GENERALIZED FUZZY INTEGER SHARING PROBLEMS WITH FUZZY CAPACITY CONSTRAINTS

YUE GE¹, HIROAKI ISHII² AND MINGHAO CHEN³

¹,³Department of Mathematics
Harbin Institute of Technology
Harbin 150001, P.R. China
¹gyue2001@hotmail.com
³chenmh130264@yahoo.com.cn

²School of Science and Technology
Kwansei Gakuin University
Hyogo 669-1337, Japan
ishiihiroaki@kwansei.ac.jp

ABSTRACT. The sharing problem is a method for finding an equitable distribution of resources by maximizing the smallest value of all tradeoff functions, where a tradeoff function is a function of the flux to a sink node. In this paper, we propose a generalized fuzzy integer sharing problem with fuzzy capacity constraints. Our model has two criteria, i.e., minimal satisfaction among all fuzzy capacity constraints and among fluxes to all sink nodes, both of which are to be maximized. Because it is not usually possible to perform optimal flow pattern maximizing for two objectives at a time, we propose an algorithm for finding non-dominated flow patterns after defining non-domination. We finally determine the time complexity of the algorithm.

Keywords: Sharing Problem; Integer Flow; Fuzzy Capacity; Bi-criteria; Non-dominated Flow Pattern

1. Introduction. Until now, many network flow problems have been well studied. But recently, some extensions have been done. Jansen (2006) proposed an approximation algorithm for the general max-min resource sharing problem with nonnegative concave constraints on a convex set. Theory and algorithms for solving the multiple objective minimum cost flow problem have been reviewed by Hamacher et al. (2007). Nace et al. (2008) provided a study of max-min fair multi-commodity flows. Lin (2007) discussed the system capacity for a two-commodity multistate flow network with unreliable nodes and capacity weight.

This paper proposes a new sharing problem. Sharing problem, originated by Brown (1979), is a method to find an equitable distribution of resources. Its objective is to maximize the smallest value of all tradeoff function where a tradeoff function is a function of the flux to a sink node (i.e., demand point), and which took a weight for the sink node into account. The sharing problem assumes that weights are constant. However, in actuality, it is difficult but essential to determine the values. Thus, it is proper to think them with
flexibility. For instance, about a distribution of goods under a disaster, the information about the number of victims in a shelter is vague because of the emergence. Then, to improve the matter, Tada et al. (1989) took account of a membership function of the fuzzy weight and considered a fuzzy sharing problem, each membership function designates the satisfaction degree of the flux to each sink node. In some situations, the equitable distribution should be done under the constraint that the received amount of resources at each sink node is a multiple of some block-unit (e.g., a dozen), thus, the author (Tada et al., 1989) considered a generalized version of the fuzzy sharing problem under the constraint that the received amount at each sink node is restricted to be multiple of some certain positive integer value. To deal with the case that the violation of capacity constraints of distribution network arcs is acceptable in a certain range, Ishii and Itoh (1996) considered a fuzzy integer sharing problem with fuzzy capacity constraints (that is, upper limit of capacity for each arc is flexible), this model is bi-criteria one, bi-criteria are minimal satisfaction degree among all fuzzy capacity constraints and that among fluxes to all sink nodes, both of which are to be maximized. Besides, in some situation, the flow passing through each arc is also a multiple of some block-unit. In actual situation, all the above conditions may be happened simultaneously, we need to find a distribution method has a ‘possibility large satisfaction degree of demander’ and which simultaneously in the ‘possibility best way’ satisfies the capacity constraints. Base on this, we considered a fuzzy generalized integer sharing problem with fuzzy capacity constraints such that the received amount at each sink node and the flow passing through each arc are restricted to be multiple of some certain positive integer values.

Section 2 formulates our problem and defines non-dominated flow pattern since usually there does not exist an optimal flow pattern maximizing two objectives at a time. Section 3 proposes a solution algorithm to find some non-dominated flow patterns and clarifies its validity. Section 4 discusses its complexity. Section 5 shows how our algorithm runs by an example. Finally, section 6 concludes this paper and discusses further research problems.

