4503200 \geq \text{ca}_{to_{ca}}_{cb} \times 225160 + \text{ca}_{to_{d}}_{cb} \times 225160 + \text{ca}_{to_{ca}}_{mg} \times 251992 + \text{ca}_{to_{d}}_{mg} \times \]
251992 + ca_to_ca_mac
* 152516 + ca_to_d_mac * 152516;

subject to Capacity_Dallas:
8556080 >= d_to_d_cb * 225160 + d_to_ca_cb * 225160 + d_to_ch_cb * 225160 + d_to_d_mg * 2519
92 + d_to_ca_mg *
251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152516 + d_to_ca_mac * 152516 + d_to_ch_mac * 15
2516;

subject to Capacity_Chicago:
9006400 >= ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_to_ch_mg * 251992 + ch_to_d_mg *
251992 + ch_to_ch_mac
* 152516 + ch_to_d_mac * 152516;

subject to Num_Calif_cb:
ca_to_ca_cb + d_to_ca_cb = 7 - diff_ca_cb;

subject to Num_Calif_mg:
ca_to_ca_mg + d_to_ca_mg = 2 - diff_ca_mg;

subject to Num_Calif_mac:
ca_to_ca_mac + d_to_ca_mac = 22 - diff_ca_mac;

subject to Num_Dallas_cb:
d_to_d_cb + ca_to_d_cb + ch_to_d_cb = 28 - diff_d_cb;

subject to Num_Dallas_mg:
d_to_d_mg + ca_to_d_mg + ch_to_d_mg = 3 - diff_d_mg;

subject to Num_Dallas_mac:
d_to_d_mac + ca_to_d_mac + ch_to_d_mac = 45 - diff_d_mac;

subject to Num_Chicago_cb:
ch_to_ch_cb + d_to_ch_cb = 55 - diff_ch_cb;

subject to Num_Chicago_mg:
ch_to_ch_mg + d_to_ch_mg = 12 - diff_ch_mg;

subject to Num_Chicago_mac:
ch_to_ch_mac + d_to_ch_mac = 72 - diff_ch_mac;

problem model:
difference,
ca_to_ca_cb,
ca_to_d_cb,
ca_to_ca_mg,
ca_to_d_mg,
ca_to_ca_mac,
ca_to_d_mac,
d_to_d_cb,
d_to_ca_cb,
d_to_ch_cb,
d_to_d_mg,
d_to_ca_mg,
d_to_ch_mg,
d_to_d_mac,
d_to_ca_mac,
d_to_ch_mac,
ch_to_ch_cb,
ch_to_d_cb,
ch_to_ch_mg,
ch_to_d_mg,
ch_to_ch_mac,
ch_to_d_mac,
diff_ca_cb,
diff_ca_mg,
diff_ca_mac,
diff_d_cb,
diff_d_mg,
diff_d_mac,
diff_ch_cb,
diff_ch_mg,
diff_ch_mac,
Capacity_Calif,
Capacity_Dallas,
Capacity_Chicago,
Num_Calif_cb,
Num_Calif_mg,
Num_Calif_mac,
Num_Dallas_cb,
Num_Dallas_mg,
Num_Dallas_mac,
Num_Chicago_cb,
Num_Chicago_mg,
Num_Chicago_mac;

problem model2:
  cost,
  ca_to_ca_cb,
  ca_to_d_cb,
  ca_to_ca_mg,
  ca_to_d_mg,
  ca_to_ca_mac,
  ca_to_d_mac,
d_to_d_cb,
d_to_ca_cb,
d_to_ch_cb,
d_to_d_mg,
d_to_ca_mg,
d_to_ch_mg,
d_to_d_mac,
d_to_ca_mac,
d_to_ch_mac,
ch_to_ch_cb,
ch_to_d_cb,
ch_to_ch_mg,
ch_to_d_mg,
ch_to_ch_mac,
ch_to_d_mac,
Capacity_Calif,
Capacity_Dallas,
Capacity_Chicago,
Num_Calif_cb,
Num_Calif_mg,
Num_Calif_mac,
Num_Dallas_cb,
Num_Dallas_mg,
Num_Dallas_mac,
Num_Chicago_cb,
Num_Chicago_mg,
Num_Chicago_mac;

problem model1;
solve;
printf"diff_ca_cb = %3.0f\n",diff_ca_cb;;
printf"diff_ca_mg = %3.0f\n",diff_ca_mg;;
printf"diff_ca_mac = %3.0f\n",diff_ca_mac;;
printf"diff_d_cb = %3.0f\n",diff_d_cb;;
printf"diff_d_mg = %3.0f\n",diff_d_mg;;
printf"diff_ch_cb = %3.0f\n",diff_ch_cb;;
printf"diff_ch_mg, = %3.0f\n",diff_ch_mg;;
printf"diff_ch_mac = %3.0f\n",diff_ch_mac;;

fix diff_ca_cb;
fix diff_ca_mg;
fix diff_ca_mac;

fix diff_d_cb;
fix diff_d_mg;
fix diff_d_mac;
fix diff_ch_cb;
fix diff_ch_mg;
fix diff_ch_mac;

problem model2;
solve;
printf"ca_to_ca_cb = %3.0f\n",ca_to_ca_cb;;
printf"ca_to_d_cb = %3.0f\n",ca_to_d_cb;;
printf"ca_to_ca_mg = %3.0f\n",ca_to_ca_mg;;
printf"ca_to_d_mg = %3.0f\n",ca_to_d_mg;;
printf"ca_to_ca_mac = %3.0f\n",ca_to_ca_mac;;
printf"ca_to_d_mac = %3.0f\n",ca_to_d_mac;;
printf"d_to_d_cb = %3.0f\n",d_to_d_cb;;
printf"d_to_ca_cb = %3.0f\n",d_to_ca_cb;;
printf"d_to_ch_cb = %3.0f\n",d_to_ch_cb;;
printf"d_to_d_mg = %3.0f\n",d_to_d_mg;;
printf"d_to_ca_mg = %3.0f\n",d_to_ca_mg;;
printf"d_to_ch_mg = %3.0f\n",d_to_ch_mg;;
printf"d_to_d_mac = %3.0f\n",d_to_d_mac;;
printf"d_to_ca_mac = %3.0f\n",d_to_ca_mac;;
printf"d_to_ch_mac = %3.0f\n",d_to_ch_mac;;
printf"ch_to_ch_cb = %3.0f\n",ch_to_ch_cb;;
printf"ch_to_ch_cb = %3.0f\n",ch_to_d_cb;;
printf"ch_to_ch_mg = %3.0f\n",ch_to_ch_mg;;
printf"ch_to_d_mg = %3.0f\n",ch_to_d_mg;;
printf"ch_to_ch_mac = %3.0f\n",ch_to_ch_mac;;
printf"ch_to_d_mac = %3.0f\n",ch_to_d_mac;;

