Managing Transfer-Based Datacenter Connections

Daniel Naro - Victoria Beleuta - María Gabriela Valdés

Problem Statement

Decide whether a transfer-mode request is accepted or not, the controller solves the routing and scheduled spectrum allocation (RSSA) problem.

The RSSA problem decides the elastic operations to be done so as to make enough room for the incoming request, and still respect the constraints.

Scenarios

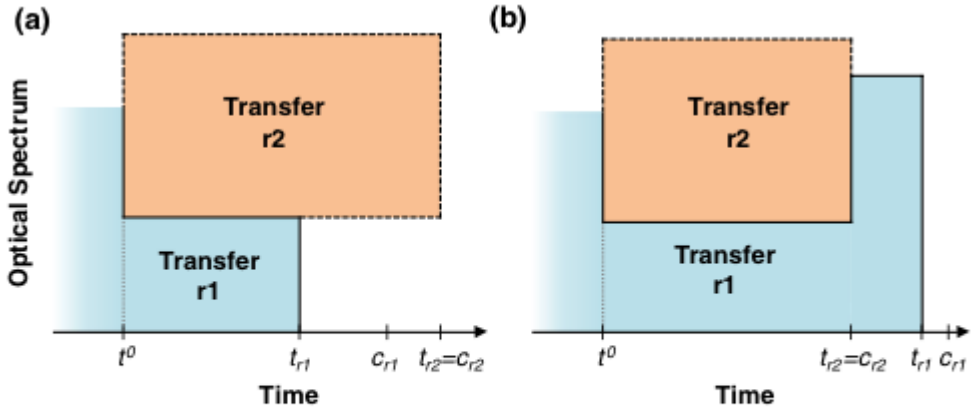


Fig. 2. Scheduling schemes for spectrum allocation. (a) Bitrate squeezing only and (b) with scheduled resource reservation.

Fig. 2(a) can be squeezed, thus increasing its scheduled completion time tr1, but without violating the committed completion time, cr1. Released resources can be used to set up an optical connection to support the requested transfer r2.

Scenarios

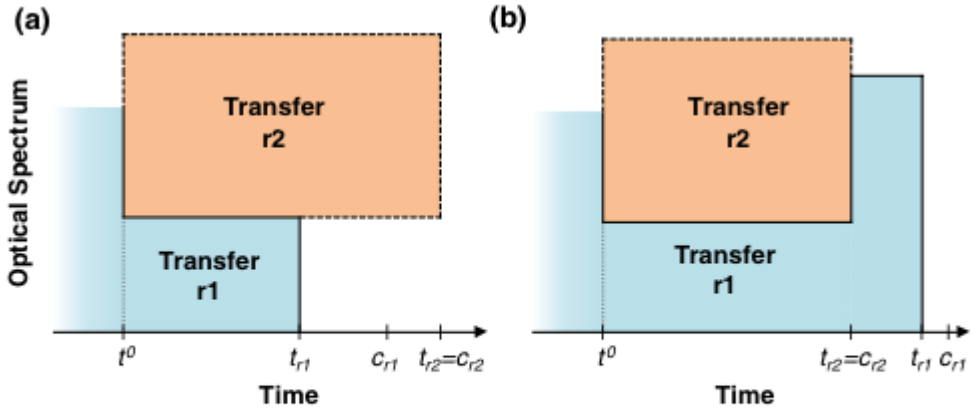


Fig. 2. Scheduling schemes for spectrum allocation. (a) Bitrate squeezing only and (b) with scheduled resource reservation.

Fig. 2(b), where to ensure the committed completion time for transfer r1, some resources need to be returned to that transfer as soon as transfer r2 is completed.

Implementations:

- ILP
- Greedy Algorithm + Local Search
- GRASP + Local Search

ILP Problem Statement

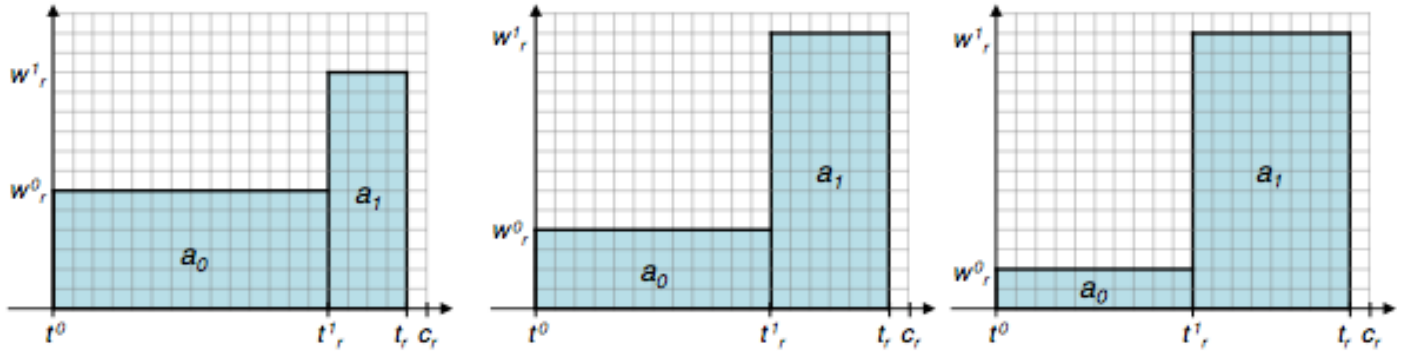
Given:

- network topology G(L,E);
- datacenters $D \subseteq L$;
- a set S of available spectrum slices for each link;
- capacity and number of flows in each location;
- a set R of current transferences in progress;
- for $r \in R$, the tuple (o_r; d_r; v_r; c_r; r_r; s_{0r}; s_{1r}; t_{1r}; t_r);
- the new transfer-mode request with (o_r ; d_r ; v_r ; c_r) and t_0 denoting the current time;

ILP Model

Precomputation: sets of feasible rectangles for each ongoing transference;

- set A0 contains rectangles specifying allocation w_{0r} from time t_0 to t_{1r};
- set A1 specify scheduled allocation w_{1r} from t_{1r} to t_r;
- matrix β_{01} indicates whether rectangle pair $a_0 \in A0$ and $a_1 \in A1$ is a feasible scheduling;



Problem Statement

Output:

- the route (r_r);
- the slot allocation (s_{0r});
- the scheduled completion time (t_r) for the new transference request;
- the bitrate of the connection $b(s_{0r}; l(r_r))$;
- the new spectrum allocation (w_{0r});
- scheduled reallocation ($w_{1r};t_{1r}$);
- completion time (t_r);

Problem Statement

Objective:

minimize the number of connections to be rescheduled to make room for the incoming request.

Greedy Algorithm

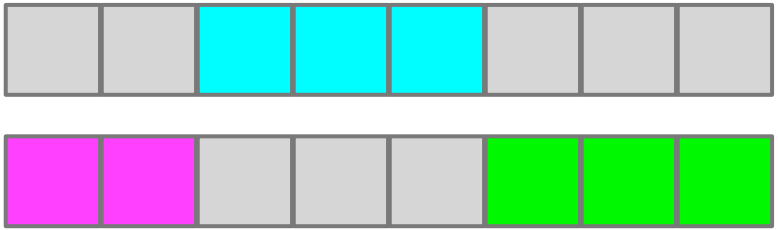
- get minimum admissible bitrate for requested transference;
- get shortest-path for route required;
- in case a feasible solution is found, the slot with maximum width is assigned to the required transfer;
- otherwise, adjacent transfers are asked to free necessary slots;

GRASP Algorithm

C = path(source, destination);
q_c = sizeof(path);
α = 0.3;
RCL = q_{min} + α (q_{max} - q_{min});