2. Problem Formulation. Let 

\[ G = (N, A) \]

be a distribution network where \( N \) is the set of nodes including special nodes, called source nodes (i.e., distribution points) and sink nodes (i.e., demand points) and \( A \) the set of directed arcs connecting nodes. Let \( S, T \) be the set of source nodes and sink nodes, respectively. Further, we add the super source node \( \sigma \) and super sink node \( u \) with the arc set \( \{(\sigma, s) \mid s \in S\} \) and \( \{(t, u) \mid t \in T\} \) to

\[ G' = (N', A') = (N \cup \{\sigma\} \cup \{u\}, A \cup \{(\sigma, s) \mid s \in S\} \cup \{(t, u) \mid t \in T\}). \]

Let \( f_{ij} \) denote the flow value in arc \((i, j)\), and we use a simplified notation \( f_t = f_{wu} \) for \( t \in T \). Each arc \((i, j) \in A\) has a fuzzy capacity with the membership function:

\[
\mu_{ij}(f_{ij}) = \begin{cases} 
1 & (f_{ij} \leq c_{ij}) \\
\frac{c_{ij} - f_{ij}}{c_{ij} - \bar{c}_{ij}} & (c_{ij} < f_{ij} < \bar{c}_{ij}) \\
0 & (f_{ij} \geq \bar{c}_{ij}) 
\end{cases}
\]
where $c_y < \bar{c}_y$ and $c_y, \bar{c}_y$ are integers.

We assume the capacity of each arc $(\sigma, s) (s \in S)$ is $C(\sigma, s) = \infty (s \in S)$. While, the capacity of each arc $(t, u) (t \in T)$, denoted as $C(t, u) (t \in T)$, is one of the key points and it may be determined and updated as to maximize the second criteria under the membership functions $\mu_i(f_i)$ (which is simplified notations of $\mu_{w_i}(f_i)$) characterizing the fuzzy weights $w_i$ of sink node $t \in T$ given as follows.

$$
\mu_i(f_i) = \begin{cases} 
0 & (f_i \leq a_i) \\
\frac{f_i - a_i}{b_i - a_i} & (a_i < f_i < b_i) \\
1 & (f_i \geq b_i)
\end{cases}
$$

where $a_i < b_i$ and $a_i, b_i$ are integers. Each membership function designates the satisfaction degree of flux to the sink node.

Further we restrict flow values $f_i$ to be nonnegative integer. In some situations, the equitable distribution should be under the constraint that the received amount of resources at each sink node is a multiple of some block-unit (e.g., a dozen). Thus, we assume that for each sink node $t \in T$, the received amount be a multiple of a certain positive integer $d_i$ (we call it $d_i$-multiple), i.e., $f_i \equiv 0(\text{mod } d_i), t \in T$. Further, the flow passing through each arc is also a multiple of some block-unit, thus we assume $f_{ij} \equiv 0(\text{mod } d), (i, j) \in A, d$ is also a certain positive integer. In order to assure the feasibility, we assume that each $d_i (t \in T)$ is a multiple of $d$, that is, $d_i = k_i d (t \in T)$, where $k_i$ is positive integer, $t \in T$.

Under the above setting, we consider the bi-criteria, i.e., minimal satisfaction degree among all fuzzy capacity constraints and that among fluxes to all sink nodes, both of which are to be maximized. Then the fuzzy generalized integer sharing problem with fuzzy capacity constraints is formulated as follows.

**P:** Maximize $\min_{(i, j) \in A} \mu_y(f_{ij})$, Maximize $\min_{t \in T} \mu_t(f_t)$

subject to $\sum_{i \in N^\prime \setminus \{u\}} f_{ij} = \sum_{j \in N} f_{jk}, j \in N$

$f_{ij} \equiv 0(\text{mod } d), (i, j) \in A, f_t \equiv 0(\text{mod } d_i), t \in T$

$f_y : \text{nonnegative integer}, (i, j) \in A$

Next we define a flow pattern vector of flow pattern $f = (f_y)$ to be

$$v_f = (f^1, f^2) = (\min_{(i, j) \in A} \mu_y(f_{ij}), \min_{t \in T} \mu_t(f_t)).$$

Generally speaking, optimal flow pattern maximizing two objectives at a time does not exist and so we seek non-dominated flow patterns which definition is given as follows.