printf"Capacity Used in California = %3.0f\n", ca_to_ca_cb * 225160 + ca_to_d_cb * 225160 +
    ca_to_d_mg * 251992 + ca_to_ca_mac * 152516 + ca_to_d_mac * 152516;;
printf"Capacity Remaining in California = %3.0f\n",4503200-(ca_to_ca_cb * 225160 + ca_to_d_c
    b * 225160 + ca_to_ca_mg * 251992 + ca_to_ca_mac * 152516 + ca_to_d_mac * 152516);;
printf"Capacity Used in Dallas = %3.0f\n", d_to_d_cb * 225160 + d_to_ca_cb * 225160 + d_to_c
    h_cb * 225160 + d_to_d_mg * 251992 + d_to_ca_mg * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152
    516 + d_to_ch_mac * 152516; ;
printf"Capacity Remaining in Dallas = %3.0f\n",8556080 - (d_to_d_cb * 225160 + d_to_ca_cb * 
    225160 + d_to_ch_cb * 225160 + d_to_d_mg * 251992 + d_to_ca_mg * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152
    516 + d_to_ch_mac * 152516); ;
printf"Capacity Used in Chicago = %3.0f\n", ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_
    to_ch_mg * 251992 + ch_to_d_mg * 251992 + ch_to_ch_mac * 152516 + ch_to_d_mac * 152516; ;
printf"Capacity Remaining in Chicago = %3.0f\n",9006400 - (ch_to_ch_cb * 225160 + ch_to_d_c
    b * 225160 + ch_to_ch_mg * 251992 + ch_to_d_mg * 251992 + ch_to_ch_mac * 152516 + ch_to_d_mac 
    * 152516); ;

printf"Califomia CB's Serviced = %3.0f\n", ca_to_ca_cb + d_to_ca_cb;;
printf"California MG's Serviced = %3.0f\n", ca_to_ca_mg + d_to_ca_mg;;
printf"California Mac's Serviced = %3.0f\n", ca_to_ca_mac + d_to_ca_mac;;

printf"Dallas CB's Serviced = %3.0f\n", d_to_d_cb + ch_to_d_cb + ca_to_d_cb;;
printf"Dallas MG's Serviced = %3.0f\n", d_to_d_mg + ch_to_d_mg + ca_to_d_mg;;
printf"Dallas Mac's Serviced = %3.0f\n", d_to_d_mac + ch_to_d_mac + ca_to_d_mac;;

printf"Chicago CB's Serviced = %3.0f\n", ch_to_ch_cb + d_to_ch_cb;;
printf"Chicago MG's Serviced = %3.0f\n", ch_to_ch_mg + d_to_ch_mg;;
printf"Chicago Mac's Serviced = %3.0f\n", ch_to_ch_mac + d_to_ch_mac;;
option solver cplex;
var ca_to_ca_cb >= 0 integer;
var ca_to_d_cb >= 0 integer;
var ca_to_ca_mg >= 0 integer;
var ca_to_d_mg >= 0 integer;
var ca_to_ca_mac >= 0 integer;
var ca_to_d_mac >= 0 integer;
var d_to_d_cb >= 0 integer;
var d_to_ca_cb >= 0 integer;
var d_to_ch_cb >= 0 integer;
var d_to_d_mg >= 0 integer;
var d_to_ca_mg >= 0 integer;
var d_to_ch_mg >= 0 integer;
var d_to_d_mac >= 0 integer;
var d_to_ca_mac >= 0 integer;
var d_to_ch_mac >= 0 integer;
var ch_to_ch_cb >= 0 integer;
var ch_to_d_cb >= 0 integer;
var ch_to_ch_mg >= 0 integer;
var ch_to_d_mg >= 0 integer;
var ch_to_ch_mac >= 0 integer;
var ch_to_d_mac >= 0 integer;
var diff_cacb >= 0 integer;
var diff_ca__mg > 0 integer;
var diff_ca_mac > 0 integer;
var diff_cacb >= 0 integer;
var diff_ca__mg > 0 integer;
var diff_ca_mac > 0 integer;
var diff_cacb >= 0 integer;
var diff_ca__mg > 0 integer;
var diff_ca_mac > 0 integer;
var diff_cacb >= 0 integer;
var diff_ca__mg > 0 integer;
var diff_ca_mac > 0 integer;
minimize difference:

   diff_ca__cb+diff_ca__mg+diff_ca_mac+diff_d__cb+diff_d__mg+diff_d_mac+diff__ch__cb+diff__ch__mg+diff__ch__mac;

minimize cost:

   13959.92 * ca_to_ca_cb + 13959.92 * ca_to_d_cb + 22100 * ca_to_d_cb + 15623.4 * ca_to_ca_mg 
   + 15623.4 * ca_to_d_mg +
   25500 * ca_to_d_mg + 9456.2 * ca_to_ca_mac + 9456.2 * ca_to_d_mac + 11900 * ca_to_d_mac + 33098.52 * d_to_d_cb +
   33098.52 * d_to_ca_cb + 22100 * d_to_ca_cb + 33098.52 * d_to_ch_cb + 10400 * d_to_ch_cb + 37042.72 * d_to_d_mg +
   37042.72 * d_to_ca_mg + 25500 * d_to_ca_mg + 37042.72 * d_to_ch_mg + 12000 * d_to_ch_mg + 22419.8 * d_to_d_mac +
   22419.8 * d_to_ca_mac + 11900 * d_to_ca_mac + 22419.8 * d_to_ch_mac + 5600 * d_to_ch_mac + 51786.8 * ch_to_d_cb +
   51786.8 * ch_to_d_cb + 10400 * ch_to_d_cb + 57958.16 * ch_to_ch_mg + 57958.16 * ch_to_d_mg +
   12000 * ch_to_d_mg +
   35078.68 * ch_to_ch_mac + 35078.68 * ch_to_d_mac + 5600 * ch_to_d_mac;

subject to Capacity_Calif:

   5403840 >= ca_to_ca_cb + ca_to_d_cb + ca_to_ca_mg + ca_to_d_mg +

Page 1
251992 + ca_to_ca_mac
* 152516 + ca_to_d_mac * 152516;

subject to Capacity_Dallas:
8556080 >= d_to_d_cb * 225160 + d_to_ca_cb * 225160 + d_to_ch_cb * 225160 + d_to_d_mg * 251992 + d_to_ca_mg
251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152516 + d_to_ca_mac * 152516 + d_to_ch_mac * 152516;

subject to Capacity_Chicago:
10807680 >= ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_to_ch_mg * 251992 + ch_to_d_mg
251992 + ch_to_ch_mac * 152516 + ch_to_d_mac * 152516;

subject to Num_Calif_cb:
ca_to_ca_cb + d_to_ca_cb = 7 - diff_ca_cb;

subject to Num_Calif_mg:
ca_to_ca_mg + d_to_ca_mg = 2 - diff_ca_mg;

subject to Num_Calif_mac:
ca_to_ca_mac + d_to_ca_mac = 22 - diff_ca_mac;

subject to Num_Dallas_cb:
d_to_d_cb + ca_to_d_cb + ch_to_d_cb = 28 - diff_d_cb;

subject to Num_Dallas_mg:
d_to_d_mg + ca_to_d_mg + ch_to_d_mg = 3 - diff_d_mg;

