Generalized graph coloring for spectrum efficient frequency assignment in cellular network

Presented by
Dr. Sasthi C. Ghosh
Assistant Professor
Advanced Computing & Microelectronics Unit
Indian Statistical Institute
Kolkata, India
Channel Assignment Problem (CAP)

- Area is divided into a number of cells (typically hexagonal in shape)

- Available bandwidth (very limited) is divided into a number of nonoverlapping frequency bands of equal width termed as channels (0, 1, 2, …)

- Each cell is required to be assigned a set of frequency channels to provide services to the calls in that cell

- Frequency separation constraints: frequencies assigned to nearby cells must be separated sufficiently to avoid channel interference

- CAP: assigning frequency channels to the calls satisfying the frequency separation constraints and using bandwidth as small as possible.

- In its most general form CAP is NP-Complete [Hale, 1980]
Channel Assignment Problem (CAP)

- **Cellular graph:**
  - Each cell of the cellular network is represented by a node.
  - Two nodes have an edge between them if the corresponding cells are adjacent to each other.
  - Adjacent: cell boundaries share a common edge.
Channel Assignment Problem (CAP)

- Set of cells: \( X = \{0, 1, \ldots, n-1\} \)
- Demand vector \( W = (w_i) \)
  - \( w_i \): demand from cell \( i \) (\( 0 \leq i \leq n-1 \))
- Frequency separation matrix \( C = (c_{ij}) \)
  - \( c_{ij} \): minimum frequency separation required between a call in cell \( i \) and a call in cell \( j \) (\( 0 \leq i, j \leq n-1 \))
  - \( f_{ij} \): frequency assigned to call \( j \) in cell \( i \) (\( 0 \leq i \leq n-1, 0 \leq j \leq w_i - 1 \))

**Frequency separation constraints:**
- \(|f_{ik} - f_{jl}| \geq c_{ij} \) for all \( i, j, k, l \) (except when both \( i = j \) & \( k = l \))

The triplet \( P = (X, W, C) \) characterizes a CAP
Minimum span channel assignment

Given \((X, W, C)\), find \(F = (f_{ij})\) such that

- Channel demands of all the cells are satisfied
- Frequency separation constraints are satisfied
- The span \(= \max_{i,j}(f_{ij})\) is minimized
Channel Assignment Problem (CAP)

Fixed bandwidth channel assignment

Given $\mathcal{X}, \mathcal{W}, \mathcal{C}$ and $\lambda$, find $\mathcal{F} = (f_{ij})$ such that

- $f_{ij} \in \{0, 1, \ldots, \lambda\}$
- Frequency separation constraints are satisfied
- Maximum demands are satisfied or minimizes blocked calls
The benchmark instances

- The eight benchmark instances widely used in the literature
- The cellular layout of 21 cells system
- Two non-homogeneous demand vectors $D_1$ and $D_2$
- Frequency separation constraints: $s_0$, $s_1$ and $s_2$ are minimum separation required between calls at cells distance 0, 1 and 2 apart

$Demanded vector D_1$

Label $[\alpha]$ implies the channel demand of that node
The benchmark instances

Demand vector $D_2$

The specification of eight benchmark problems

<table>
<thead>
<tr>
<th>Problem number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency separation constraints</td>
<td>$s_0$</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>$s_1$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$s_2$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Demand vector</td>
<td>$D_1$</td>
<td>$D_1$</td>
<td>$D_1$</td>
<td>$D_1$</td>
<td>$D_2$</td>
<td>$D_2$</td>
<td>$D_2$</td>
<td>$D_2$</td>
</tr>
</tbody>
</table>
Map to the Graph-Coloring Problem

The CAP Graph

- Call-j of cell-i is mapped to a node \((i, j)\).
- Nodes \((i, j)\) and \((k, l)\) are connected by an edge with weight \(c_{ik}\) if \(c_{ik} > 0\).
- Number of nodes in CAP graph: \(\sum_i w_i = \text{total demands}\).