**Definition 2.1.** For two flow patterns $f_a$ and $f_b$, if $f^1_a \geq f^1_b$, $f^2_a \geq f^2_b$ and $(f^1_a, f^2_a) \neq (f^1_b, f^2_b)$, we say $f_a$ dominates $f_b$. And, if there exists no flow pattern dominating $f$, $f$ is called non-dominated flow pattern.
3. Solution Procedure. Since all flow values $f_{ij}$, $f_i$ are integer, we only need to consider integer capacity values.

Let $\nu(1)^*$ and $\nu(0)^*$ be the total desired amount of supply value under the capacity $c_{ij}$ and $\overline{c}_{ij}$, respectively. First we solve the following fuzzy sharing problem P(1) and P(0), find the non-dominated flow pattern $f(1)$ and $f(0)$ whose corresponding flow pattern vector has the value 1 and 0 as the first component, respectively.

P(1): \[
\text{Maximize } \min_{i \in T} \mu_i(f_i) \\
\text{subject to } \sum_{i \in T} f_i = \nu(1)^*, \sum_{i \in N - \{u\}} f_{ij} = \sum_{k \in N - \{\sigma\}} f_{jk}, j \in N \\
f_{ij} \equiv 0(\text{mod } d), (i, j) \in A, f_i \equiv 0(\text{mod } d), t \in T \\
0 \leq f_{ij} \leq c_{ij}, (i, j) \in A
\]

P(0): \[
\text{Maximize } \min_{i \in T} \mu_i(f_i) \\
\text{subject to } \sum_{i \in T} f_i = \nu(0)^*, \sum_{i \in N - \{u\}} f_{ij} = \sum_{k \in N - \{\sigma\}} f_{jk}, j \in N \\
f_{ij} \equiv 0(\text{mod } d), (i, j) \in A, f_i \equiv 0(\text{mod } d), t \in T \\
0 \leq f_{ij} \leq \overline{c}_{ij}, (i, j) \in A
\]

Set $f_{ij}' = f_{ij} / d$, $(i, j) \in A$, $f_i' = f_i / d$, $t \in T$, then in order to solve P(1), we only need to solve the following problem P(1)'.

P(1)': \[
\text{Maximize } \min_{i \in T} \mu_i(f_i') \\
\text{subject to } \sum_{i \in T} f_i' = \nu(1)^*?, \sum_{i \in N - \{u\}} f_{ij}' = \sum_{k \in N - \{\sigma\}} f_{jk}', j \in N \\
f_i' \equiv 0(\text{mod } k), t \in T \\
0 \leq f_{ij}' \leq c_{ij}' (= [c_{ij} / d]), (i, j) \in A
\]

where $\mu_i'(f_i') = \begin{cases} 
0 & (f_i' \leq a_i') \\
\frac{f_i' - a_i'}{b_i' - a_i'} & (a_i' < f_i' < b_i'), \quad a_i' = a_i / d, \quad b_i' = b_i / d \quad \text{and} \quad \nu(1)^*? = \nu(1)^* / d \\
1 & (f_i' \geq b_i')
\end{cases}$

We use P(1)'' to denote an auxiliary problem of P(1)' as following and give the algorithm to solve P(1)'.

P(1)'': \[
\text{Maximize } \min_{i \in T} \mu_i'(f_i') \\
\text{subject to } \sum_{i \in T} f_i'' = \nu(1)^*?, \sum_{i \in N - \{u\}} f_{ij}'' = \sum_{k \in N - \{\sigma\}} f_{jk}'', j \in N \\
0 \leq f_{ij}'' \leq c_{ij}', (i, j) \in A
\]

Algorithm for P(1)'

Step 1. By using the algorithm similar to Algorithm FSP by Tada et al. (1989) for integer case (only let $C'(t, u) \cup C(t, u) / d$ be integer in each step), solve the fuzzy sharing problem P(1)'' If $f_i'' \equiv 0(\text{mod } k_i)$ for all $t \in T$, then terminate, the current flow is optimal.
Otherwise, set \( i = 1 \) and \( C'_i(t,u) = \left\lfloor \frac{C'_0(t,u)}{k_j} \right\rfloor \cdot k_j, t \in T \), where \( C'_0(t,u) \) denotes the capacity from \( t \) to \( u \) in the final network realized by Algorithm of \( P(1)' \), then go to Step 2.