subject to Num_Dallas_mac:
d_to_d_mac + ca_to_d_mac + ch_to_d_mac = 45 - diff_d_mac;

subject to Num_Chicago_cb:
ch_to_ch_cb + d_to_ch_cb = 55 - diff_ch_cb;

subject to Num_Chicago_mg:
ch_to_ch_mg + d_to_ch_mg = 12 - diff_ch_mg;

subject to Num_Chicago_mac:
ch_to_ch_mac + d_to_ch_mac = 72 - diff_ch_mac;

problem model 1:

difference,
ca_to_ca_cb,
ca_to_d_cb,
ca_to_ca_mg,
ca_to_d_mg,
ca_to_ca_mac,
ca_to_d_mac,
d_to_d_cb,
d_to_ca_cb,
d_to_ch_cb,
d_to_d_mg,
d_to_ca_mg,
d_to_ch_mg,
d_to_d_mac,
d_to_ca_mac,
d_to_ch_mac,
ch_to_ch_cb,
ch_to_d_cb,
ch_to_ch_mg,
ch_to_d_mg,
ch_to_ch_mac,
ch_to_d_mac,
diff_ca_cb,
diff_ca_mg,
diff_ca_mac,
diff_d_cb,
diff_d_mg,
diff_d_mac,
diff_ch_cb,
diff_ch_mg,
diff_ch_mac,
Capacity_Calif,
Capacity_Dallas,
Capacity_Chicago,
Num_Calif_cb,
Num_Calif_mg,
Num_Calif_mac,
Num_Dallas_cb,
Num_Dallas_mg,
Num_Dallas_mac,
Num_Chicago_cb,
Num_Chicago_mg,
Num_Chicago_mac;

problem model2:
cost,
ca_to_ca_cb,
ca_to_d_cb,
ca_to_ca_mg,
ca_to_d_mg,
ca_to_ca_mac,
ca_to_d_mac,
d_to_d_cb,
d_to_ca_cb,
d_to_ch_cb,
d_to_d_mg,
d_to_ca_mg,
d_to_ch_mg,
d_to_d_mac,
d_to_ca_mac,
d_to_ch_mac,
ch_to_ch_cb,
ch_to_d_cb,
ch_to_ch_mg,
ch_to_d_mg,
ch_to_ch_mac,
ch_to_d_mac,
Capacity_Calif,
Capacity_Dallas,
Capacity_Chicago,
Num_Calif_cb,
Num_Calif_mg,
Num_Calif_mac,
Num_Dallas_cb,
Num_Dallas_mg,
Num_Dallas_mac,
Num_Chicago_cb,
Num_Chicago_mg,
Num_Chicago_mac;

problem model1;
solve;
printf"diff_ca_cb = %3.0f\n",diff_ca_cb;;
printf"diff_ca_mg = %3.0f\n",diff_ca_mg;;
printf"diff_ca_mac = %3.0f\n",diff_ca_mac;;
printf"diff_d_cb = %3.0f\n",diff_d_cb;;
printf"diff_d_mg = %3.0f\n",diff_d_mg;;
printf"diff_ch_cb = %3.0f\n",diff_ch_cb;;
printf"diff_mg = %3.0f\n",diff_mg;;
printf"diff_mac = %3.0f\n",diff_mac;;

fix diff_ca_cb;
fix diff_ca_mg;
fix diff_ca_mac;
fix diff_d_cb;
fix diff_d_mg;
fix diff_d_mac;
fix diff_ch_cb;
fix diff_ch_mg;
fix diff_ch_mac;

problem model2;
solve;
printf"ca_to_ca_cb = %3.0f\n",ca_to_ca_cb;;
printf"ca_to_d_cb = %3.0f\n",ca_to_d_cb;;
printf"ca_to_ca_mg = %3.0f\n",ca_to_ca_mg;;
printf"ca_to_d_mg = %3.0f\n",ca_to_d_mg;;
printf"ca_to_ca_mac = %3.0f\n",ca_to_ca_mac;;
printf"ca_to_d_mac = %3.0f\n",ca_to_d_mac;;
printf"d_to_d_cb = %3.0f\n",d_to_d_cb;;
printf"d_to_d_mg = %3.0f\n",d_to_d_mg;;
printf"d_to_d_mac = %3.0f\n",d_to_d_mac;;
printf "d_to_ca_mac = %3.0f\n", d_to_ca_mac;
printf "d_to_ch_mac = %3.0f\n", d_to_ch_mac;
printf "ch_to_ch_cb = %3.0f\n", ch_to_ch_cb;
printf "ch_to_d_cb = %3.0f\n", ch_to_d_cb;
printf "ch_to_ch_mg = %3.0f\n", ch_to_ch_mg;
printf "ch_to_d_mg = %3.0f\n", ch_to_d_mg;
printf "ch_to_ch_mac = %3.0f\n", ch_to_ch_mac;
printf "ch_to_d_mac = %3.0f\n", ch_to_d_mac;

printf "Capacity Used in California = %3.0f\n", ca_to_ca_cb * 225160 + ca_to_d_cb * 225160 + 
          ca_to_d_mg * 251992 + ca_to_ca_mac * 152516 + ca_to_d_mac * 1525166;
printf "Capacity Remaining in California = %3.0f\n", 5403840 - (ca_to_ca_cb * 225160 + ca_to_d_ 
               cb * 225160 + ca_to_ca_mg * 251992 + ca_to_ca_mac * 152516 + ca_to_d_mac * 152516);
printf "Capacity Used in Dallas = %3.0f\n", d_to_d_cb * 225160 + d_to_ca_cb * 225160 + d_to_c 
               h_cb * 225160 + d_to_d_mg * 251992 + d_to_ca_mg * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152 
               516 + d_to_d_mac * 152516;
printf "Capacity Remaining in Dallas = %3.0f\n", 8556080 - (d_to_d_cb * 225160 + d_to_ca_cb * 
               225160 + d_to_ch_cb * 225160 + d_to_d_mg * 251992 + d_to_ca_mg * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152516 + d_to_d_mac * 152516);
printf "Capacity Used in Chicago = %3.0f\n", ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_ 
               to_ch_mg * 251992 + ch_to_d_mg * 251992 + ch_to_ch_mac * 152516 + ch_to_d_mac * 152516;
printf "Capacity Remaining in Chicago = %3.0f\n", 10807680 - (ch_to_ch_cb * 225160 + ch_to_d_ 
               cb * 225160 + ch_to_ch_mg * 251992 + ch_to_d_mg * 251992 + ch_to_ch_mac * 152516 + ch_to_d_mac * 152516);

printf "California CB's Serviced = %3.0f\n", ca_to_ca_cb + d_to_ca_cb;
printf "California MG's Serviced = %3.0f\n", ca_to_ca_mg + d_to_ca_mg;
printf "California Mac's Serviced = %3.0f\n", ca_to_ca_mac + d_to_ca_mac;

