Hybrid Unicast and Multicast Flow Control: 
A Max-Min Fairness Primal Optimization Approach

Homayoun Yousefi’zadeh  Fatemeh Fazel  Hamid Jafarkhani

Department of Electrical Engineering and Computer Science
University of California, Irvine
[hyousefi,fazel,hamidj]@uci.edu

Abstract—In this paper, we present an analytical solution to the general problem of flow control for both unicast and multicast IP networks. Relying on the so-called max-min flow fairness metric, we formulate a pair of centralized and decentralized convex optimization problems that can be analytically solved with quadratic and linear complexities respectively. Utilizing the solution to the decentralized optimization problem, we then propose a flow control algorithm requiring no per flow state information.

Index Terms—Unicast IP Networks, Multicast IP Networks, Heterogeneity, Inter-Session Fairness, Flow Control, Flow Priority, Max-Min Fairness, Optimality.

I. INTRODUCTION

In the past decade, multicasting techniques have been in widespread use for communication networking applications as efficient means of network resource sharing. However, utilizing multicasting techniques has introduced significant technical challenges at different levels. Enforcing inter-session (flow) fairness among a set of competing flows is one of the most important challenges of utilizing multicasting techniques. For the lack of any built-in flow fairness support in UDP and due to the fact that multicast sessions are typically built on top of UDP, achieving flow fairness in hybrid unicast and multicast networks is in fact a complex task.

In what follows, we briefly review related flow control work in the context of our current research work. The original TCP flow control was discussed by Jacobson [12] and further enhanced by Floyd et al. [7]. In the recent years, proposition of ECN (Explicit Congestion Notification) marking techniques proposed by Ramakrishnan et al. in [21] and by Lapsley et al. in [15] has brought the promise of practical deployment of effective flow and congestion control algorithms for the existing Internet infrastructure. In addition, applications of control and optimization theories such as the ones described by [3], [5], [8], [9], [22] have shed light on the general problem of flow control in computer communication networks. Although leading to rather different flow control strategies, the key promise of most of the recent results is to maximize a set of utility functions pertaining to the benefit of various network entities while potentially considering pricing issue. Another closely related literature approach to our current topic advocates a game-theoretic approach as described by [20] and [14] in which reaching a stable Nash equilibrium solution is desired.

With the above introduction, we clarify that the main focus of this research work is to develop inter-session (flow) fairness algorithms relying on the concept of max-min fairness. In this study, we pay special attention to the results of Athuraliya et al. [1], Graves et al. [11], Kelly et al. [13], Low et al. [19], Kunniyur et al. [14], Ramakrishnan et al. [21], and Sarkar et al. [23]. Our formulation of the flow control problem is best categorized under the optimization flow control techniques. It is hence aimed at maximizing a global and a per link set of utility functions defined over the complete path of unicast and multicast tree topologies. More specifically, our formulation of the flow control problem is a convex optimization problem defined over a set of piecewise linear utility functions. The main advantage of utilizing such a set of utility functions compared to the previously proposed nonlinear utility functions is simplicity. Not only appealing from the complexity standpoint, our technique can also satisfy important characteristics of the well-behaved algorithms such as guaranteed existence, boundedness, stability, and scalability. With respect to practicality, the resulting proposed algorithm can be implemented in real-time by merely taking advantage of a binary ECN marking mechanism currently under review by IETF [21].

In summary, our solution to the formulation of the flow control problem identifies maximum achievable fair rates for individual unicast and multicast sessions sharing the same underlying network infrastructure. An outline of the paper follows. In Section II, we formulate and analytically solve a pair of global and per link optimal flow control problems relying on the so-called max-min fairness metric. Our solutions to these problems are capable of addressing inter-session fairness issue in order to specify a fair assignment of the available bandwidth among a set of competing unicast or multicast flows. In Section III, we numerically validate our analytical results. Finally, Section IV includes a discussion of concluding remarks and future work.

II. FLOW CONTROL OPTIMIZATION PROBLEMS

In this section, we focus on a pair of optimal flow control problems categorized under constraint convex optimization problems with piecewise linear objective functions and their solutions. We start from a centralized global formulation of the problem aiming at guaranteeing inter-session fairness among competing unicast and multicast flows utilizing the set of links over an existing network topology. We then move on to a decentralized local formulation of the problem relying on the concept of max-min fairness and compare the two methods regarding complexity and overall fairness issues.
In the formulation of our problems, we consider the max-min fairness concept of [4] defined below.