Step 2. Find a maximum flow \( f'_i \) from \( \sigma \) to \( u \) and its value \( v'_i \), then go to Step 3.

Step 3. If \( v'_i < v(1)' \), go to Step 4. Otherwise, terminate, the current flow is optimal.

Step 4. Denote \( F_i \) to be the set of \( t \in T \) such that there exists an augmenting path from \( \sigma \) to \( u \) via \( t \) when \( C'_i(t,u) \) for every \( t \in T \) is updated to \( C'_i(t,u) = C'_i(t,u) + k_j \). If \( F_i \) is empty, then terminate, it is infeasible. Otherwise, find a sink node \( \tilde{t} \in F_i \) such that \( \mu'_i(C'_i(\tilde{t},u)) = \min_{t \in F_i} \mu'_i(C'_i(t,u)) \), and set \( C'_{i+1}(\tilde{t},u) = C'_i(\tilde{t},u) + k_j \), then go to Step 5.

Step 5. Set \( i = i + 1 \) and return to Step 2.

It is obvious that the Algorithm is valid and so we discuss the complexity of the Algorithm.

**Lemma 3.1.** \( v(1)' - v'_1 \leq \sum_{t \in T} (k_t - 1) \).

Proof: Let \( f''_i \) be the flux to sink node \( t \) in case the flow is optimal without \( k_i \)-multiple constraint. Then
\[
v(1)' - v'_1 = \sum_{t \in T} f''_i - \left( \sum_{t \in T} \left\lfloor \frac{C'_0(t,u)}{k_j} \right\rfloor \cdot k_j \right) = \sum_{t \in T} \left( f''_i - \left\lfloor \frac{C'_0(t,u)}{k_j} \right\rfloor \cdot k_j \right) \leq \sum_{t \in T} (k_t - 1).
\]

**Lemma 3.2.** If \( v'_i < v(1)' \), then \( v'_{i+1} = v'_i + k_j, F_{i+1} \subseteq F_i \), where \( \tilde{t} \) is defined in Step 4 of the above algorithm.

Proof: It is clear from the definition of \( F_i \).

**Theorem 3.1.** The time complexity of \( P(1)' \) is \( O(R \cdot T \cdot cf(n,m)) \), where \( n = |N|, m = |A|, R = \left\lceil \max_{t \in T} k_i / \min_{t \in T} k_j \right\rceil \), \( cf(n,m) \) is the time bound of the maximum flow problem for a graph \( (N,A) \) (Iri, 1979).

Proof: Step 1 takes at most \( O(|T| \cdot cf(n,m)) \) operations. For the iterations from Step 2 to Step 5, Step 2 is \( O(cf(n,m)) \) and Step 4 is \( O(T \cdot cf(n,m)) \). From the above lemmas, it is clear that the Algorithm for \( P(1)' \) takes at most \( R \cdot T \cdot \left\lceil \max_{t \in T} k_i / \min_{t \in T} k_j \right\rceil \) times iterations. Thus, the Algorithm for \( P(1)' \) takes at most \( O(R \cdot T \cdot cf(n,m)) \).

We denote the optimal flow pattern and optimal value of \( P(1)' \) be \( f'(1) \) and \( f'(1)^2 \), respectively. So it is obvious that the optimal flow pattern and optimal value of \( P(1) \) are \( d \cdot f'(1) \) and \( f'(1)^2 \), respectively, i.e., \( f(1) = d \cdot f'(1) \), \( f(1)^2 = f'(1)^2 \). It is similar for the optimal flow pattern \( f(0) \) and optimal value \( f(0)^2 \) of \( P(0) \).

