Security Level:

Interval Graph Coloring related to Optical Network Planning

Math modeling and Algorithm Optimization Competence Centre,

Moscow Research Centre of Huawei

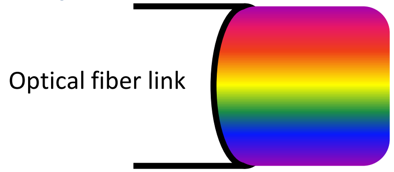
www.huawei.com

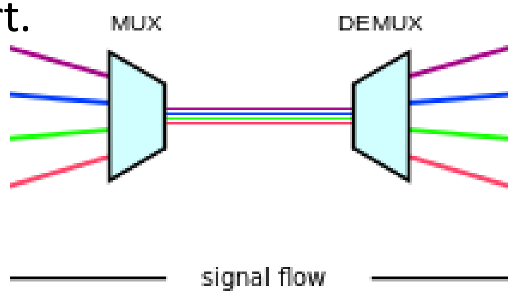
Author/ Email: Dmitri Shmelkin/Shmelkin.Dmitri@huawei.com

HUAWEI TECHNOLOGIES CO., LTD.

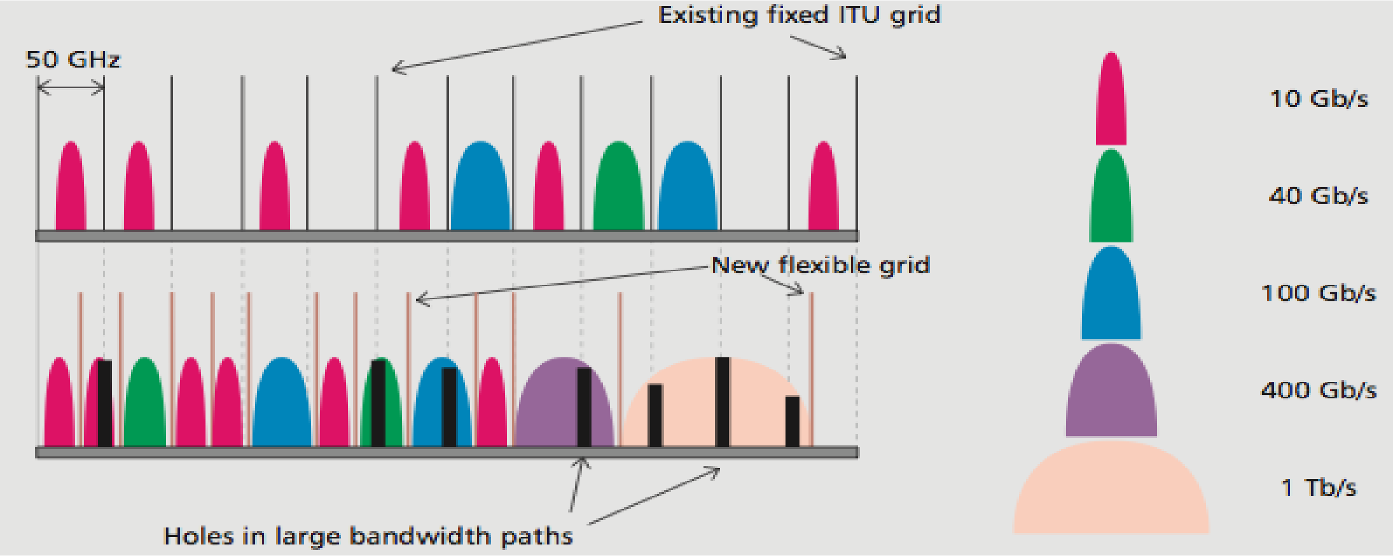
WDM: Wavelength Division Multiplexing

WDM multiplexes a number of optical carrier signals onto a single optical fiber by using different wavelengths (i.e., colors) to increase capacity.
WDM uses a multiplexer (MUX) to combine several signals together, and a de-multiplexer (DEMUX) at the receiver to split them apart.





Total frequency bandwidth is then divided into slices.