**Definition 2.1:** A bandwidth allocation scheme among a number of competing flows is max-min fair if no flow can be allocated a higher bandwidth without reducing the allocation of another flow with an equal or a lower rate.

Intuitively, max-min fair allocation implies that for a number of competing flows over an existing network topology, each flow should be able to receive a “fair share” of the available bandwidth. If a flow cannot fully utilize its fair share due to a limitation imposed by another link, then the residual bandwidth of that flow is split fairly among other flows.

### A. The Problem of Centralized Flow Control

Taking into consideration the above notion of max-min fairness, we formulate a convex optimization problem by means of defining a per flow fairness utility with the objective of maximizing the sum of utilities over the set of links of a given network topology.

Assume \( f \) flows are sharing a set of links \( L \) over a particular network topology. Further, assume the capacity of link \( j \) where \( j \in L \) is specified by \( C_j \). Each flow \( i \) has a maximum required bandwidth denoted by \( X_i \). Depending on the characteristics of flow \( i \) the term \( X_i \) could vary from a minimum guaranteed available bandwidth for a restricted flow to the full capacity of the bottleneck link over a unicast or a multicast path for an unrestricted flow. Hence, assigning a bandwidth higher than the requested value \( X_i \) to flow \( i \) leads to capacity wastage of the set of links utilized by flow \( i \) due to the fact that flow \( i \) cannot utilize more than its maximum required bandwidth. In accordance with the latter assumption, we select the following concave utility function to represent the fairness of individual flows.

\[
U_i(x_i) = \min \left( \frac{x_i}{X_i}, 1 \right) = \begin{cases} 
\frac{x_i}{X_i} & x_i \leq X_i \\
1 & x_i > X_i 
\end{cases}
\]

(1)

Fig. 1 illustrates sample drawings of such a utility function.

![Utility](image)

**Fig. 1.** Sample drawings of the utility function \( U_i(x_i) \) for flows \( i \in \{1, 2\} \) with maximum required bandwidth \( X_i \).

Assuming an ordered set of bandwidth requirements \( X_1, X_2, \ldots, X_f \) such that \( X_1 \leq X_2 \leq \ldots \leq X_f \), our formulation of the global flow control problem is now described in the form of the following optimization problem.

\[
\max_{x_1, \ldots, x_f} \sum_{j \in L} \sum_{i=1}^f w_{ji} U_i(x_i)
\]

Subject To:

\[
\sum_{i=1}^f w_{ji} x_i \leq C_j \quad \forall j \in L
\]

\[
x_1 \leq x_2 \leq \ldots \leq x_f
\]

(2)

where \( f \) is the total number of flows over the network topology, \( C_j \) is the capacity of link \( j \), \( U_i(x_i) \) is the utility function defined in Equation (1), and \( w_{ji} \) is the weighting function defined below indicating whether link \( j \) is utilized by flow \( i \).

\[
w_{ji} = \begin{cases} 
1 & \text{if flow } i \text{ utilizes link } j \\
0 & \text{otherwise}
\end{cases}
\]

(3)

Next, we convert the problem of (2) to a standard linear programming (LP) problem as the result of replacing the second linear piece in the saturation area by a new constraint. The equivalent standard linear programming problem is expressed as

\[
\max_{x_1, \ldots, x_f, \lambda} \sum_{i=1}^f W_i \left( \frac{x_i}{X_i} \right)
\]

Subject To:

\[
\sum_{i=1}^f w_{ji} x_i \leq C_j \quad \forall j \in L
\]

\[
x_i \leq X_i \quad \forall i \in \{1, \ldots, f\}
\]

\[
x_i \leq x_{i+1} \quad \forall i \in \{1, \ldots, f-1\}
\]