Now sorting \( \mu_y(kd), k \) is integer and \( k \in (c'_y, c'_y]), (i,j) \in A \), here \( c'_y = \lceil c_y / d \rceil \), and let the result be \( 1 \equiv \mu^0 > \mu^1 > \cdots > \mu^l > \mu^{l+1} \equiv 0 \) ( \( l \) is the number of different \( \mu_y(kd) \in (0,1) \)).

Let \( v(\mu^g) \) be the total desired amount of supply value under the capacity
\[ c^q_j = (1 - \mu^q) \bar{c}_j + \mu^q c^q_j ), \quad q = 1, \ldots, l \]. We use \( P(\mu^q) \) \((q = 1, \ldots, l)\) to denote the following problem.

\[
P(\mu^q): \text{Maximize } \min_{i \in T} \mu_i(f_i) \]
subject to
\[
\sum_{i \in T} f_i = v(\mu^q)^*, \quad \sum_{i \in N - \{u\}} f_{ij} = \sum_{k \in N - \{\sigma\}} f_{jk}, \quad j \in N
\]
\[
f_{ij} \equiv 0 \text{ (mod } d_i\text{)}, \quad (i, j) \in A, \quad f_{i} \equiv 0 \text{ (mod } d_i\text{)}, \quad t \in T
\]
\[
0 \leq f_{ij} \leq c^q_j, \quad (i, j) \in A
\]

By the solution procedure similar to problem \( P(1) \), solve the problem \( P(\mu^q) \). Let an optimal flow pattern and the optimal value of \( P(\mu^q) \) be \( f(\mu^q) \) and \( f(\mu^q)^2 \), respectively.

Next, we give the algorithm for \( P \) as follows.

Algorithm for \( P \)

Step 1. Set \( q = 1 \), \( DS = \{f(1)\} \) and \( DV = \{(1, f(1)^2)\} \), then go to Step 2.

Step 2. Solve \( P(\mu^q) \). If \( f(\mu^q) \) is dominated by some flow pattern of \( DS \), then go to Step 3 directly. Otherwise, set \( DS = DS \cup \{f(\mu^q)\} \) and \( DV = DV \cup \{(\mu^q, f(\mu^q)^2)\} \), then go to Step 3.

Step 3. Set \( q = q + 1 \). If \( q \neq l + 1 \), then return to Step 2. Otherwise, check whether \( f(0) \) is dominated by some flow pattern of \( DS \). If dominated, terminate. Otherwise, set \( DS = DS \cup \{f(0)\} \) and \( DV = DV \cup \{(0, f(0)^2)\} \), terminate.

Validity of the above algorithm is clear from the fact that it check all possibilities of the first component of non-dominated flow pattern vectors and the greater the flow value sent from \( \sigma \) to \( u \) in \( G' \) then so is \( \min_{i \in T} \mu_i(f_i) \).

4. Complexity of the Algorithm for \( P \). The time complexity of the Algorithm for \( P \) is as follows.

**Theorem 4.1.** The Algorithm for \( P \) obtains non-dominated flow patterns in at most \( O(L \cdot \max(\log L, R \cdot |T|^2 \cdot cf(n, m))) \) computational times, where \( L = \sum_{(i, j) \in A} (\bar{c}_j - c^q_j) / d \).

Proof: Solving \( P(0) \) and \( P(1) \) both takes at most \( O(R \cdot |T|^2 \cdot cf(n, m)) \) operations. Because we treating integer flow, then \( l \leq L \) holds. Therefore, sorting \( \mu^0 , \mu^1, \ldots, \mu^{l+1} \) takes at most \( O(L \log L) \) operations. On Step 2, solving \( P(\mu^q) \) takes at most \( O(R \cdot |T|^2 \cdot cf(n, m)) \) operations, check whether there exists a flow pattern in \( DS \) which dominates \( f(\mu^q) \) or not needs at most \( O(L) \) operations. On Step 3 without going to Step 2, judging whether there exists a flow pattern in \( DS \) which dominates \( f(0) \) or not needs at most \( O(L) \) operations. As Step 2 to Step 3 is repeated at most \( L \) times, solving \( P(\mu^q) \) for all \( \mu^q \) takes \( O(LR \cdot |T|^2 \cdot cf(n, m)) \) computational times. Therefore, the total complexity is \( O(\max(L \log L, LR \cdot |T|^2 \cdot cf(n, m))) = O(L \cdot \max(\log L, R \cdot |T|^2 \cdot cf(n, m))) \).
5. Numerical Example. In this section, we show how our algorithm runs by an example.