Optimization Problem

Input:

- Network $H=(V_H,E_H)$
- Commodities $D=\{d_i,i=1,..K\}$, $d_i = \{s_i, t_i\}$, $s_i, t_i \in V_H$ and a single path P_i for each commodity
- Resource of each edge is the spectrum interval, say, $[0,W)\subseteq \mathcal{N}$
- Spectrum bandwidth needed for each commodity $w:D\text{->}\mathcal{N}$

Decision variables:

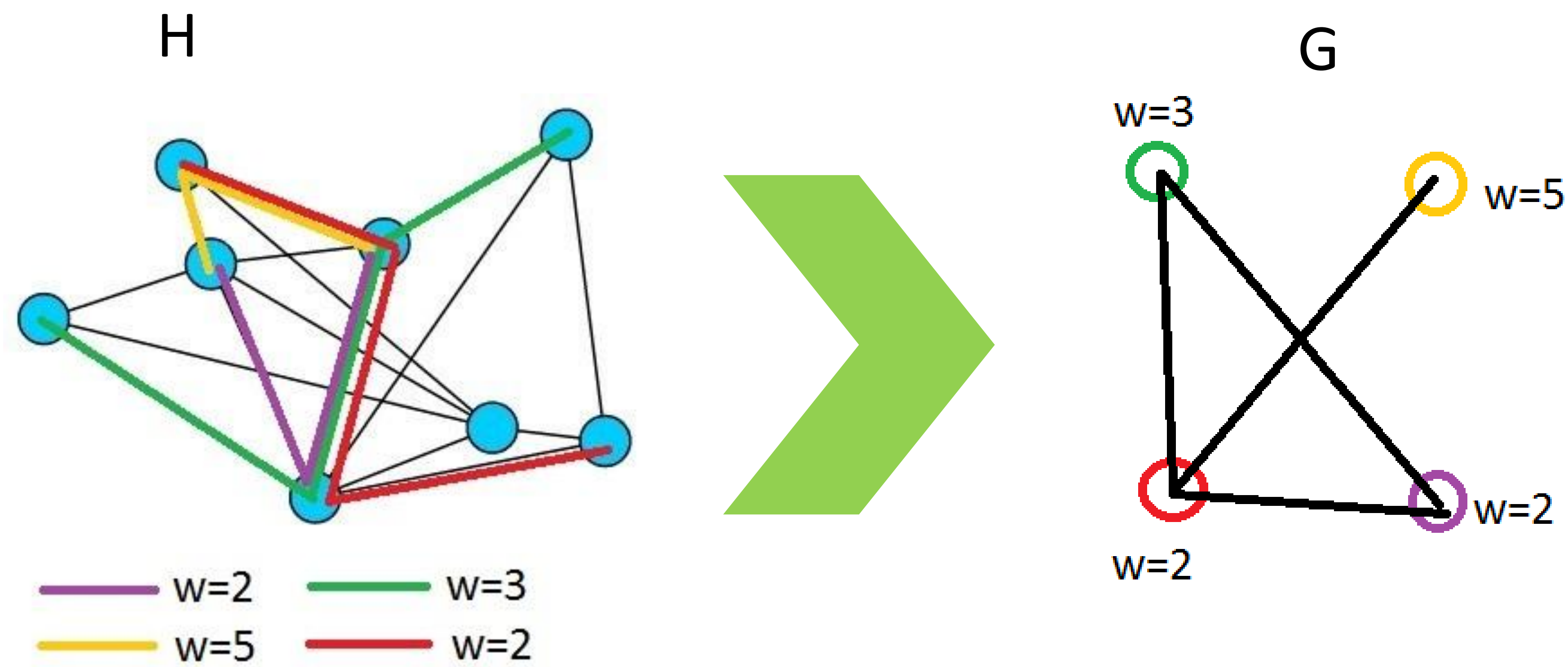
$\forall d \in D$, a spectrum interval $[l_d,r_d) \subseteq [0,W)$, such that $r_d - l_d = w(d)$ and $[l_d,r_d) \cap [l_e,r_e) = \varnothing$ *if* $P_d \cap P_e \neq \varnothing$ (*intervals are disjoint)*

Possible Objective Functions:

(A): Fix W and find a feasible subset $F \subseteq D$ with maximal $|F|$

(B): Find the minimal W such that a feasible solution for the whole D exists

Interval Vertex Coloring Approach



commodities >> nodes, paths intersection >> edges, bandwidth >> node’s weight

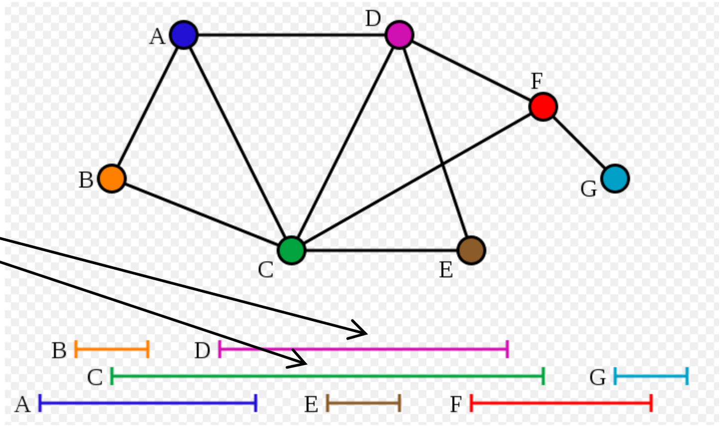
Let $G=(V_G,E_G)$ be the resulted graph.

When Network is just a Chain: Coloring of the Interval Graph

If a *H* is a **chain** • — • — • — • … — • ,

then all the paths are subchains.

Assume: all commodities have the same bandwidth, get usual coloring of **Interval Graph,** or **Interval Scheduling**



First Fit coloring yields the optimum for (B)

1. Sort intervals by their start in increasing order $d_1, \ldots, d_K$.
2. *W := 0*
3. For *j=1 to K* do
   1. If d_j spectrum can be allocated into *[0,W):*
      1. allocate it into *[0,W)*
   2. Else:
      1. assign d_j spectrum as *[W,W+1)*
      2. *W=W+1*

O(n ln(n)) working time

Interval Coloring of the Interval Graph

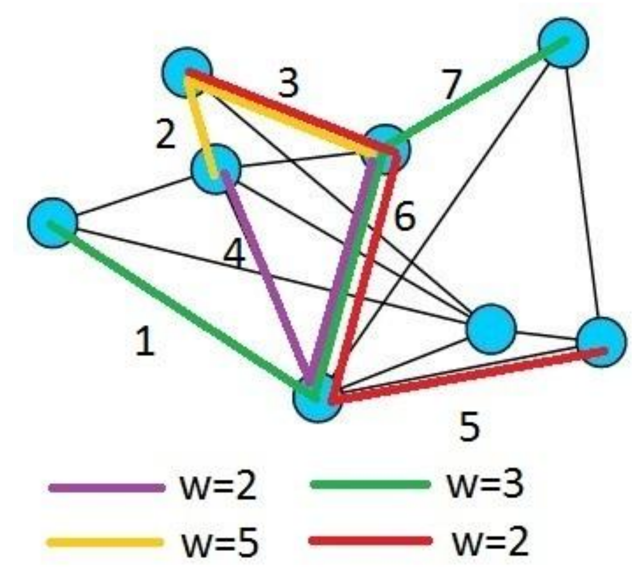
- *G* is a chain *[1,..,N],* paths are subschains, $P_k=[l_k,r_k],k=1,\dots,K$.
- In general bandwidths $w(D_k)$ are different.
- Another name: *Dynamic Storage Allocation.*
- Application area: *Compiler Memory Allocation.*
- The problem is NP hard (Stockmeyer, 1976).

Several approximate optimum for (B) problem

- Let *Load* be $L = \max_{n\in[1,N]} \sum_{k\in[1,..,K]:l_k\le n\le r_k} w(d_k)$
- Let W_{MIN} be the minimal bandwidth needed for a feasible solution.
- Obviously holds: $\boldsymbol{L \le W_{MIN}}$.
- Gergov (1999) provided a $O(n\ ln(n))$ algorithm that yields $\boldsymbol{W_{Gergov} \le 3L}$.
- Hence, $W_{MIN} \le 3L$ holds. Buchsbaum et. al (AT&T, 2003) proved the bound

