Chapter 3

Two Models with Weighted Utility Functions

3.1 Model I

Let w_i be each weight of $f_i(q_i), \forall i = 1, \ldots, m$, for a strictly concave, increasing function $f_i : \mathbb{R} \to \mathbb{R}$, the function $\sum_{i=1}^{m} w_i f_i(q_i)$ is a strictly monotonic and strictly Schur-concave function [9]. $f_i(q_i)$ is also continuous, increasing, and concave, so is $\sum_{i=1}^{m} w_i f_i(q_i)$.

In the following, we construct a model, Model I, to solve a problem of nonlinear objective function, such that

$$\begin{aligned} \text{(Model I)} \quad \text{maximize} \quad & \sum_{i=1}^{m} w_i f_i(\mathbf{x}) \\ \text{subject to} \quad & \sum_{i=1}^{m} w_i = 1 \\ & w_i \geq 0 \\ & \mathbf{x} \in Q^*. \end{aligned} \tag{3.1}$$

Decision variables in Model I are $\mathbf{x} = (q_i, \chi_{i,j}(e))$, and w_i, for all $i \in I,\ j \in \mathcal{K}_i,\ e \in E$.

In Model I, we assume under the sum of total weights equal to 1, a weighted sum of logarithms of the bandwidth for each class to be maximized.

For simplifying Model I, we take a simple network problem with three classes which take one network connection mechanism for each class. We tried to solve it using the electromagnetism-like mechanism [1] and genetic algorithm [8]. To face hundreds of constraints, it will take almost one hour to get a satisfied numerical result. The complexity of programming for parameter modification will waste so much time and energy on it, so those algorithms were given up in the end.

Finally, we use solver BARON in software GAMS (see [11] and [12]) to solve this kind of network problems. In general, it takes almost less than several minutes to get a numerical result. Besides, we also use it to solve the two weighted models: Model I and II. The numerical results of (3.1) are given in Chapter 4 for demonstration.

3.2 Model II

Consider the Ordered Weighted Averaging Method [9]. First, we define the ordering map

$$\hat{\mathbf{\Psi}} : \mathbb{R}^m \to \mathbb{R}^m.$$

Assume that

$$\hat{\mathbf{\Psi}}(\mathbf{f}(\mathbf{x})) = (\, \hat{\Psi}_1(\mathbf{f}(\mathbf{x})), \hat{\Psi}_2(\mathbf{f}(\mathbf{x})), \ldots, \hat{\Psi}_m(\mathbf{f}(\mathbf{x}))\,) \tag{3.2}$$

where $\hat{\Psi}_1(\mathbf{f}(\mathbf{x}))\leq\hat{\Psi}_2(\mathbf{f}(\mathbf{x}))\leq\ldots\leq\hat{\Psi}_m(\mathbf{f}(\mathbf{x}))$ and there exists a permutation τ of set $S=\{1,2,\ldots,m\}$ such that $\hat{\Psi}_k(\mathbf{f}(\mathbf{x})) = f_{\tau(k)}(\mathbf{x})$ for $k = 1,\ldots,m$. Then, we define the cumulative ordering map $\mathbf{Y}(\mathbf{f}(\mathbf{x})) = (\mathbf{y}_1(\mathbf{f}(\mathbf{x})),\ldots, \mathbf{y}_m(\mathbf{f}(\mathbf{x})))$ as $\mathbf{y}_k(\mathbf{f}(\mathbf{x})) = \sum_{i=1}^{k} \hat{\Psi}_i(\mathbf{f}(\mathbf{x})),$ for $i = 1,2,\ldots,m$. In the following, we adopt an effective modeling technique for quantities $\mathbf{y}_k(\mathbf{f}(\mathbf{x}))$ with arbitrary i. In [9], for a given outcome vector

$\mathbf{f}(\mathbf{x})$ the quantity $\mathbf{y}_k(\mathbf{f}(\mathbf{x}))$ may be found by solving the following linear program:

$$\begin{aligned} \mathbf{y}_k(\mathbf{f}(\mathbf{x})) = \quad & \max\ kt_k - \sum_{i=1}^{m} d_i \\ \text{subject to} \quad & t_k - f_i(\mathbf{x}) \leq d_i,\ i = 1, \ldots, m \\ & d_i \geq 0,\ i = 1, \ldots, m \end{aligned} \tag{3.3}$$

where t_k is an unrestricted variable when nonnegative variables d_i represent their downside deviations from the value of t_k for several values $f_i(\mathbf{x})$. Taking an example, the simplest outcome may be defined by the following optimization:

$$\mathbf{y}_1(\mathbf{f}(\mathbf{x})) = \max\ \{t_1 : t_1 \leq f_i(\mathbf{x}) \text{ for } i = 1, \ldots, m\}$$

where t_1 is an unrestricted variable.