printf "Dallas CB's Serviced = %3.0f\n", d_to_d_cb + ch_to_d_cb + ca_to_d_cb;
printf "Dallas MG's Serviced = %3.0f\n", d_to_d_mg + ch_to_d_mg + ca_to_d_mg;
printf "Dallas Mac's Serviced = %3.0f\n", d_to_d_mac + ch_to_d_mac + ca_to_d_mac;

printf "Chicago CB's Serviced = %3.0f\n", ch_to_ch_cb + d_to_ch_cb;
printf "Chicago MG's Serviced = %3.0f\n", ch_to_ch_mg + d_to_ch_mg;
printf "Chicago Mac's Serviced = %3.0f\n", ch_to_ch_mac + d_to_ch_mac;
option solver cplex;
var ca_to_ca_cb >= 0 integer;
var ca_to_d_cb >= 0 integer;
var ca_to_ca_mg >= 0 integer;
var ca_to_d_mg >= 0 integer;
var ca_to_ca_mac >= 0 integer;
var ca_to_d_mac >= 0 integer;
var d_to_d_cb >= 0 integer;
var d_to_ca_cb >= 0 integer;
var d_to_ch_cb >= 0 integer;
var d_to_d_mg >= 0 integer;
var d_to_ca_mg >= 0 integer;
var d_to_d_mac >= 0 integer;
var d_to_ca_mac >= 0 integer;
var d_to_ch_mac >= 0 integer;
var ch_to_ch_cb >= 0 integer;
var ch_to_d_cb >= 0 integer;
var ch_to_d_mg >= 0 integer;
var ch_to_d_mac >= 0 integer;
var ch_to_ch_mac >= 0 integer;
var ch_to_d_mac >= 0 integer;
var diff_ca_cb >= 0 integer;
var diff_ca_mg >= 0 integer;
var diff_ca_mac >= 0 integer;
var diff_d_cb >= 0 integer;
var diff_d_mg >= 0 integer;
var diff_d_mac >= 0 integer;
var diff_ch_cb >= 0 integer;
var diff_ch_mg >= 0 integer;
var diff_ch_mac >= 0 integer;
minimize difference:
diff_ca_cb+diff_ca_mg+diff_ca_mac+ diff_d_cb+diff_d_mg+diff_d_mac+ diff_ch_cb+diff_ch_mg+diff_ch_mac;
minimize cost:
13959.92 * ca_to_ca_cb + 13959.92 * ca_to_d_cb + 22100 * ca_to_d_cb + 15623.4 * ca_to_ca_mg
 + 15623.4 * ca_to_d_mg +
25500 * ca_to_d_mg + 9456.2 * ca_to_ca_mac + 9456.2 * ca_to_d_mac + 11900 * ca_to_d_mac + 33
098.52 * d_to_d_cb +
33098.52 * d_to_ca_cb + 22100 * d_to_ca_cb + 33098.52 * d_to_ch_cb + 10400 * d_to_ch_cb + 37
024.72 * d_to_d_mg +
37042.72 * d_to_ca_mg + 25000 * d_to_ca_mg + 37042.72 * d_to_ch_mg + 12000 * d_to_ch_mg + 22
419.8 * d_to_d_mac +
22419.8 * d_to_ca_mac + 11900 * d_to_ca_mac + 22419.8 * d_to_ch_mac + 5600 * d_to_ch_mac + 5
1786.8 * ch_to_d_cb +
51786.8 * ch_to_d_cb + 10400 * ch_to_d_cb + 57958.16 * ch_to_ch_mg + 57958.16 * ch_to_d_mg +
12000 * ch_to_d_mg +
35078.68 * ch_to_ch_mac + 35078.68 * ch_to_d_mac + 5600 * ch_to_d_mac;
subject to Capacity_Calif:
10920000 >= ca_to_ca_cb * 225160 + ca_to_d_cb * 225160 + ca_to_ca_mg * 251992 + ca_to_d_mg *
251992 + 
ca_to_ca_mac * 152516 + ca_to_d_mac * 152516;

subject to Capacity_Dallas:
10920000 >= d_to_d_cb * 225160 + d_to_ca_cb * 225160 + d_to_ch_cb * 225160 + d_to_d_mg * 251992 + d_to_ca_mg * 
251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152516 + d_to_ca_mac * 152516 + d_to_ch_mac * 15 
2516;

subject to Capacity_Chicago:
10920000 >= ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_to_ch_mg * 251992 + ch_to_d_mg * 
251992 + 
ch_to_ch_mac * 152516 + ch_to_d_mac * 152516;

subject to Num_Calif_cb:
ca_to_ca_cb + d_to_ca_cb = 7 - diff_ca_cb;

subject to Num_Calif_mg:
ca_to_ca_mg + d_to_ca_mg = 2 - diff_ca_mg;

subject to Num_Calif_mac:
ca_to_ca_mac + d_to_ca_mac = 22 - diff_ca_mac;

subject to Num_DalIas_cb:
d_to_d_cb + ca_to_d_cb + ch_to_d_cb = 28 - diff_d_cb;

subject to Num_Dallas_mg:
d_to_d_mg + ca_to_d_mg + ch_to_d_mg = 3 - diff_d_mg;

subject to Num_Dallas_mac:
d_to_d_mac + ca_to_d_mac + ch_to_d_mac = 45 - diff_d_mac;

subject to Num_Chicago_cb:
ch_to_ch_cb + d_to_ch_cb = 55 - diff_ch_cb;

subject to Num_Chicago_mg:
ch_to_ch_mg + d_to_ch_mg = 12 - diff_ch_mg;

subject to Num_Chicago_mac:
ch_to_ch_mac + d_to_ch_mac = 72 - diff_ch_mac;

problem model1:
difference, ca_to_ca_cb, ca_to_d_cb, ca_to_ca_mg, ca_to_d_mg, ca_to_ca_mac, ca_to_d_mac, d_to_d_cb, 
d_to_ca_cb, d_to_ch_cb, d_to_d_mg, d_to_ca_mg, d_to_ch_mg, d_to_d_mac, d_to_ca_mac, d_to_ch_mac, 
ch_to_ch_cb, ch_to_d_cb, ch_to_ch_mg, ch_to_d_mg, ch_to_ch_mac, ch_to_d_mac, diff_ca_ 
cb, diff_ca_mg, diff_ca_mac, 
diff_d_cb, diff_d_mg, diff_d_mac, diff_ch_cb, diff_ch_mg, diff_ch_mac, Capacity_Calif, 
Capacity_Dallas, 
Capacity_Chicago, Num_Calif_cb, Num_Calif_mg, Num_Calif_mac, Num_Dallas_cb, Num_Dallas_mg,
Num_Dallas_mac, Num_Chicago_cb, Num_Chicago_mg, Num_Chicago_mac;

problem model2:
  cost, ca_to_ca_cb, ca_to_d_cb, ca_to_ca_mg, ca_to_d_mg, ca_to_ca_mac, ca_to_d_mac, d_to_d_cb, d_to_ca_cb, 
  d_to_ch_cb, d_to_d_mg, d_to_ca_mg, d_to_ch_mg, d_to_d_mac, d_to_ca_mac, d_to_ch_mac, ch_to_d_cb, ch_to_ch_mg, 
  ch_to_d_mg, ch_to_ch_mac, ch_to_d_mac, Capacity_Calif, Capacity_Dallas, Capacity_Chicago, Num_Calif_cb, Num_Calif_mg, Num_Calif_mac, Num_Dallas_cb, Num_Dallas_mg, Num_Dallas_mac,