$$W_{MIN} \le L\left(1+O\left(\max_k w(d_k)/L\right)^{1/7}\right)$$

General Case

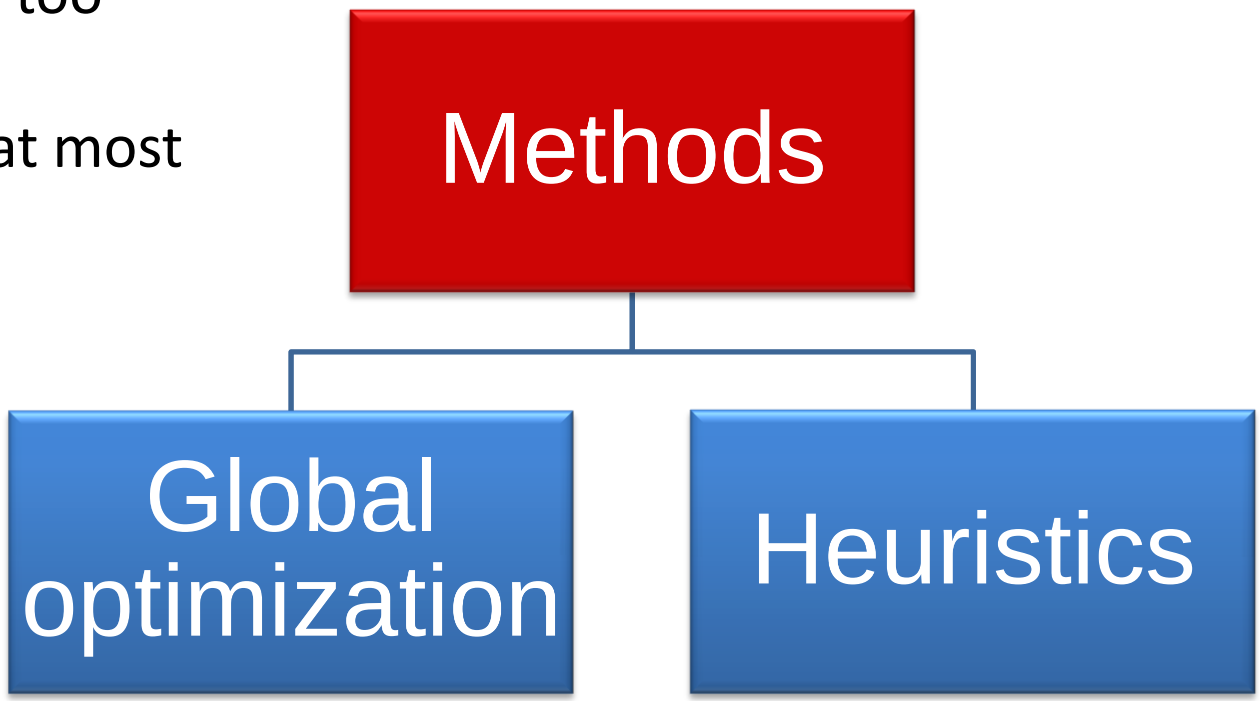


Difference compared with chain:

- “Bricks” are not connected anymore
- Commodities have different bandwidths

Coloring methods

Global optimization is too slow for our problem dimension. We need at most $O(max\{|V_G|\}^2)$ time complexity.



Ordering vertices as an approach

- The optimal solution of Interval Vertex Coloring (both for objectives (A) and (B)) can be obtained just by First Fit greedy coloring in some optimal order of the vertices.
- We reformulate the problem as Finding the optimal vertex order such that some greedy coloring (not necessarily **First** Fit) of the vertices in this order yields a good solution.
- As a matter of fact, existing methods look for such an order, e.g. Gergov's solution for interval coloring of interval graph or *Jolt Ramming*, introduced below.

Heuristic coloring approaches

The following approaches are useful in heuristic coloring:

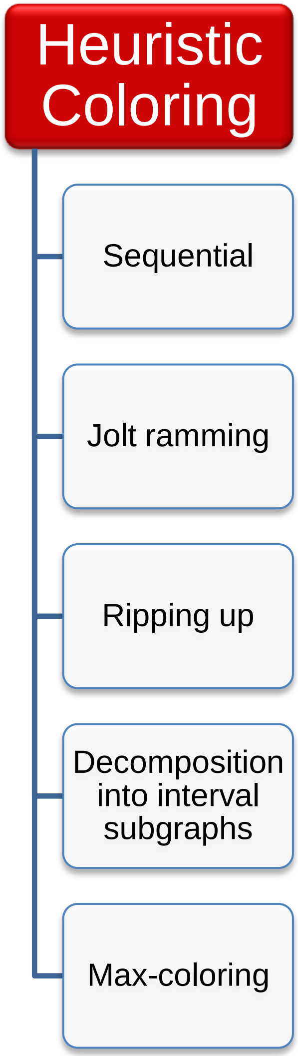
•**Sequential.** Greedy coloring of vertices in *some* order.