Formula (3.3) provides us with a computational formulation for the worst conditional mean $M_{\frac{k}{m}}(\mathbf{f}(\mathbf{x}))$ defined as the mean outcome for the k worst-off services, i.e.,

$$M_{\frac{k}{m}}(\mathbf{f}(\mathbf{x})) = \frac{1}{k}\mathbf{y}_k(\mathbf{f}(\mathbf{x})),\ \text{for } k = 1, \ldots, m.$$

For $k = 1$, $M_{\frac{1}{m}}(\mathbf{f}(\mathbf{x})) = \mathbf{y}_1(\mathbf{f}(\mathbf{x})) = \hat{\Psi}_1(\mathbf{f}(\mathbf{x}))$ represents the minimum outcome and for $k = m$, $M_{\frac{m}{m}}(\mathbf{f}(\mathbf{x})) = \frac{1}{m}\mathbf{y}_m(\mathbf{f}(\mathbf{x})) = \frac{1}{m}\sum_{k=1}^{m} \hat{\Psi}_k(\mathbf{f}(\mathbf{x})) = \frac{1}{m}\sum_{i=1}^{m} f_i(\mathbf{x})$ represents the mean outcome.

For modeling various fair preferences, one may use some combinations of the cumulative ordered outcomes $\mathbf{y}_k(\mathbf{f}(\mathbf{x}))$. In specific, for $v_k \geq 0$, the weighted sum is

$$\sum_{k=1}^{m} v_k \mathbf{y}_k(\mathbf{f}(\mathbf{x})). \tag{3.4}$$

Note that, due to the definition of map $\mathbf{y}_k$, the above function can be expressed in the form with ordered weights $\hat{w}_k = \sum_{j=k}^{m} v_j$ $(k = 1, \ldots, m)$ allocated to coordinates of the ordered outcome vector. When substituting v_k with $\hat{w}_k$ where $\hat{w}_h$ is an ordered weight, (3.4) becomes $\sum_{h=1}^{m} \hat{w}_k \hat{\Psi}_k(\mathbf{f}(\mathbf{x}))$, where $\sum_{k=1}^{m} \hat{w}_k = \sum_{i=1}^{m} w_i = 1$ and $\hat{w}_k \geq 0$, $\forall k = 1, \ldots, m$.

Applying the Ordered Weighted Averaging Method to problem (2.21), we get

$$\max\ \{\sum_{k=1}^{m} \hat{w}_k \hat{\Psi}_k(\mathbf{f}(\mathbf{x})) : \mathbf{x} \in Q^*\} \tag{3.5}$$

where (3.5) becomes $\sum_{i=1}^{m} w_i f_i(\mathbf{x})$. If ordered weights $\hat{w}_k$ are decreasing and non-negative, that is $\hat{w}_1 \geq \hat{w}_2 \geq \ldots \geq \hat{w}_{m-1} \geq \hat{w}_m \geq 0$, then each optimal solution of the problem (3.5) is a fair solution of (2.21). Actually, formulas (3.3) and (3.4) allow us to formulate the following mathematical programming of the original multiple criteria problem:

$$
\begin{aligned}
\text{(Model II)} \quad \text{maximize} \quad & \sum_{k=1}^{m} v_k \mathbf{y}_k \\
\text{subject to} \quad & \mathbf{y}_k = k t_k - \sum_{i=1}^{m} d_{ki}, \ \forall k = 1, \ldots, m \\
& t_k - d_{ki} \leq f_i(\mathbf{x}), \ \forall i, \ k = 1, \ldots, m \\
& d_{ki} \geq 0, \ \forall i, \ k = 1, \ldots, m \\
& t_k \text{ unrestricted}, \ \forall k = 1, \ldots, m \\
& \sum_{k=1}^{m} k v_k = 1 \\
& v_k \geq 0, \forall k = 1, \ldots, m \\
& \mathbf{x} \in Q^*.
\end{aligned}
\tag{3.6}
$$

Decision variables in Model II are $\mathbf{x} = (q_i, \chi_{i,j}(e))$, t_k, d_{ki}, and v_k, for all $i, k \in I$, $j \in \mathcal{K}_i$.

Where $v_m = w_m$, $v_k = \hat{w}_k - \hat{w}_{k+1}$ for $k = 1, \ldots, m-1$, $\hat{w}_k \in [0, 1]$ for each k, and $\sum_{k=1}^{m} \hat{w}_k = 1$. The individual function $\mathbf{y}_k$ is the first k sum of the ordered multiple objective functions $\hat{\boldsymbol{\Psi}}(\mathbf{f}(\mathbf{x}))$ in the allocation pattern $\mathbf{x} \in Q^*$.

In this work, we use solver BARON in software GAMS (see [11] and [12]) to maximize the weighted sum of logarithms of the bandwidth for each class i, $i = 1, \ldots, m$. First of all, we carry on changing the parameter J_i to observe the variations of q_i, w_i and total utilization value. In the next step, we keep on changing the parameter B, a_i, r_i to observe the variations and see what affects the constraints about B, a_i, r_i. All numerical results are given in Chapter 4.