We consider the extended network shown in FIGURE 1, where \( S = \{1, 2\} \), \( T = \{4, 5, 6\} \) and \( A = \{(1, 3), (2, 3), (2, 5), (2, 6), (3, 4), (3, 5)\} \).

![FIGURE 1. Initial network](image)

For each arc \((i, j) \in A\), fuzzy capacity is given as follows:

\[
\mu_{ij}(f_{ij}) = \begin{cases} 
1 & (f_{ij} \leq 5) \\
\frac{10 - f_{ij}}{5} & (5 < f_{ij} < 10) \\
0 & (f_{ij} \geq 10)
\end{cases}
\]

\[
\mu_{26}(f_{26}) = \begin{cases} 
1 & (f_{26} \leq 2) \\
\frac{7 - f_{26}}{5} & (2 < f_{26} < 7) \\
0 & (f_{26} \geq 7)
\end{cases}
\]

Besides, for each sink node \( t \in T \), fuzzy weight is given as follows:

\[
\mu_{t}(f_{t}) = \begin{cases} 
0 & (f_{t} \leq 0) \\
\frac{f_{t}}{10} & (0 < f_{t} < 10) \\
1 & (f_{t} \geq 10)
\end{cases}
\]

The network with capacity \( c_{ij} \) and \( \overline{c}_{ij} \) for 1 and 0 are shown in FIGURE 2 and FIGURE 3, respectively, where the left number attached to each arc denotes its capacity and the right one the current flux through it hereafter.

![FIGURE 2. Network with capacity \( c_{ij} \)](image)

![FIGURE 3. Network with capacity \( \overline{c}_{ij} \)](image)

Let \( d = 3 \), \( d_4 = 3 \), \( d_5 = 6 \), \( d_6 = 3 \), then \( k_4 = 1 \), \( k_5 = 2 \), \( k_6 = 1 \) and

\[
\mu_{*}(f_{*}) = \begin{cases} 
0 & (f_{*} \leq 0) \\
\frac{f_{*}'}{\delta_{*}} & (0 < f_{*}' < \delta_{*}) \\
1 & (f_{*}' \geq \delta_{*})
\end{cases}
\]

\[
\mu_{*}'(f_{*}') = \begin{cases} 
0 & (f_{*}' \leq \gamma_{*}) \\
\frac{f_{*}'' - \gamma_{*}}{\gamma_{*}} & (\gamma_{*} < f_{*}' < \gamma_{*}) \\
1 & (f_{*}' \geq \gamma_{*})
\end{cases}
\]

\[
\mu_{*}''(f_{*}'') = \begin{cases} 
0 & (f_{*}'' \leq \gamma_{*}) \\
\frac{f_{*}'' - \gamma_{*}}{\gamma_{*}} & (\gamma_{*} < f_{*}' < \gamma_{*}) \\
1 & (f_{*}' \geq \gamma_{*})
\end{cases}
\]
Figure 4 and Figure 5 show the network with capacity $c_{ij}'$ and $c_{ij}'$, respectively.

Now sorting $\mu_i(kd)$, $k$ is integer and $k \in (c_{ij}', c_{ij}]$, $(i, j) \in A$, and let the result be $1 \equiv \mu^0 > \mu^1 = 0.8 > \mu^2 = 0.75 > \mu^3 = 0.5 > \mu^4 = 0.25 > \mu^5 = 0.2 > \mu^6 \equiv 0$, $l = 5$.

Let $v(1)^* = 3$, $v(0.8)^* = 6$, $v(0.75)^* = 12$, $v(0.5)^* = 12$, $v(0.25)^* = 15$, $v(0.2)^* = 18$, $v(0)^* = 18$. Next solving problem $P(1)''$ and $P(0)'$.