•**Jolt ramming.** Loop of 2 sequential colorings. The first one optimizes (A), the second one optimizes (B).

•**Ripping up.** Suppose some nodes are colored and there is no feasible color for a node *v.* Let *U* be a small subset of *v*'s colored neighbors, such that if we uncolor *U,* then there arises a feasible color for *v.* Ripping *U* up is a recursive method based on the procedure above.

•**Decomposition into interval subgraphs.** Divide V_G into subsets V_1, V_2,… such that the corresponding subgraphs are interval and reduce the complexity.

•**Max-coloring.** Some reduction to the usual vertex coloring problem (see the next slide).



Max-coloring approach

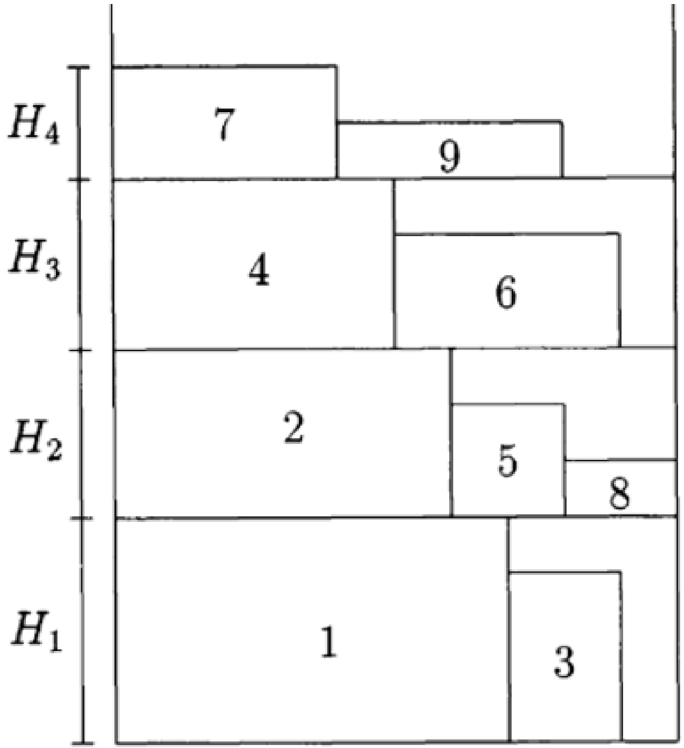
An interesting model reduces Interval coloring to a specific **usual** coloring problem:
Take a coloring $c: V\text{-}> \{0,\dots,M\text{-}1\}$ such that no edge has endpoints of the same color.
Let $v_i \in V(i)=c^{-1}(i)$ be the node with maximal weight: $v_i=\mathrm{argmax}_{v\in V(i)}(w(v)),\ i=0,..,M\text{-}1$
Let $W_i = \sum_{j=1,\dots,i-1} w(v_j)$.

Then the solution $Int(v)=[W_i,W_i+w(v))$ is feasible.

So an optimization problem arises, to find a coloring with an **arbitrary** number M of colors and the minimal possible $\boldsymbol{T=\sum_{i=0,\dots,M-1} w(v_i)}$.