**Example**: 3 cells \(\{0, 1, 2\}\) with demands 1, 2, and 2 respectively.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

Frequency separation matrix
Frequency Exhaustive

Consider an ordering of the vertices, say $v_0, v_1, \ldots, v_{n-1}$.
Color the vertices one by one as per the ordering.
While coloring $v_i$, choose the minimum color that satisfies the constraints with already allocated vertices $v_0, v_1, \ldots, v_{i-1}$.

Order: $u, x, v$
Colors: 0, 3, 5

Order: $u, v, x$
Colors: 0, 3, 5

Bandwidth depends on the order of nodes traversed.
Impractical to find the best ordering (**m! orderings**)

Complexity: $O(n^2 k)$, $k=$ maximum edge weight $\times$ maximum degree
Forced Assignment (IEE NCA 2014)

- Consider an ordering of the vertices, say \( v_0, v_1, ..., v_{n-1} \).
- Color the vertices as per the ordering using FE.
- If color of \( v_i \) \((i > 2)\) exceeds the maximum color of \( v_0, v_1, ..., v_{i-1} \) do the following:
  - For all \( j = 0, 1, ..., i-1 \) do
    - Save the colors of \( v_i \) and \( v_j \) and then free them.
    - Color \( v_i \) first and then \( v_j \) using FE.
  - If the so far obtained span is reduced keep this new colors and exit the loop
  - Else return back the previous colors of \( v_i \) and \( v_j \)

**Complexity:** \( n \times \text{complexity of FE} = O(n^3 k) \)
Forced Assignment

Example:
Order: \( u, x, v \)
Colors: 0, 3, 5 \( \text{span}=5 \)
Free the colors of \( u, v \) and consider \( v \) first and then \( u \)
(color of \( x \) remain unchanged)
New colors: 5, 3, 0 \( \text{span}=5 \)
Free the colors of \( x, v \) and consider \( v \) first and then \( x \)
(color of \( u \) remain unchanged)
New colors: 0, 4, 2 \( \text{span}=4 \)
Triangle Compression

- If \((f(u), f(v), f(x))\) \([f(u) \leq f(x) \leq f(v)]\) is a conflict free assignment of triangle \(T(u, v, x)\) then \((f'(u), f'(v), f'(x))\) is also a conflict free assignment of \(T\).
  - \(f'(u) = f(u)\),
  - \(f'(v) = f(u) + w(u, v)\) and
  - \(f'(x) = \max(f(x), f'(v) + w(v, x))\).

- Here \(f'(u) \leq f'(v) \leq f'(x)\)
- If \(f'(x) < f(v)\) we say triangle compression is successful.
- Here \(u\) is called \(\mu\) vertex and \((f'(x) - f(x))\) as extra value

vertices : \(u, v, x\)
Color \(f\) : 0, 5, 3
Color \(f'\) : 0, 0+2, \(\max(3, 2+2)\)
\(\mu\) vertex = \(u\), extra=4-3=1
Consider a given conflict-free assignment of the graph.

Let v be the only vertex with the maximum color.

Consider all possible triangles to which v is a member and look for a successful triangle compression among all such triangles.

If such a triangle T(u,v,x) is found, save the $\mu$ vertex and the value of extra.

If $\text{extra} < (\text{maximum} \ - \ \text{second maximum})$ do the following:

Partition $G \setminus \{v\}$ into H and L where H contains all the vertices which has color $> f(\mu)$ and L contains the remaining vertices.

Increase the color of each vertex in $H \setminus \{x\}$ by extra value and then check whether this assignment is conflict free (can be checked in $O(n)$ time as conflict could be only between $\{v\}$ and $G \setminus \{v\}$).

If conflict free, accept this assignment and repeat the process as long as reduction is possible.

**Complexity: $O(n^3 k)$**
Assignment Algorithm (IEEE TMC 2013)

- Partition the given CAP into a sequence of subproblems, each corresponds to a CAP with homogeneous demands.
- The CAP with nonhomogeneous demands is solved using the solution of those subproblems with homogeneous demands.
- The CAP with homogeneous demands can be solved using single frequency assignment.
- We present the proposed algorithm stepwise with an example of (the most difficult) Philadelphia benchmark problem 6.
Assignment Algorithm

Phase I

- Consider the subgraph that has only the highest demand nodes (only node 11 has the highest demand 45)

- Next, consider a single frequency assignment for the set of highest demand node. Since only one node, we assign frequency 0 (origin) to node 11.

\[ s_0 = 5 \]
\[ s_1 = 2 \]
\[ s_2 = 1 \]
• In a **multiple frequency assignment**, the next frequency that can be assigned to node 11 is 5; So the value of **increment** = 5.