Num_Chicago_cb, Num_Chicago_mg, Num_Chicago_mac;

problem model1;
  solve;
  printf"diff_ca_cb = %3.0f\n",diff_ca_cb;
  printf"diff_ca_mg = %3.0f\n",diff_ca_mg;
  printf"diff_ca_mac = %3.0f\n",diff_ca_mac;
  printf"diff_d_cb = %3.0f\n",diff_d_cb;
  printf"diff_d_mg = %3.0f\n",diff_d_mg;
  printf"diff_d_mac = %3.0f\n",diff_d_mac;
  printf"diff_ch_cb = %3.0f\n",diff_ch_cb;
  printf"diff_ch_mg = %3.0f\n",diff_ch_mg;
  printf"diff_ch_mac = %3.0f\n",diff_ch_mac;

  fix diff_ca_cb;
  fix diff_ca_mg;
  fix diff_ca_mac;
  fix diff_d_cb;
  fix diff_d_mg;
  fix diff_d_mac;
  fix diff_ch_cb;
  fix diff_ch_mg;
  fix diff_ch_mac;

problem model2;
  solve;
  printf"ca_to_ca_cb = %3.0f\n",ca_to_ca_cb;
  printf"ca_to_d_cb = %3.0f\n",ca_to_d_cb;
  printf"ca_to_ca_mg = %3.0f\n",ca_to_ca_mg;
  printf"ca_to_d_mg = %3.0f\n",ca_to_d_mg;
  printf"ca_to_ca_mac = %3.0f\n",ca_to_ca_mac;
  printf"ca_to_d_mac = %3.0f\n",ca_to_d_mac;
  printf"d_to_d_cb = %3.0f\n",d_to_d_cb;
  printf"d_to_ca_cb = %3.0f\n",d_to_ca_cb;
  printf"d_to_ch_cb = %3.0f\n",d_to_ch_cb;
  printf"d_to_d_mg = %3.0f\n",d_to_d_mg;
  printf"d_to_ca_mg = %3.0f\n",d_to_ca_mg;
  printf"d_to_ch_mg = %3.0f\n",d_to_ch_mg;
  printf"d_to_d_mac = %3.0f\n",d_to_d_mac;
  printf"d_to_ca_mac = %3.0f\n",d_to_ca_mac;
  printf"d_to_ch_mac = %3.0f\n",d_to_ch_mac;
  printf"ch_to_ch_cb = %3.0f\n",ch_to_ch_cb;
  printf"ch_to_d_cb = %3.0f\n",ch_to_d_cb;
printf("ch_to_ch_mg = %3.0f\n",ch_to_ch_mg);
printf("ch_to_d_mg = %3.0f\n",ch_to_d_mg);
printf("ch_to_ch_mac = %3.0f\n",ch_to_ch_mac);
printf("ch_to_d_mac = %3.0f\n",ch_to_d_mac);

printf("Capacity Used in California = %3.0f\n", ca_to_ca_cb * 225160 + ca_to_d_cb * 225160 + ca_to_ca_mg * 251992 + ca_to_d_mg * 251992 + ca_to_ca_mac * 152516 + ca_to_d_mac * 152516);
printf("Capacity Remaining in California = %3.0f\n",10920000 - (ca_to_ca_cb * 225160 + ca_to_d_cb * 225160 + ca_to_ca_mg * 251992 + ca_to_d_mg * 251992 + ca_to_ca_mac * 152516 + ca_to_d_mac * 152516);
printf("Capacity Used in Dallas = %3.0f\n", d_to_d_cb * 225160 + d_to_ca_cb * 225160 + d_to_ch_cb * 225160 + d_to_d_mg * 251992 + d_to_ca_mg * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152516 + d_to_ca_mac * 152516);
printf("Capacity Remaining in Dallas = %3.0f\n", 10920000 - (d_to_d_cb * 225160 + d_to_ca_cb * 225160 + d_to_ch_cb * 225160 + d_to_d_mg * 251992 + d_to_ca_mg * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152516 + d_to_ca_mac * 152516);
printf("Capacity Used in Chicago = %3.0f\n", ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_to_ch_mg * 251992 + ch_to_d_mg * 251992 + ch_to_ch_mac * 152516 + ch_to_d_mac * 152516);
printf("Capacity Remaining in Chicago = %3.0f\n",10920000 - (ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_to_ch_mg * 251992 + ch_to_d_mg * 251992 + ch_to_ch_mac * 152516 + ch_to_d_mac * 152516);

printf("California CB's Serviced = %3.0f\n", ca_to_ca_cb + d_to_ca_cb);
printf("California MG's Serviced = %3.0f\n", ca_to_ca_mg + d_to_ca_mg);
printf("California Mac's Serviced = %3.0f\n", ca_to_ca_mac + d_to_ca_mac);

printf("Dallas CB's Serviced = %3.0f\n", d_to_d_cb + ch_to_d_cb + ca_to_d_cb);
printf("Dallas MG's Serviced = %3.0f\n", d_to_d_mg + ch_to_d_mg + ca_to_d_mg);
printf("Dallas Mac's Serviced = %3.0f\n", d_to_d_mac + ch_to_d_mac + ca_to_d_mac);

printf("Chicago CB's Serviced = %3.0f\n", ch_to_ch_cb + d_to_ch_cb);
printf("Chicago MG's Serviced = %3.0f\n", ch_to_ch_mg + d_to_ch_mg);
printf("Chicago Mac's Serviced = %3.0f\n", ch_to_ch_mac + d_to_ch_mac);
option solver cplex;
var ca_to_ca_cb >= 0 integer;
var ca_to_d_cb >= 0 integer;
var ca_to_ca_mg >= 0 integer;
var ca_to_d_mg >= 0 integer;
var ca_to_ca_mac >= 0 integer;
var ca_to_d_mac >= 0 integer;
var d_to_d_cb >= 0 integer;
var d_to_ca_cb >= 0 integer;
var d_to_ch_cb >= 0 integer;
var d_to_ca_mg >= 0 integer;
var d_to_ch_mg >= 0 integer;
var d_to_d_mac >= 0 integer;
var d_to_ca_mac >= 0 integer;
var d_to_ch_mac >= 0 integer;
var ch_to_ch_cb >= 0 integer;
var ch_to_d_cb >= 0 integer;
var ch_to_ch_mg >= 0 integer;
var ch_to_d_mg >= 0 integer;
var ch_to_ch_mac >= 0 integer;
var ch_to_d_mac >= 0 integer;
var diff_ca_cb >= 0 integer;
var diff_ca_mg >= 0 integer;
var diff_ca_mac >= 0 integer;
var diff_d_cb >= 0 integer;
var diff_d_mg >= 0 integer;
var diff_d_mac >= 0 integer;
var diff_ch_cb >= 0 integer;
var diff_ch_mg >= 0 integer;
var diff_ch_mac >= 0 integer;