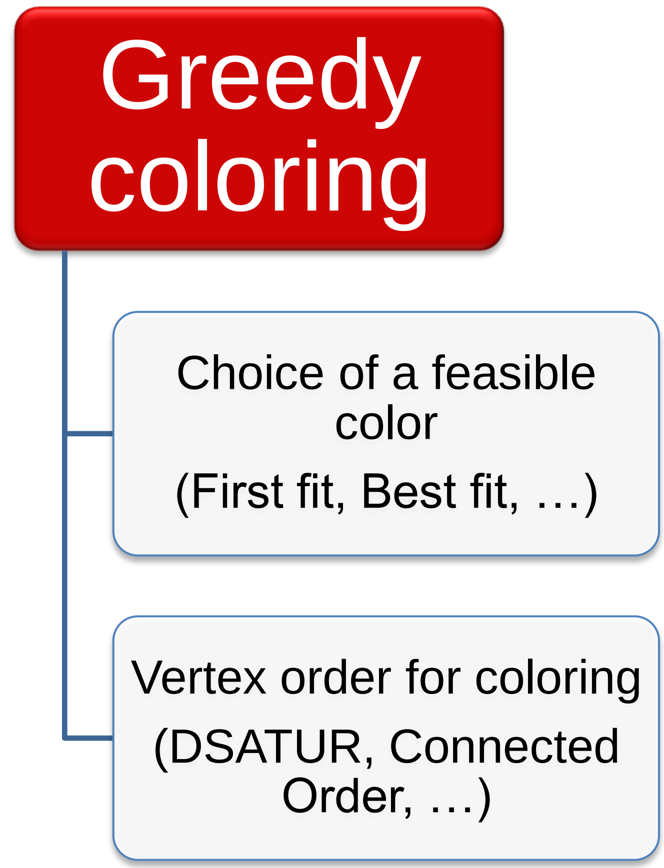
This is a **new** specific optimization of **classic** coloring.

The minimal M will not necessarily yield the minimum of T.



Greedy coloring

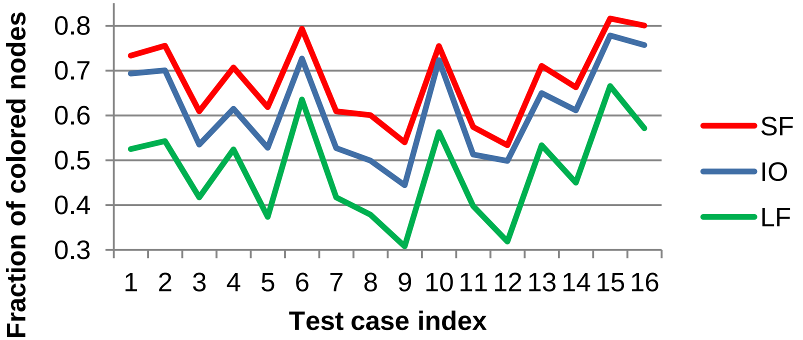
- For a vertex *v*, we can use different methods to select one color out of the list of feasible ones:
    - *First fit.* Select the lowest possible interval
    - *Best fit.* Select the narrowest possible interval within the feasible set, of size not less than *w(v)*
    - Minimize some *fragmentation* function?
- The order of coloring matters. We use the options:
    - *DSATUR.* Nodes are ordered by the number of different colors in the neighborhood of *v.*
    - *"Connected" order.* Preserve connectivity of the colored subgraph at each step.



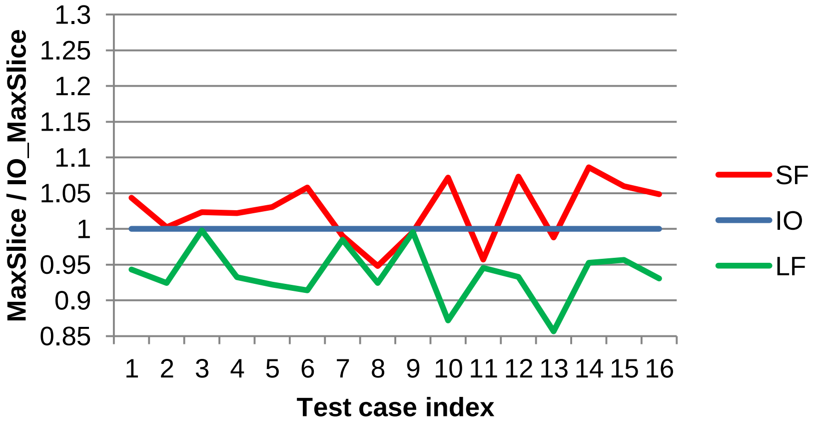
Vertex orders and Objectives

- Greedy coloring in the **same** order have different results for the problems (A) and (B).
- Consider the following orders combined with the First Fit heuristics:
- “SF” is the lexicographic order of pairs:
  Vertex weight, Sum of weights of the neighbors
- “LF” is the reversed order to LF.
- “IO” is a random order.

