Assignment 1 Solutions

1 Problem Modeling

1.1 Capital Budgeting (10 marks)

Variables: \( x_1, x_2, x_3 \) and \( x_4 \) are binary variables, representing the investment decisions of project 1, 2, 3, and 4. A value 1 indicating invest while a value 0 indicating don’t invest.

Representation: Binary string

\[
\begin{array}{cccc}
  x_1 & x_2 & x_3 & x_4 \\
\end{array}
\]

Constraints: \( 0.5x_1 + 1.0x_2 + 1.5x_3 + 0.1x_4 \leq 3.1; 0.3x_1 + 0.8x_2 + 1.5x_3 + 0.4x_4 \leq 2.5; 0.2x_1 + 0.2x_2 + 0.3x_3 + 0.1x_4 \leq 0.4. \)

Fitness function: maximize \( f(x_1, x_2, x_3, x_4) = 0.2x_1 + 0.3x_2 + 0.5x_3 + 0.1x_4. \)

1.2 Tournament Scheduling (10 marks)

Variables: \( S \) is the schedule of the tournament.

Representation: We rename the team as 1 to 12. \( S \) is a \( 12 \times 12 \) integer matrix, where \( S_{i,1} \) is fixed

<table>
<thead>
<tr>
<th></th>
<th>day 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

with the team names 1 to 12. Other element of \( S_{i,j} \) (where \( 1 \leq i \leq 12, 2 \leq j \leq 12 \)) is the name
of the team that plays against $S_{i,1}$ on day $j - 1$. For example, let $i = 1, j = 2$, team $S_{i,j}$ (3) plays against team $S_{i,1}$ (1) on day $j - 1$ (1).

**Constraints:** Each team should play against all other 11 teams, hence each row should contain all values from 1 to 12. Similarly, each day all teams should play one game, hence each column should contain all values from 1 to 12. This is similar to the Sudoku game rules, except that the matrix is of a different size:

$$\forall S_i, \sum_{j=1,12} S_{i,j} = 78; \quad \forall S_j, \sum_{i=1,12} S_{i,j} = 78;$$

Combined with the above constraints, the following rules ensure that each row and column is a permutation of values from 1 to 12:

$$\forall S_i, \prod_{j=1,12} S_{i,j} = 12!; \quad \forall S_j, \prod_{i=1,12} S_{i,j} = 12!,$$

Additionally, on each day $j - 1$ (2 ≤ $j$ ≤ 12), if $S_{i,1}$ plays against $S_{i,j}$, $S_{S_{i,j},1}$ must play against $S_{i,1}$. Using the previous example where $i = 1, j = 2$, on day $j - 1$ (1) team $S_{i,1}$ (1) plays against team $S_{i,j}$ (3), team $S_{S_{i,j},1}$ = 3 must play against $S_{i,1}$ = 1.

Under the above constraints, schedule $S$ ensures that on each day, the 12 teams play against each other (permutations) in 6 fields. In terms of assigning field to each tournament, we can iterate the teams in a certain order. For example, on day 1, we first assign field A to team 1, who plays against team 3. Next, we assign field B to team 2, who plays against team 5. Continuing this process one team at a time, each tournament would occupy exactly one field. Hence there is no need to enforce a constraint for field allocation.

**Fitness function:** By treating the above as soft constraints, the objective is to minimize the following:

$$f(S) = f_1(S) + f_2(S) + f_3(S).$$

$$f_1(S) = \forall S_i, |\sum_{j=1,12} S_{i,j} - 78| + |\prod_{j=1,12} S_{i,j} - 12!|$$

$$f_2(S) = \forall S_j, |\sum_{i=1,12} S_{i,j} - 78| + |\prod_{i=1,12} S_{i,j} - 12!|$$

$$f_3(S) = \forall S_j, \sum_{i=1,12} |S_{S_{i,j},1} - S_{i,1}|$$

### 1.3 Sudoku (10 marks)

**Variables:** $S$ is the solution of a Sudoku puzzle.

**Representation:** $S$ is a $9 \times 9$ integer matrix, where each row $S_i$ and each column $S_j$ is a permutation

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>6</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>4</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
of integers 1 to 9. The cells whose values are given are fixed. Only the empty cells can be mutated to solve the puzzle.

**Constraints:**

\[
\forall S_i, \sum_{j=1,9} S_{i,j} = 45; \quad \forall S_j, \sum_{i=1,9} S_{i,j} = 45;
\]

Combined with the above constraints, the following rules ensure that each row and column is a permutation of values from 1 to 9:

\[
\forall S_i, \prod_{j=1,9} S_{i,j} = 9!; \quad \forall S_j, \prod_{i=1,9} S_{i,j} = 9!
\]

**Fitness function:** By treating the above as soft constraints, the objective is to minimize the following:

\[
f(S) = f_1(S) + f_2(S).
\]

\[
f_1(S) = \forall S_i, |\sum_{j=1,9} S_{i,j} - 45| + |\prod_{j=1,9} S_{i,j} - 9!|
\]

\[
f_2(S) = \forall S_j, |\sum_{i=1,9} S_{i,j} - 45| + |\prod_{i=1,9} S_{i,j} - 9!|
\]

# 2 Computer Exercises

## 2.1 Generational Selection-only Model (25 marks)

The solutions are given in Figures 1, 2 and 3.

![Figure 1: Generational Selection-only Model: Population Average Fitness.](image)
2.2 Steady-state Selection-only Model (25 marks)

The solutions are given in Figures 4, 5 and 6.
2.3 Population Size Variation (20 marks)

The solutions are given in Figures 7 – 18.
Figure 6: Steady-state Selection-only Model: Population Diversity.

Figure 7: Generational Fitness Proportionate Selection-only Model: Population Average Fitness.
Figure 8: Generational Fitness Proportionate Selection-only Model: Population Best Fitness.

Figure 9: Generational Fitness Proportionate Selection-only Model: Population Diversity.
Figure 10: Generational Tournament Selection-only Model: Population Average Fitness.

Figure 11: Generational Tournament Selection-only Model: Population Best Fitness.
Figure 12: Generational Tournament Selection-only Model: Population Diversity.

Figure 13: Generational Linear Ranking Selection-only Model ($\epsilon = 1/n^2$): Population Average Fitness.
Figure 14: Generational Linear Ranking Selection-only Model ($\epsilon = 1/n^2$): Population Best Fitness.

Figure 15: Generational Linear Ranking Selection-only Model ($\epsilon = 1/n^2$): Population Diversity.
Figure 16: Generational Truncation Selection-only Model: Population Average Fitness.

Figure 17: Generational Truncation Selection-only Model: Population Best Fitness.
Figure 18: Generational Truncation Selection-only Model: Population Diversity.
Figure 19: Steady-state Fitness Proportionate Selection-only Model: Population Average Fitness.

Figure 20: Steady-state Fitness Proportionate Selection-only Model: Population Best Fitness.
Figure 21: Steady-state Fitness Proportionate Selection-only Model: Population Diversity.