minimize cost:
13959.92 * ca_to_ca_cb + 13959.92 * ca_to_d_cb + 22100 * ca_to_d_cb + 15623.4 * ca_to_ca_mg + 15623.4 * ca_to_d_mg + 25500 * ca_to_d_mg + 9456.2 * ca_to_ca_mac + 9456.2 * ca_to_d_mac + 11900 * ca_to_d_mac + 33098.52 * d_to_d_cb + 33098.52 * d_to_ca_cb + 22100 * d_to_ca_cb + 33098.52 * d_to_ch_cb + 10400 * d_to_ch_cb + 37042.72 * d_to_d_mg + 37042.72 * d_to_ca_mg + 25500 * d_to_ca_mg + 37042.72 * d_to_ch_mg + 10400 * d_to_ch_mg + 12000 * d_to_ch_mg + 11900 * d_to_d_mac + 11900 * d_to_d_mac + 22419.8 * d_to_ch_mac + 51786.8 * ch_to_ch_cb + 51786.8 * ch_to_d_cb + 10400 * ch_to_d_cb + 57958.16 * ch_to_ch_mg + 57958.16 * ch_to_d_mg + 12000 * ch_to_d_mg + 35078.68 * ch_to_ch_mac + 35078.68 * ch_to_d_mac + 5600 * ch_to_d_mac;

subject to Capacity_Calif:
13104000 >= ca_to_ca_cb * 225160 + ca_to_d_cb * 225160 + ca_to_ca_mg * 251992 + ca_to_d_mg *
251992 + ca_to_ca_mac * 152516 + ca_to_d_mac * 152516;

subject to Capacity_Dallas:
10920000 >= d_to_d_cb * 225160 + d_to_ca_cb * 225160 + d_to_ch_cb * 225160 + d_to_d_mg * 251992 + d_to_ca_mg * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152516 + d_to_ca_mac * 152516 + d_to_ch_mac * 152516;

subject to Capacity_Chicago:
13104000 >= ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_to_ch_mg * 251992 + ch_to_d_mg * 251992 + ch_to_ch_mac * 152516 + ch_to_d_mac * 152516;

subject to Num_Calif_cb:
ca_to_ca_cb + d_to_ca_cb = 7 - diff_ca_cb;

subject to Num_Calif_mg:
ca_to_ca_mg + d_to_ca_mg = 2 - diff_ca_mg;

subject to Num_Calif_mac:
ca_to_ca_mac + d_to_ca_mac = 22 - diff_ca_mac;

subject to Num_Dallas_cb:
d_to_d_cb + ca_to_d_cb + ch_to_d_cb = 28 - diff_d_cb;

subject to Num_Dallas_mg:
d_to_d_mg + ca_to_d_mg + ch_to_d_mg = 3 - diff_d_mg;

subject to Num_Dallas_mac:
d_to_d_mac + ca_to_d_mac + ch_to_d_mac = 45 - diff_d_mac;

subject to Num_Chicago_cb:
ch_to_ch_cb + d_to_ch_cb = 55 - diff_ch_cb;

subject to Num_Chicago_mg:
ch_to_ch_mg + d_to_ch_mg = 12 - diff_ch_mg;

subject to Num_Chicago_mac:
ch_to_ch_mac + d_to_ch_mac = 72 - diff_ch_mac;

problem model1:

difference, ca_to_ca_cb, ca_to_d_cb, ca_to_ca_mg, ca_to_d_mg, ca_to_ca_mac, ca_to_d_mac, d_to_d_cb,
d_to_ca_cb, d_to_d_cb, d_to_d_mg, d_to_ca_mg, d_to_ch_mg, d_to_d_mac, d_to_ca_mac, d_to_ch_mac,
ch_to_ch_cb, ch_to_d_cb, ch_to_ch_mg, ch_to_d_mg, ch_to_ch_mac, ch_to_d_mac, ch_to_d_mac,
diff_ca_mg, diff_ca_mac,
diff_d_cb, diff_d_mg, diff_d_mac,
diff_ch_cb, diff_ch_mg, diff_ch_mac, Capacity_Calif,
Capacity_Dallas,
Capacity_Chicago, Num_Calif_cb, Num_Calif_mg, Num_Calif_mac, Num_Dallas_cb, Num_Dallas_mg,
Num_Dallas_mac, Num_Chicago(cb, mg, mac)
Num_Dallas_mac, Num_Chicago_cb, Num_Chicago_mg, Num_Chicago_mac;

problem model2:
    cost, ca_to_ca_cb, ca_to_d_cb, ca_to_ca_mg, ca_to_d_mg, ca_to_ca_mac, ca_to_d_mac, d_to_d_cb, d_to_ca_cb, d_to_d_mg, d_to_ca_mg, d_to_d_mac, d_to_ca_mac, d_to_ch_cb, ch_to_d_cb, ch_to_ch_mg, ch_to_d_mg, ch_to_ch_mac, ch_to_d_mac, Capacity_Calif, Capacity_Dallas, Capacity_Chicago,
Num_Calif_cb, Num_Calif_mg, Num_Calif_mac, Num_Dallas_cb, Num_Dallas_mg, Num_Dallas_mac,

Num_Chicago_cb, Num_Chicago_mg, Num_Chicago_mac;

problem model1;
solve;
printf"diff_ca_cb = %3.0f
",diff_ca_cb;
printf"diff_ca_mg = %3.0f
",diff_ca_mg;
printf"diff_ca_mac = %3.0f
",diff_ca_mac;
printf"diff_d_cb = %3.0f
",diff_d_cb;
printf"diff_d_mg = %3.0f
",diff_d_mg;
printf"diff_ch_cb = %3.0f
",diff_ch_cb;
printf"diff_ch_mg = %3.0f
",diff_ch_mg;
printf"diff_ch_mac = %3.0f
",diff_ch_mac;

fix diff_ca_cb;
fix diff_ca_mg;
fix diff_ca_mac;
fix diff_d_cb;
fix diff_d_mg;
fix ch_to_d_mc;
fix ch_to_d_cm;
fix ch_to_d_mpc;
fix ch_to_d_mcm;

problem model2;
solve;
printf"ca_to_ca_cb = %3.0f
",ca_to_ca_cb;
printf"ca_to_d_cb = %3.0f
",ca_to_d_cb;
printf"ca_to_ca_mg = %3.0f
",ca_to_ca_mg;
printf"ca_to_d_mg = %3.0f
",ca_to_d_mg;
printf"ca_to_ca_mac = %3.0f
",ca_to_ca_mac;
printf"ca_to_d_mac = %3.0f
",ca_to_d_mac;
printf"d_to_d_cb = %3.0f
",d_to_d_cb;
printf"d_to_d_mg = %3.0f
",d_to_d_mg;
printf"d_to_d_mac = %3.0f
",d_to_d_mac;
printf"d_to_ca_pb = %3.0f
",d_to_ca_pb;
printf"d_to_ch_mg = %3.0f
",d_to_ch_mg;
printf"d_to_d_mac = %3.0f
",d_to_d_mac;
printf"ch_to_d_cb = %3.0f
",ch_to_d_cb;
printf"ch_to_d_cm = %3.0f
",ch_to_d_cm;
printf"ch_to_d_mpc = %3.0f
",ch_to_d_mpc;
printf"ch_to_d_mcm = %3.0f
",ch_to_d_mcm;
printf"ch_to_d_mc = %3.0f
",ch_to_d_mc;