(A): Maximize num of colored nodes



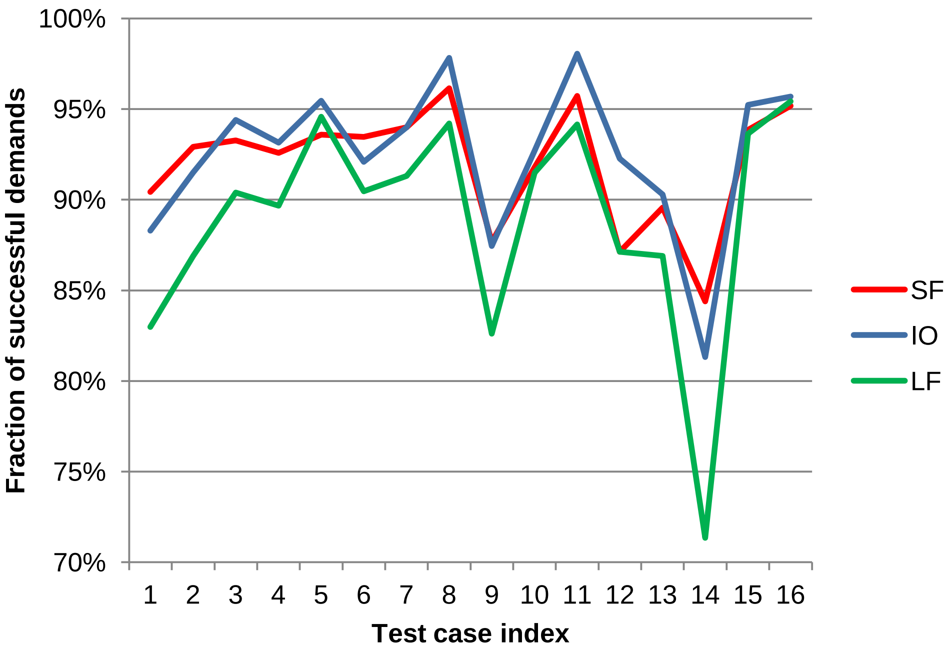
(B): Minimize max used slice’s index



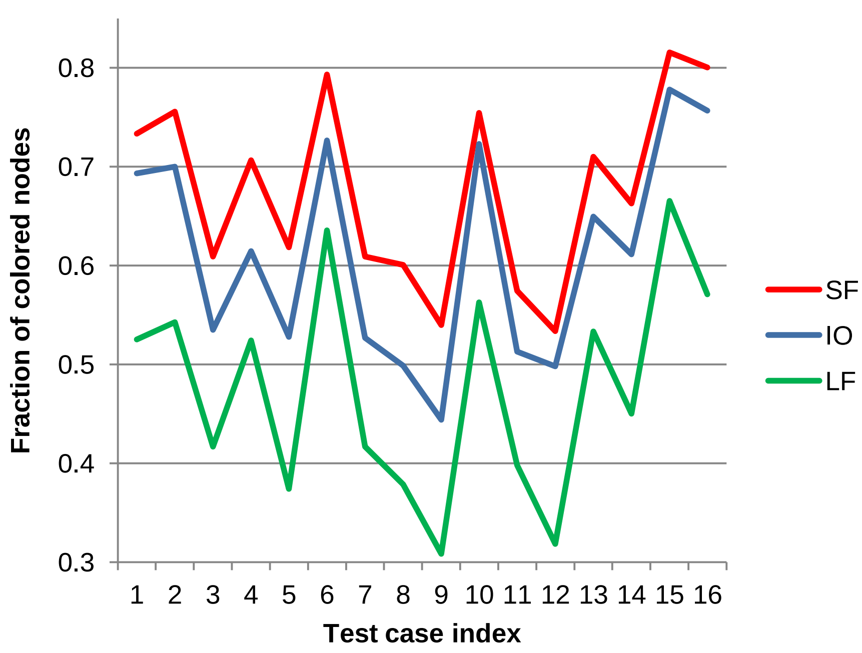
Scenarios for problem (A)

- Out of all scenarios the problem (A) is useful in the scenario with *W=L,* where *W* is the available total bandwidth, and *L* is the *load* introduced above
- This scenario significantly differs from (A) with *W~2L.*
- Simplistic ideas did not work, and we developed a more tricky Algorithm based on the above ideas