(4)

where \( W_i \triangleq \sum_{j \in L} w_{ji} \) with \( i \in \{1, \ldots, f\} \). We now note that the LP problem of (4) can be solved relying on one of the few existing methods such as the LU-decomposition method or sparse Bartel-Golub method as described in [10]. Alternatively, the problem may be solved relying on the iterative approximation method of [2]. Depending on the choice of algorithm and numerical applicability, the average complexity of solving the LP problem of (4) can be in the order of \( \mathcal{O}(l + 2f)^2 \) where \( l \) is the number of links over the network topology \( L \). It is in order to mention that the above formulation of the global flow control problem is in fact implementing a priority mechanism in which the flow weights are set proportional to the number of end nodes links traversed by the flow. Despite the fact there may be other considerations for implementing flow priorities such as the number of flow end nodes, the final formulation of the problem nevertheless comes down to (4) for a different choice of the weighting functions \( \{W_{1}, \ldots, W_{f}\} \). It is also important to note that the solution to the problem formulation of (4) follows the max-min fairness property of definition of 2.1 if \( \frac{W_i}{X_i} > \ldots > \frac{W_f}{X_f} \). The selection of the weighting functions is a design consideration that can be offset by the relative importance of priority over max-min fairness.

### B. The Problem of Decentralized Flow Control

Considering the need for accessing global state information among the set of links of a given network topology as well as
the complexity of the solution to the global problem above, we reduce the global problem into a set of per link flow control optimization problems. The set of per link problems can then be solved independently and with a linear complexity for both unicast and multicast flows and without requiring to access any state information among the links of a given topology. Not requiring to access state information, however, comes in exchange for potential estimation of flow fair shares yielding to sub-optimality. The latter is due to the fact that a fair share calculated for a flow at a link may be subject to extra limitations or relaxations imposed by another link.

Assume \( f \) flows are sharing a link with capacity \( C \) and each flow \( i \) has a maximum required bandwidth \( X_i \). Relying on the definition of the convex utility function of (1) and assuming an ordered set of bandwidth requirements \( X_1, X_2, \ldots, X_f \) such that \( X_1 \leq X_2 \leq \ldots \leq X_f \), our per link formulation of the flow control problem is now described in the form of the following optimization problem.

\[
\begin{align*}
\max_{x_1, \ldots, x_f} & \quad \sum_{i=1}^{f} U_i(x_i) \\
\text{Subject To} : & \quad \sum_{i=1}^{f} x_i \leq C \\
& \quad x_1 \leq x_2 \leq \ldots \leq x_f
\end{align*}
\tag{5}
\]

where \( f \) is the number of competing flows over a link, \( C \) is the capacity of the link, and \( U_i(x_i) \) is the utility function of flow \( i \) as defined in Equation (1). We observe that solving per link optimization problem of (5) does not require accessing any state information. The problem can be solved utilizing a similar approach as the one utilized in the previous subsection and noting the fact that \( \frac{w_i}{X_i} > \cdots > \frac{w_f}{X_f} \) holds. Rather than relying on the approach of the previous subsection, we select water-filling approach in order to find the unique solution of the problem with a lower complexity while satisfying Definition 2.1. We express the water-filling solution to the constraint optimization problem of (5) as follows.

**Case 1:** If \( C \geq \sum_{j=1}^{f} X_j \)

\[
x_i = X_i, \quad 1 \leq i \leq f
\tag{6}
\]

**Case 2:** If \( C < \sum_{j=1}^{f} X_j \)

\[
x_i = \begin{cases} 
\frac{C - \sum_{j<h} x_j}{f - h} X_i, & 1 \leq i \leq h \\
X_{h+1}, & h+1 \leq i \leq f
\end{cases}
\tag{7}
\]

where \( x_i \) is the bandwidth assigned to the \( i \)-th flow and \( h \) satisfies the following condition

\[
X_h \leq \frac{C - \sum_{j<h} x_j}{f - h} \leq X_{h+1}
\tag{8}
\]

for \( X_0 \triangleq 0 \). In [24], we prove that the water-filling solution of Equation (7) is, in fact, the optimal solution to the constraint optimization problem of (5).

We observe that the water-filling approach of Equation (7) starts by dividing the bandwidth equally among all of the \( f \) flows until the first flow reaches its maximum required bandwidth \( X_{h+1} \), then it fixes the assigned bandwidth for the first flow to \( X_1 \) and divides the remaining bandwidth among the remaining flows equally, and so on. Consequently, the flows that have reached their saturation regions receive their maximum requested bandwidth while the other flows receive equal shares of the remaining bandwidth guaranteed not to be less than the assigned shares of flows in their saturation regions. The method hence satisfies Definition 2.1 of max-min fairness. To clarify the above discussion consider the following example.