• Consider assignment of the whole network with the frequencies 0, 1, 2, 3 and 4 [origin, origin+Increment) for a single frequency assignment.

• $V$ be the set of nodes covered with frequencies 0 - 4 and $V'$ be the left out nodes

\[
V = \{11, 10, 9, 8, 7, 6, 13, 12, 18, 19, 20\} \quad \text{and} \quad V' = \{0, 1, 2, 3, 4, 5, 14, 15, 16, 17\} \]
Assignment Algorithm

- Satisfy $\omega$ demands at each node in $V$ where $\omega = \gamma_{\text{max}} - \gamma'_{\text{max}}$. Here, the highest demands in $V$ and $V'$ are $\gamma_{\text{max}}$ and $\gamma'_{\text{max}}$.

- Here $\gamma_{\text{max}} = 45$, $\gamma'_{\text{max}} = 30$; Hence, $\omega = 15$

- A multiple homogeneous assignment at each node in set $V$
- $f_v + j \times \text{increment, } j = 0, 1, \ldots, \omega - 1$
- Frequency assigned: 0 - 74
Assignment Algorithm

- Residual demand vector: subtract $\omega$ from each vertex in $V$
- The highest demand node in $V'$ now moves to the set of the highest demand nodes of the entire network so that it will be considered by the next phase
- The residual demand vector becomes the demand vector for the next phase

\[ \begin{array}{cccccccccccccccc}
X & 11 & 10 & 9 & 17 & 6 & 8 & 13 & 14 & 7 & 5 & 20 & 19 & 18 & 12 & 16 & 15 & 4 & 3 & 2 & 1 & 0 \\
X' & 17 & 11 & 10 & 9 & 14 & 5 & 16 & 6 & 8 & 15 & 13 & 4 & 7 & 20 & 3 & 2 & 18 & 19 & 12 & 1 & 0 \\
\end{array} \]

Phase II
- A single frequency assignment of the subgraph with highest demand nodes 17 and 11 (starts with frequency 75, the new origin)
- With the same logic, we consider the assignment of the whole network with the frequencies 75, 76, 77, 78 and 79
• $V = \{1,5,6,8,10,11,14,16,17\}$ and $V' = \{0,2,3,4,7,9,12,13,15,18,19,20\}$

• Here $\gamma_{\text{max}} = 30$, $\gamma'_{\text{max}} = 25$; Hence, $\omega = 5$

• Obtained multiple homogeneous assignment at each node in $V$ (Frequency : 75 -99)

• Compute the residual demand vector continue for the next phase

• The process terminates when residual demands becomes a vector of zeros
Assignment Algorithm

- **Convergence Behavior:**
  Must terminate after $n$ phases of assignments, where $n$ is the number of nodes in the network.

- **Time complexity:** $O(I^{\text{max}} \times n^3 + \omega^{\text{max}} \times n^2)$
  where $I$ = maximum increment and $\omega^{\text{max}}$ is maximum demand

  - $I = \max(s_0, 3s_1 + 6s_2)$ for hexagonal cellular network

- **Worst-Case Deviation from optimality:**
  - $10/7$ when $s_0 \geq s_1 + 5s_2$ & $s_2 \leq s_1 \leq 2s_2$
  - $10/7$ when $s_0 \geq 2s_1 + 3s_2$ & $s_1 \geq 2s_2$
  - $5/3$ otherwise

- When run on benchmark problems, actual execution time is, however, always within 5 ms on an HPxw8400 workstation with optimality gap at most 6%.
Conclusion

- We have discussed some efficient approach to solve the channel assignment problem.
- Computation time can be greatly reduced.
- Proposed techniques are most suitable for real life situation where fast channel assignment is demanded tolerating marginal deviation from the optimality.
- Proposed techniques can be used as a graph coloring techniques as well.