Algorithm for $P(1)'$' performs as follows:
Step 1. Since $\mu = 1 - \frac{3 + 3}{2} = -\frac{1}{18} < 0$, terminate, the optimal value is 0.

It is obvious that the optimal value for $P(1)$ is 0.

Algorithm for $P(0)'$' performs as follows:
The first iteration, $i = 1$.

Step 1. Since $\mu = \frac{9}{3} + 1 + \frac{5}{3} > 0$, each capacity is set as follows:

$C(4, u) = \left[\frac{7}{9} \cdot 1 + \frac{2}{3} = 1, \quad C(6, u) = \left[\frac{7}{9} \cdot 1 + \frac{2}{3} = 1, \quad C(6, u) = \left[\frac{7}{9} \cdot 1 + \frac{2}{3} = 1.$

Step 2. The result of max-flow computation is shown in FIGURE 6.

Step 3. Since $v_i = 4 < 6 = v(0)'$, go to Step 4.

Step 4. Since $\bar{X}_1 \cap T = \left\{4\right\} \neq \emptyset$, it follows that $F_{\bar{x}_1} = 2$. Thus, $\mu_{x_1} = \frac{2 - 0}{10 / 3 + 1 + 5 / 3} = \frac{3}{5}$, $\mu_{x_1} = \frac{6 - 2 - (2 / 3 + 2 / 3)}{1 + 5 / 3} = 1$, and each capacity is reset as follows: $C'(4, u) = \left[\frac{3}{5} \cdot 1 + \frac{2}{3} = 1, \quad C'(6, u) = \left[\frac{3}{5} \cdot 1 + \frac{2}{3} = 1.$

FIGURE 6. The first association network for $P(0)'$'

FIGURE 7. The second association network for $P(0)''$
2, \( C'(5, u) = [1 + \frac{2}{3}] = 1, \) \( C'(6, u) = \frac{5}{3} + \frac{2}{3} = 2.\) Return to Step 2.

The second iteration, \( i = 2.\) FIGURE 7 shows the network obtained in Step 2.

Step 3. Since \( v_2 = 5 < 6 = v(0)^*,\) go to Step 5.

Step 5. Since \( X_1 \cap X_2 \cap T = \{4\},\) it follows that \( F_{X_1 \cap X_2} = 2.\) Thus, \( \mu_{X_1 \cap X_2} = \frac{2 - 2}{2} = \frac{0}{2} = 0.\) and \( C'(4, u) = \left[ \frac{3}{5}, \frac{1}{5}, \frac{2}{5} \right] = 2.\) Also since \( X_1 \cap X_2 \cap T = \{6\},\) we have \( F_{X_1 \cap X_2} = 2.\) Thus \( \mu_{X_1 \cap X_2} = \frac{6 - 2 - 2}{2} = \frac{2}{2} = 1, \) and \( C'(6, u) = \left[ \frac{4}{5}, \frac{5}{5}, \frac{2}{3} \right] = 2, \) \( C'(5, u) = \left[ \frac{4}{3}, \frac{1}{3}, \frac{2}{3} \right] = 2.\) Return to Step 2.

The third iteration, \( i = 3.\) FIGURE 8 shows the network obtained in Step 2.

Step 3. Since \( v_3 = 6 = v(0)^*\), the current flow is optimal. \( \mu'_4(f'_4) = \mu'_4(2) = 0.6 , \)
\( \mu'_4(f'_5) = \mu'_5(2) = 1, \) \( \mu'_6(f'_2) = \mu'_6(2) = 0.8,\) therefore maximize(min \( \mu'_4(f'_4) = 0.6).\)

Since \( f'_4 = 2 \equiv 0(\text{mod } 1), \) \( f'_5 = 2 \equiv 0(\text{mod } 2), \) \( f'_6 = 2 \equiv 0(\text{mod } 1),\) the current flow is optimal for \( P(0)^*\). Further, the optimal flow pattern \( f(0)\) of \( P(0)\) is given as follows:
\( f_{13} = f_{25} = f_{26} = f_{34} = 6, f_{23} = f_{35} = 0\) and the optimal value is 0.6.