**Example 2.1** Assume three flows are sharing a link with capacity \( C = 3.5 \text{ Mbps} \). In addition, assume that the maximum bandwidth requested by each of the three flows is \( X_1 = 0.67 \text{ Mbps}, X_2 = 1 \text{ Mbps}, X_3 = 2 \text{ Mbps} \).

According to the water-filling approach, first we have to find the value of \( h \). From (8), we observe that the inequality holds for \( h = 2 \). Consequently, Equation (7) introduces the fair bandwidth assignment of the flows as

\[
x_i = \begin{cases} 
0.67 \text{ Mbps}, & i = 1 \\
1 \text{ Mbps}, & i = 2 \\
1.83 \text{ Mbps}, & i = 3
\end{cases}
\]

which is max-min fair. \( \square \)

Utilizing the water-filling approach at an intermediate node with capacity \( C \) and accommodating \( f \) competing flows, we now formalize our flow control algorithm introducing a linear complexity of \( O(f) \).

**Flow Control Optimization Algorithm**

- Initialize \( S = 0 \) and \( x_f = 0 \).
- for \( (p = 1 \text{ to } f - 1) \) {
  - \( S += X_p \)
  - if \( \frac{S}{X_p} > X_{p+1} \), then \( x_p = X_p \)
  - else {
    - \( f \text{ or } (q = (p+1) \text{ to } f) \) {
      - \( x_q = \frac{S}{X_p} \)
    }
    break
  }
  break
- if \( (x_f = 0) \), then \( x_f = \min(X_f, C - S) \)

We close this section by discussing some of the protocol implementation issues. First, it is possible to envision a hybrid flow control optimization problem that can be solved over local zones. The idea behind proposing such a scenario is to address the tradeoff between accuracy and the practicality of the solution. In such a scenario, the optimization problem of Section II.A can be solved over the topologies of local zones in which exchanging state information is not overhead prohibitive. Applying decentralized approach of this subsection to local zones can then identify the minimum fair share of each flow.
Next aside from the fact that our decentralized approach can be independently utilized for network links accommodating both unicast and multicast sessions, there are additional implementation considerations that need to be addressed in the case of multicast networks. These considerations revolve around addressing the problem of feedback implosion in the process of collecting the bottleneck information of individual receivers. We address the feedback implosion issue relying on the proposed literature aggregation methods of [16], [18] and suppression method of [6]. In [24], we discuss the details of feedback aggregation and suppression. Further, we point out that our decentralized approach calls for detecting a per flow bottleneck introducing minimum available fair share to a specific flow. Relying on the binary ECN marking technique of [21] and the mark-based estimation technique of [11], we also propose a protocol implementation of our algorithm in [24] to detect and convey bottleneck link information to the end nodes of a session. We note that unlike the iterative estimation algorithm of [11] introducing a hyperlinear complexity, our decentralized algorithm proposes a precise analytical method of calculating fair shares with linear complexity. Although in our experiments the runtime of our decentralized algorithm is typically a fraction of that of [11], considering the major difference between the utility functions we do not see a good justification to directly compare the results together.

III. NUMERICAL ANALYSIS

In this section, we provide a numerical example to further illustrate centralized and decentralized algorithms of Section II. With the assumption that all of the bandwidth units of the current section are the same, we suppress the units. Additionally for the example of this section, we denote $x_i$ as the rate of the $i$-th unicast session and $x_{ij}$ as the rate of the $j$-th virtual session of the $i$-th multicast session. Accordingly, we assume an unrestricted unicast or multicast session $i$ is requesting a bandwidth of $X_i$ equal to the capacity of the bottleneck link over its path to a source. In the case of multicast session $i$, we assume a virtual session $j$ is requesting a bandwidth of $X_{ij}$ also equal to the capacity of the bottleneck link over its path to a source. Taking into consideration that the resulting assigned rates of the virtual sessions belonging to the same multicast session are the same, the value of $X_i$ for a multicast session $i$ is related to the values $X_{ij}$ of its virtual sessions as $X_i = \min_j X_{ij}$.