printf"ch_to_ch_mg = %3.0f\n",ch_to_ch_mg; 
printf"ch_to_d_mg = %3.0f\n",ch_to_d_mg; 
printf"ch_to_ch_mac = %3.0f\n",ch_to_ch_mac; 
printf"ch_to_d_mac = %3.0f\n",ch_to_d_mac; 

printf"Capacity Used in California = %3.0f\n", ca_to_ca_cb * 225160 + ca_to_d_cb * 225160 + ca_to_d_mg * 251992 + ca_to_d_mac * 251992 + ca_to_ca_mac * 152516 + ca_to_d_mac * 152516; 
printf"Capacity Remaining in California = %3.0f\n",13104000 -(ca_to_ca_cb * 225160 + ca_to_d_cb * 225160 + ca_to_d_mg * 251992 + ca_to_d_mac * 251992 + ca_to_ca_mac * 152516 + ca_to_d_mac * 152516) ;

printf"Capacity Used in Dallas = %3.0f\n", d_to_d_cb * 225160 + d_to_ch_cb * 225160 + d_to_d_mg * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152516 + d_to_ch_mac * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 152516 + d_to_ch_mac * 152516; 
printf"Capacity Remaining in Dallas = %3.0f\n",10920000 -(d_to_d_cb * 225160 + d_to_ch_cb * 225160 + d_to_d_mg * 251992 + d_to_ch_mg * 251992 + d_to_d_mac * 251992 + d_to_ch_mac * 251992 + d_to_d_mac * 251992 + d_to_ch_mac * 251992 + d_to_d_mac * 152516 + d_to_ch_mac * 152516 + d_to_d_mac * 152516);

printf"Capacity Used in Chicago = %3.0f\n", ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_to_ch_mg * 251992 + ch_to_d_mg * 251992 + ch_to_ch_mac * 152516 + ch_to_d_mac * 152516; 
printf"Capacity Remaining in Chicago = %3.0f\n",13104000 -(ch_to_ch_cb * 225160 + ch_to_d_cb * 225160 + ch_to_d_mg * 251992 + ch_to_d_mac * 251992 + ch_to_ch_mac * 251992 + ch_to_d_mac * 251992 + ch_to_d_mac * 251992 + ch_to_d_mac * 251992 + ch_to_ch_mac * 152516 + ch_to_d_mac * 152516);

printf"California CB's Serviced = %3.0f\n", ca_to_ca_cb + d_to_ca_cb; 
printf"California MG's Serviced = %3.0f\n", ca_to_ca_mg + d_to_ca_mg; 
printf"California Mac's Serviced = %3.0f\n", ca_to_ca_mac + d_to_ca_mac; 

printf"Dallas CB's Serviced = %3.0f\n", d_to_d_cb + ch_to_d_cb + ca_to_d_cb; 
printf"Dallas MG's Serviced = %3.0f\n", d_to_d_mg + ch_to_d_mg + ca_to_d_mg; 
printf"Dallas Mac's Serviced = %3.0f\n", d_to_d_mac + ch_to_d_mac + ca_to_d_mac; 