Finally, we solve the problem \( P.\)

Algorithm for \( P\) performs as follows:

Step 1. Set \( q = 1,\) \( DS = \{f(1)\}\) and \( DV = \{(1,0)\},\) then go to Step 2.

The first iteration. Step 2. Solve \( P(0.8).\) The optimal flow pattern \( f(0.8)\) is given as follows:
\( f_{13} = f_{26} = f_{25} = 3, f_{23} = f_{25} = f_{35} = 0\) and optimal value is 0. It is obvious that \( f(0.8)\) is dominated by \( f(1),\) so go to Step 3 directly.

Step 3. Set \( q = 2.\) Since \( q \neq 6,\) then return to Step 2.

The second iteration. Step 2. Solve \( P(0.75).\) The optimal flow pattern \( f(0.75)\) is given as follows:
\( f_{13} = 6, f_{23} = 0, f_{25} = f_{26} = f_{34} = f_{35} = 3\) and optimal value is 0.2. It is obvious that \( f(0.75)\) is not dominated by \( f(1),\) set \( DS = DS \cup \{f(0.75)\} = \{f(1), f(0.75)\}\) and \( DV = DV \cup \{(0.75,0.2)\},\) then go to Step 3.

Step 3. Set \( q = 3.\) Since \( q \neq 6,\) then return to Step 2.

The third iteration. Step 2. Solve \( P(0.5).\) The optimal flow pattern \( f(0.5)\) is given as follows:
\( f_{13} = f_{23} = f_{25} = f_{26} = f_{34} = f_{35} = 3\) and optimal value is 0.2. It is obvious that

\[ \begin{align*}
\mu_{X_1 \cap X_2} &= \frac{2 - 2}{2} = \frac{0}{2} = 0, \\
C'(4, u) &= \left[ \frac{3}{5}, \frac{1}{5}, \frac{2}{5} \right] = 2, \\
C'(5, u) &= \left[ \frac{4}{3}, \frac{1}{3}, \frac{2}{3} \right] = 2.
\end{align*} \]
\( f(0.5) \) is dominated by \( f(0.75) \), so go to Step 3 directly.

Step 3. Set \( q = 4 \). Since \( q \neq 6 \), then return to Step 2.

The fourth iteration. Step 2. Solve \( P(0.25) \). The optimal flow pattern \( f(0.25) \) is given as follows: \( f_{13} = f_{34} = 6, f_{23} = f_{25} = f_{26} = f_{35} = 3 \) and optimal value is 0.2. It is obvious that \( f(0.25) \) is dominated by \( f(0.75) \), so go to Step 3 directly.

Step 3. Set \( q = 5 \). Since \( q \neq 6 \), then return to Step 2.

The fifth iteration. Step 2. Solve \( P(0.2) \). The optimal flow pattern \( f(0.2) \) is given as follows: \( f_{13} = f_{26} = f_{34} = 6, f_{23} = f_{25} = f_{35} = 3 \) and optimal value is 0.6. It is obvious that \( f(0.2) \) is not dominated by any flow pattern of \( DS \), so set \( DS = DS \cup \{f(0.2)\} = \{f(1), f(0.75), f(0.2)\} \) and \( DV = DV \cup \{(0.2,0.6)\} = \{(1,0),(0.75,0.2),(0.2,0.6)\} \), then go to Step 3.

Step 3. Set \( q = 6 \). Since \( f(0) \) is dominated by \( f(0.2) \), terminate.

6. Conclusions. In this paper, we proposed a fuzzy generalized integer sharing problem with fuzzy capacity constraints and developed an efficient algorithm to find non-dominated solutions, further, showed how our algorithm runs by an example. There may be many non-dominated flow patterns with some corresponding flow pattern vector but our algorithm only find one of them. Further though in the worst case there exists \( O(l) \) non-dominated flow patterns, we should refine the algorithm by taking not such a case into account since the algorithm check all possibilities in this case also. These are further research problems. Another is extend the problem to more general case that for each arc, flow value is restricted to multiple of their own certain positive integer. Besides, we think generalized case of sharing theory may be fruitful and interesting. Especially so more general models of fuzzy sharing problem reflecting actual situations are to be solved.

REFERENCES