Example 5.1 As an example, we borrow the sample network topology of [11] as illustrated in Fig. 2. The sample topology consists of 6 unrestricted multicast and 5 unrestricted unicast sessions distributed over a total of 19 links. We note that the 6 multicast sessions consist of a total of 14 virtual sessions. Table I provides specifications of the sample network as well as a comparison between the results of our centralized and decentralized algorithms. The first four columns of Table I respectively show the virtual session, its underlying path, its requested bandwidth, and the resulting requested bandwidth of its flow according to $X_i = \min_j X_{ij}$. While the values of the third and fourth columns are the same in the case of unicast flows, they may differ in the case of multicast sessions.

![Fig. 2. An illustration of the first sample network topology.](image)

This is due to the fact that the third column value indicates the capacity of the bottleneck link over the path of a specific virtual session while the fourth column value is the capacity of the bottleneck link over all of the virtual sessions of the same multicast session. The middle three columns of Table I respectively show the link number, the link capacity, and the set of corresponding calculated session rates according to our decentralized algorithm. Note that the values of the seventh column are sorted in order, corresponding to the value of $X_i$ for their related flows. In Table I, we also compare calculated fair shares of individual flows resulting from utilizing the centralized algorithm of Section II.A with those from the decentralized algorithm of Section II.B. In order to reward a multicast session, the flows of the centralized algorithm have been calculated by applying a priority mechanism in which the flow weights are set proportional to the the number of end nodes associated with the flow. Alternately the weights can be set proportional to the ratio of the number of end nodes to the number of links traversed by a flow in order to discourage heavy utilization of the network resources.

We justify the differences between the results of the two methods by considering the fact that the choice of the weighting functions in the centralized method implements a flow priority mechanism while it enforces max-min fairness in the decentralized method. We also argue that from a practical standpoint, the use of the decentralized algorithm is most probably the preferred choice when dealing with large size networks.

IV. CONCLUSION

In this paper, we studied the solution to the general problem of flow control for hybrid unicast and multicast IP networks. We aimed at providing centralized and decentralized optimal solutions to address inter-session fairness issue among competing unicast and multicast flows. Relying on the standard linear programming schemes and water-filling scheme respectively, our solutions to centralized and decentralized formulations of the flow control problem analytically determined
maximum allowable rates maximizing a max-min fairness metric. We pointed out that our low complexity decentralized algorithm could be implemented with minimal ECN marking support from intermediate network nodes. Further, we noted that our proposed decentralized technique did not require storing any state information in intermediate network nodes. Finally, we compared the performance of our centralized and decentralized solutions and illustrated their applicability in a sample network topology. Our future work concentrates on the expansion of our flow control results into a general combined framework for congestion and flow control. Relying on the implementation of our flow control algorithm, we are developing a reactive receiver-oriented congestion control scheme that can be applied to various multicasting applications.