printf"Chicago CB's Serviced = %3.0f\n", ch_to_ch_cb + d_to_ch_cb; 
printf"Chicago MG's Serviced = %3.0f\n", ch_to_ch_mg + d_to_ch_mg; 
printf"Chicago Mac's Serviced = %3.0f\n", ch_to_ch_mac + d_to_ch_mac;
CPLEX 7.1.0: optimal integer solution; objective 104
108 MIP simplex iterations
117 branch-and-bound nodes
derif ca cb = 7
derif ca mg = 1
derif ca mac = 0
derif d cb = 27
derif d mg = 1
derif ch cb = 53
derif ch mg = 12
derif ch mac = 0
CPLEX 7.1.0: optimal integer solution; objective 3758233
7 MIP simplex iterations
0 branch-and-bound nodes
cal to ca cb = 0
cal to d cb = 1
cal to ca mg = 1
cal to d mg = 0
cal to ca mac = 22
cal to d mac = 4
d to d cb = 0
d to ca cb = 0
d to ch cb = 0
d to d mg = 0
d to ca mg = 0
d to ch mg = 0
d to d mac = 40
d to ca mac = 0
d to ch mac = 16
chet ch cb = 2
chet ch mg = 0
chet ch mg = 0
chet ch mac = 56
chet ch mg = 0
Capacity Used in California = 4442568
Capacity Remaining in California = 60632
Capacity Used in Dallas = 8540896
Capacity Remaining in Dallas = 15184
Capacity Used in Chicago = 8991216
Capacity Remaining in Chicago = 15184
California CB's Serviced = 0
California MG's Serviced = 1
California Mac's Serviced = 22
Dallas CB's Serviced = 1
Dallas MG's Serviced = 0
Dallas Mac's Serviced = 44
Chicago CB's Serviced = 2
Chicago MG's Serviced = 0
Chicago Mac's Serviced = 72
CPLEX 7.1.0: optimal integer solution; objective 92
21 MIP simplex iterations
0 branch-and-bound nodes
diff_ca_cb = 7
diff_ca_mg = 2
diff_ca_mac = 0
diff_d_cb = 28
diff_d_mg = 2
diff_ch_cb = 41
diff_ch_mg = 11
diff_ch_mac = 0
CPLEX 7.1.0: optimal integer solution; objective 4342325.8
4 MIP simplex iterations
0 branch-and-bound nodes
c_to_ca_cb = 0
c_to_d_cb = 0
c_to_ca_mg = 0
c_to_d_mg = 0
c_to_ca_mac = 22
c_to_d_mac = 13
d_to_d_cb = 0
d_to_ca_cb = 0
d_to_ch_cb = 0
d_to_d_mg = 0
d_to_ca_mg = 0
d_to_ch_mg = 0
d_to_d_mac = 32
d_to_ca_mac = 0
d_to_ch_mac = 24
ch_to_ch_cb = 14
ch_to_d_cb = 0
ch_to_ch_mg = 1
ch_to_d_mg = 0
ch_to_ch_mac = 48
ch_to_d_mac = 0
Capacity Used in California = 5338060
Capacity Remaining in California = 65780
Capacity Used in Dallas = 8540896
Capacity Remaining in Dallas = 15184
Capacity Used in Chicago = 10725000
Capacity Remaining in Chicago = 82680
California CB's Serviced = 0
California MG's Serviced = 0
California Mac's Serviced = 22
Dallas CB's Serviced = 0
Dallas MG's Serviced = 0
Dallas Mac's Serviced = 45
Chicago CB's Serviced = 14
Chicago MG's Serviced = 1
Chicago Mac's Serviced = 72
CPLEX 7.1.0: optimal integer solution; objective 56
72 MIP simplex iterations
29 branch-and-bound nodes
diff_ca_cb = 6
diff_ca_mg = 2
diff_ca_mac = 0
diff_d_cb = 21
diff_d_mg = 2
diff_ch_cb = 13
diff_ch_mg = 12
diff_ch_mac = 0
CPLEX 7.1.0: optimal integer solution within mipgap or absmipgap;
objective 5719511.08
10 MIP simplex iterations
0 branch-and-bound nodes
ca_to_ca_cb = 1
c_to_ca_d = 2
c_to_ca_mg = 0
c_to_d_mg = 0
c_to_ca_mac = 22
c_to_d_mac = 45
d_to_d_cb = 5
d_to_ca_d = 0
d_to_ch_d = 1
d_to_d_mg = 1
d_to_ca_mg = 0
d_to_d_mac = 0
d_to_ca_mac = 0
d_to_ch_mac = 61
ch_to_ch_d = 41
ch_to_d_cb = 0
ch_to_ch_mg = 0
ch_to_d_mg = 0
ch_to_ch_mac = 11
ch_to_d_mac = 0
Capacity Used in California = 10894052
Capacity Remaining in California = 25948
Capacity Used in Dallas = 10906428
Capacity Remaining in Dallas = 13572
Capacity Used in Chicago = 10909236
Capacity Remaining in Chicago = 10764
California CB's Serviced = 1
California MG's Serviced = 0
California Mac's Serviced = 22
Dallas CB's Serviced = 7
Dallas MG's Serviced = 1
Dallas Mac's Serviced = 45
Chicago CB's Serviced = 42
Chicago MG's Serviced = 0
Chicago Mac's Serviced = 72
CPLEX 7.1.0: optimal integer solution within mipgap or absmipgap; objective 37
16 MIP simplex iterations
0 branch-and-bound nodes
diff_ca_cb = 7
diff_ca_mg = 2
diff_ca_mac = 0
diff_d_cb = 13
diff_d_mg = 2
diff_ch_cb = 0
diff_ch_mg = 12
diff_ch_mac = 0
CPLEX 7.1.0: optimal integer solution; objective 6603065.12
12 MIP simplex iterations
1 branch-and-bound nodes
ca_to_ca_cb = 0
c_to_d_cb = 14
c_to_ca_mg = 0
c_to_d_mg = 0
c_to_ca_mac = 22
c_to_d_mac = 43
d_to_d_cb = 1
d_to_ca_cb = 0
d_to_ch_cb = 0
d_to_d_mg = 0
d_to_ca_mg = 0
d_to_ch_mg = 0
d_to_d_mac = 2
d_to_ca_mac = 0
d_to_ch_mac = 68
c_to_ch_cb = 55
c_to_d_cb = 0
c_to_ch_mg = 0
c_to_d_mg = 0
c_to_ch_mac = 4
c_to_d_mac = 0
Capacity Used in California = 13065780
Capacity Remaining in California = 38220
Capacity Used in Dallas = 10901280
Capacity Remaining in Dallas = 18720
Capacity Used in Chicago = 12993864
Capacity Remaining in Chicago = 110136
California CB's Serviced = 0
California MG's Serviced = 0
California Mac's Serviced = 22
Dallas CB's Serviced = 15
Dallas MG's Serviced = 0
Dallas Mac's Serviced = 45
Chicago CB's Serviced = 55
Chicago MG's Serviced = 0
Chicago Mac's Serviced = 72
Appendix

C
AMPL Code Documentation:

**Fixed Values:**
All production and shipping values are found in the "minimize cost" objective statement. All values can be modified accordingly.

**Production costs**
- $13959.92 = Price to service one Corner Bakery from California plant for one year
- $15623.4 = Price to service one Maggiano's from California Plant for one year
- $9456.2 = Price to service one Macaronni Grill from California plant for one year
- $33098.52 = Price to service one Corner Bakery from Dallas plant for one year
- $37042.72 = Price to service one Maggiano's from Dallas plant for one year
- $9456.2 = Price to service one Macaronni Grill from Dallas plant for one year
- $33098.52 = Price to service one Corner Bakery from Chicago plant for one year
- $37042.72 = Price to service one Maggiano's from Chicago plant for one year
- $57958.16 = Price to service one Macaronni Grill from Chicago plant for one year

**Shipping Costs**
- $22100 = Price to ship one Corner Bakery on route Calif / Dallas for one year
- $25500 = Price to ship one Maggiano's on route Calif / Dallas for one year
- $11900 = Price to ship one Macaronni Grill on route Calif / Dallas for one year
- $10400 = Price to ship one Corner Bakery on route Chicago / Dallas for one year
- $12000 = Price to ship one Maggiano's on route Chicago / Dallas for one year
- $5600 = Price to ship one Macaronni Grill on route Chicago / Dallas for one year

**Pound Requirements:**
All pound requirements are found in the following constraints:

subject to Capacity_Calif:
subject to Capacity_Dallas:
subject to Capacity_Chicago:

All values can be modified accordingly

- $225160 = Total number of pounds required to service one Corner Bakery for one year
- $251992 = Total number of pounds required to service one Maggiano's for one year
- $152516 = Total number of pounds required to service one Macaronni Grill for one year

**Capacities:**
All capacities are found on the left hand side of the capacity constraint inequalities. All values can be modified accordingly.

**Store Requirements:**
Store requirements are the total number of stores to be serviced in a particular region. Store requirements are numeric values found on the right hand side of the following constraints:

subject to Num_Calif_cb: (i.e. # of Corner Bakeries to be serviced from California Plant)
subject to Num_Calif_mg:
subject to Num_Calif_mag:
subject to Num_Dallas_cb:
subject to Num_Dallas_mg:
subject to Num_Dallas_mac:
subject to Num_Chicago_cb:
subject to Num_Chicago_mg:
subject to Num_Chicago_mac:

All values can be modified according to growth requirements.

**Variables:**
Number of Stores Serviced:
These variables denote the number of stores in a particular region serviced by a particular plant. For example, ca_to_ch_mg denotes the number of Maggiano's in Chicago serviced by the California plant.
All such variables are written in such a fashion.

ca = California
ch = Chicago
da = Dallas
cb = Corner Bakery
mg = Maggiano’s
mac = Macaronni Grill

**Difference Variables:**
Difference variables are found on the right hand side of the store requirement constraints. These variables are used to provide flexibility so as to maximize utilized capacity.
For example, if the current capacity cannot fulfill all 7 Corner Bakeries to be serviced in California, the variable diff_ca_cb will denote the number of stores unable to be serviced. Thus, the maximum possible number of stores will be serviced.

**Objectives:**
*minimize difference:*
This constrain minimizes the difference variables so that the maximum number of stores are serviced.

*minimize cost:*
This constraint takes into account all production and distribution costs so that the most cost effective distribution of stores is established.