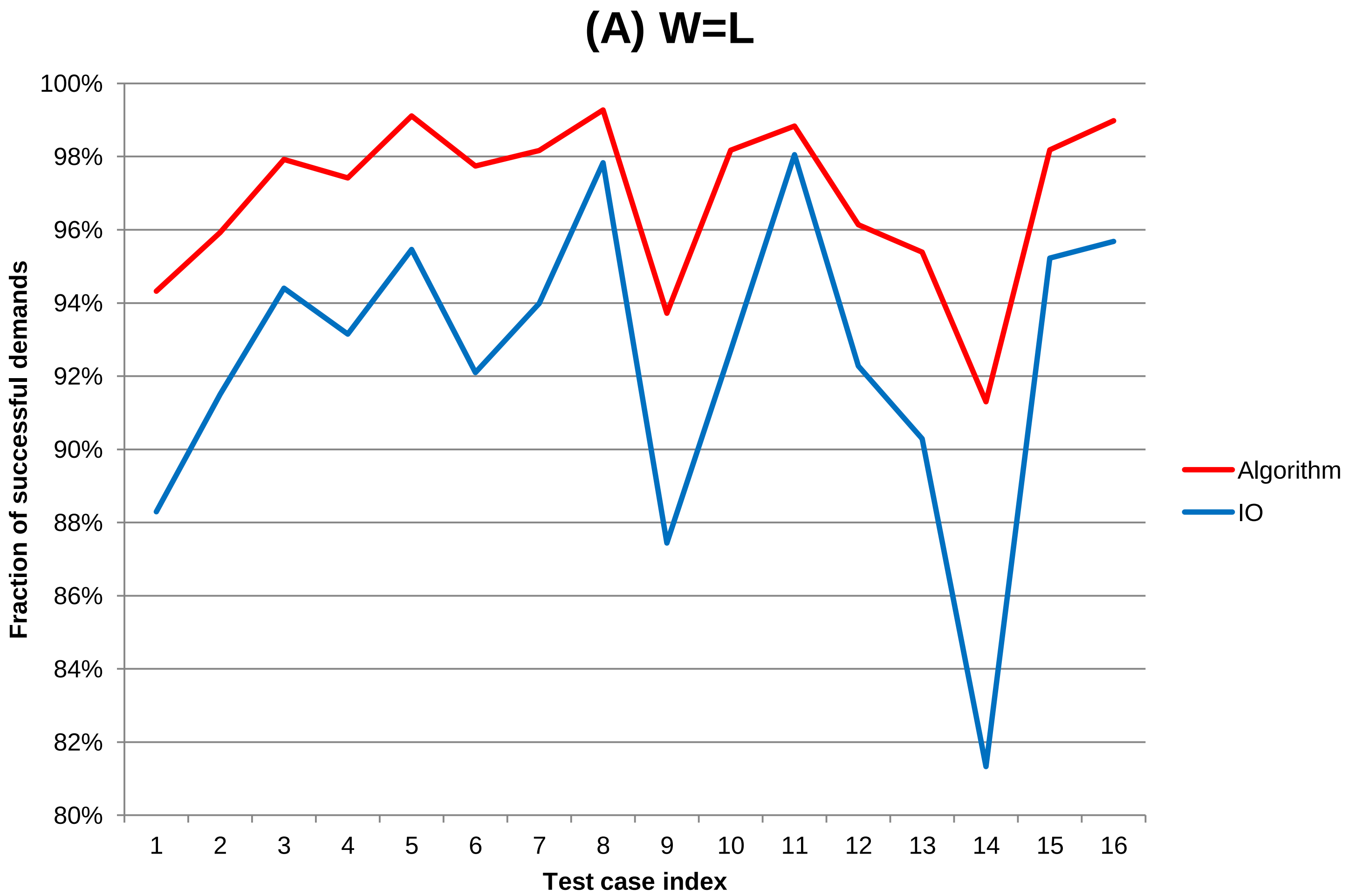
(A) W=L



(A) W~2L



Current results of our Algorithm



Input complexity and performance requirements

- **Both problems (A), feasibility, and (B), optimal *W* are NP hard**
- **We need a time efficient method able to work with**
  - 10-500 nodes
  - Up to 10000 demands
- **For small cases (about 10 nodes) the solution should be at 10% from the global optimum (provided e.g. by ILP solvers)**

Thank you

www.huawei.com