REFERENCES


TABLE I

<table>
<thead>
<tr>
<th>Virtual</th>
<th>Path</th>
<th>X_{ij}</th>
<th>X_i</th>
<th>Link</th>
<th>Capacity</th>
<th>Per Link Session Rates</th>
<th>Session Rates</th>
<th>Centralized MFS</th>
<th>Decentralized MFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x_{01}</td>
<td>1, 3, 8</td>
<td>4</td>
<td>4</td>
<td>l_i</td>
<td>14</td>
<td>2, 2, 2, 2, 2, 2</td>
<td>x_0</td>
<td>1.25</td>
<td>2</td>
</tr>
<tr>
<td>x_{02}</td>
<td>1, 3, 9, 15</td>
<td>4</td>
<td>4</td>
<td>l_2</td>
<td>9</td>
<td>1.5, 2.5, 2.5, 2.5</td>
<td>x_1</td>
<td>4.75</td>
<td>2</td>
</tr>
<tr>
<td>x_1</td>
<td>1, 4, 10, 15</td>
<td>6</td>
<td>6</td>
<td>l_3</td>
<td>6</td>
<td>2, 2, 2</td>
<td>x_2</td>
<td>1.25</td>
<td>1.25</td>
</tr>
<tr>
<td>x_{21}</td>
<td>1, 5, 11, 16</td>
<td>5</td>
<td>5</td>
<td>l_4</td>
<td>6</td>
<td>3, 3, 3</td>
<td>x_3</td>
<td>1.25</td>
<td>1.25</td>
</tr>
<tr>
<td>x_{22}</td>
<td>1, 5, 12, 18</td>
<td>5</td>
<td>5</td>
<td>l_5</td>
<td>5</td>
<td>1.25, 1.25, 1.25, 1.25</td>
<td>x_4</td>
<td>1.45</td>
<td>1.5</td>
</tr>
<tr>
<td>x_3</td>
<td>1, 5, 11, 17</td>
<td>4</td>
<td>4</td>
<td>l_6</td>
<td>6</td>
<td>2.2, 2, 2, 2, 2, 2, 2</td>
<td>x_5</td>
<td>1.25</td>
<td>1.5</td>
</tr>
<tr>
<td>x_{41}</td>
<td>1, 6, 12, 18</td>
<td>5</td>
<td>5</td>
<td>l_7</td>
<td>6</td>
<td>1.5, 1.5, 1.5, 1.5, 1.5</td>
<td>x_6</td>
<td>1.25</td>
<td>2</td>
</tr>
<tr>
<td>x_{42}</td>
<td>1, 6, 14, 19</td>
<td>5</td>
<td>5</td>
<td>l_8</td>
<td>4</td>
<td>1.5, 1.5, 1.5, 1.5, 1.5</td>
<td>x_7</td>
<td>1.25</td>
<td>1.25</td>
</tr>
<tr>
<td>x_5</td>
<td>1, 7, 13, 18</td>
<td>1.5</td>
<td>1.5</td>
<td>l_9</td>
<td>4</td>
<td>2, 2, 2, 2, 2, 2, 2</td>
<td>x_8</td>
<td>1.649</td>
<td>1.5</td>
</tr>
<tr>
<td>x_6</td>
<td>1, 3, 9, 15</td>
<td>4</td>
<td>4</td>
<td>l_{10}</td>
<td>6</td>
<td>3, 3, 3, 3, 3, 3</td>
<td>x_9</td>
<td>1.649</td>
<td>1.5</td>
</tr>
<tr>
<td>x_{71}</td>
<td>1, 3, 8</td>
<td>4</td>
<td>4</td>
<td>l_{11}</td>
<td>8</td>
<td>1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6</td>
<td>x_A</td>
<td>1.25</td>
<td>1.25</td>
</tr>
<tr>
<td>x_{72}</td>
<td>1, 9, 11, 16</td>
<td>5</td>
<td>4</td>
<td>l_{12}</td>
<td>7</td>
<td>2.3333, 2.3333, 2.3333</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x_{73}</td>
<td>1, 5, 11, 17</td>
<td>4</td>
<td>4</td>
<td>l_{13}</td>
<td>1.5</td>
<td>1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x_{81}</td>
<td>2, 6, 12, 18</td>
<td>6</td>
<td>5</td>
<td>l_{14}</td>
<td>5</td>
<td>1.6667, 1.6667, 1.6667</td>
<td>x_{x}</td>
<td>1.649</td>
<td>1.5</td>
</tr>
<tr>
<td>x_{82}</td>
<td>2, 6, 11, 16</td>
<td>5</td>
<td>5</td>
<td>l_{15}</td>
<td>10</td>
<td>2.5, 2.5, 2.5, 2.5, 2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x_{83}</td>
<td>2, 7, 14, 19</td>
<td>5</td>
<td>5</td>
<td>l_{16}</td>
<td>5</td>
<td>1.6667, 1.6667, 1.6667</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x_9</td>
<td>2, 7, 14, 19</td>
<td>5</td>
<td>5</td>
<td>l_{17}</td>
<td>4</td>
<td>1.3333, 1.3333, 1.3333</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x_{91}</td>
<td>1, 4, 10, 15</td>
<td>6</td>
<td>4</td>
<td>l_{18}</td>
<td>9</td>
<td>1.5, 2.5, 2.5, 2.5</td>
<td>x_{x}</td>
<td>1.25</td>
<td>1.25</td>
</tr>
<tr>
<td>x_{92}</td>
<td>1, 5, 11, 17</td>
<td>4</td>
<td>4</td>
<td>l_{19}</td>
<td>5</td>
<td>1.6667, 1.6667, 1.6667</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>