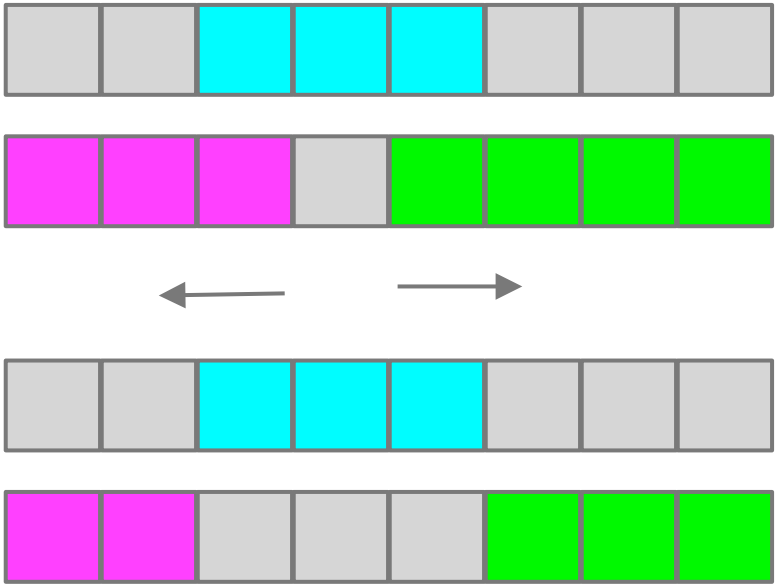
Main Algorithm Structure

Trivial Case:



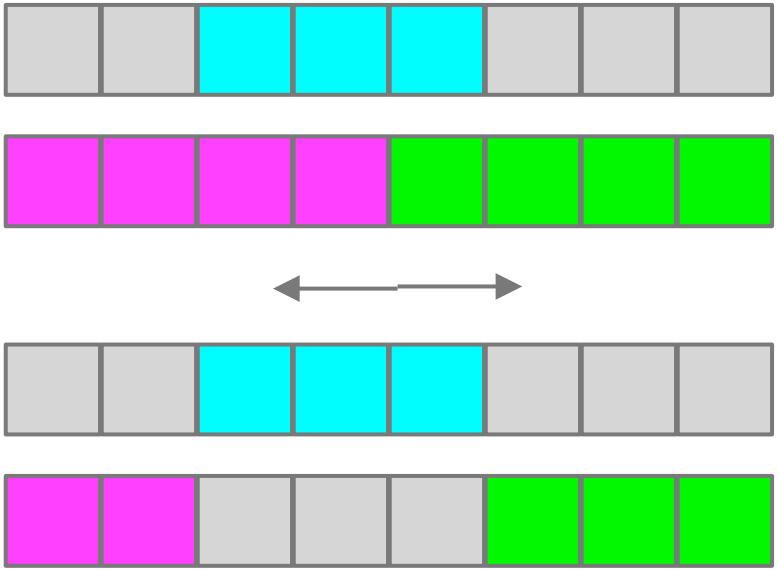
Main Algorithm Structure

Half Hard Case:



Main Algorithm Structure

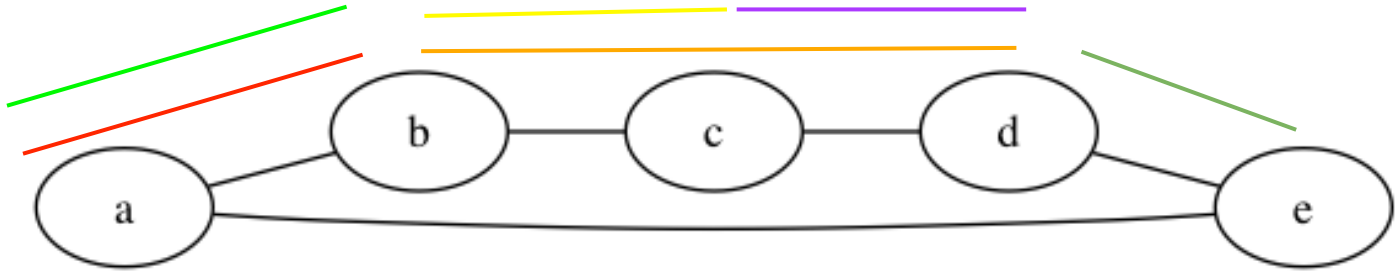
Hard Case:



Example 1

inputTestHardAndTrivial

RT: a - e



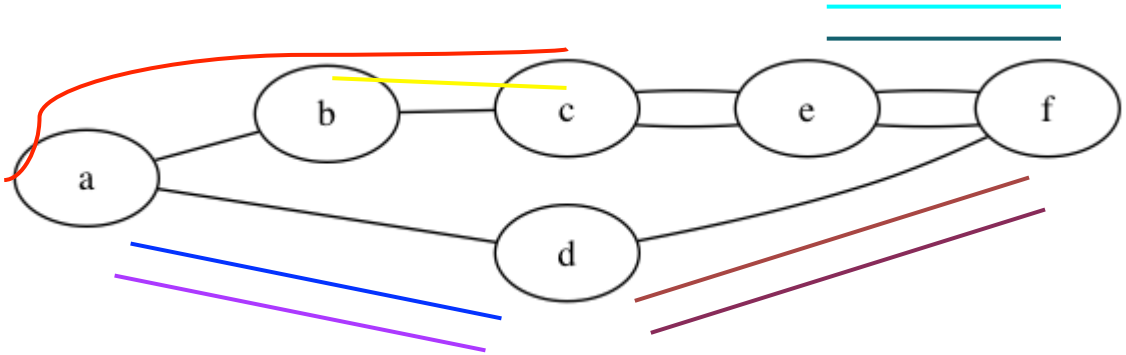
	CPLEX	Greedy	GRASP
Result	0	0	0
Time	37.04	0.8	139.6

NOTE: All CPLEX time correspond to “CPLEX MIP Optimization” in the Profiler

Example 2

inputSeeDifferencesGreedyCplex

RT: a - e

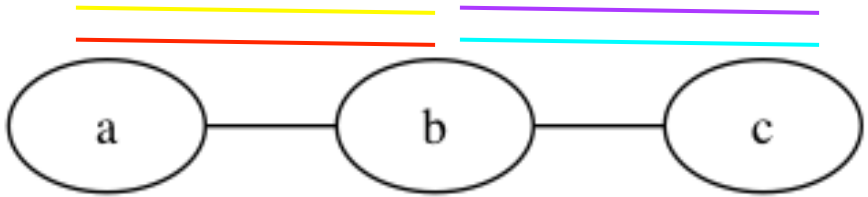


	CPLEX	Greedy	GRASP
Result	1	3	2
Time	55.567	3	134

Example 3

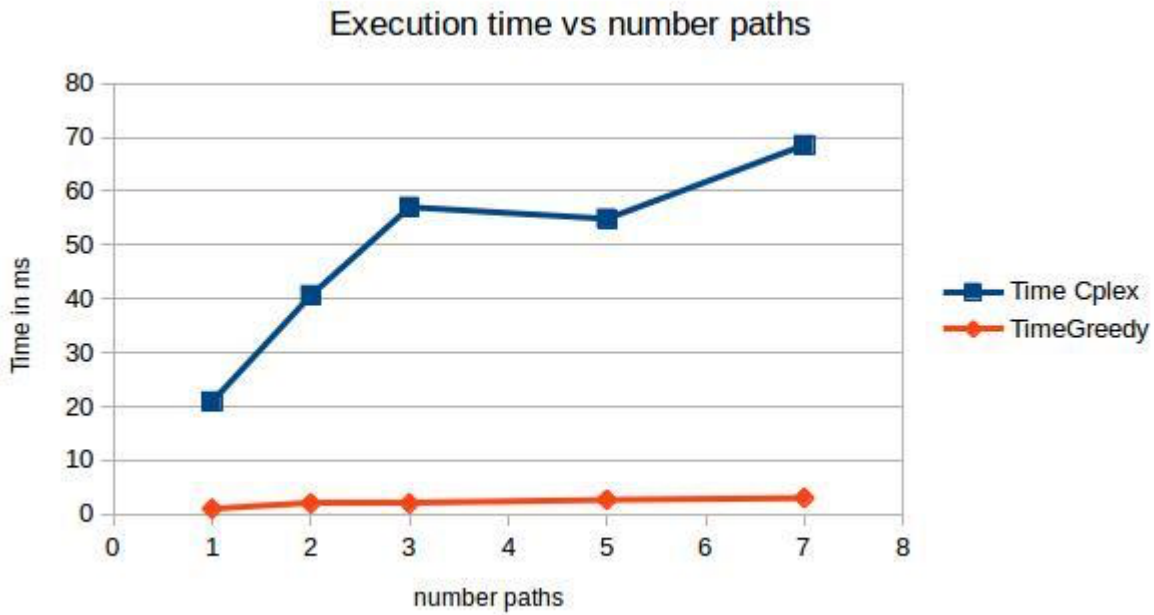
inputTestHalfHard

RT: a - e



	CPLEX	Greedy	GRASP
Result	1	2	1
Time	50.58	1.4	180

Example multiple hard paths



Local search

- Input: all the free slices obtained after reschedules
- We iterate over them and check if there is a subset big enough
- Can we undo reschedules with the space we do not use?

Heuristic Versus ILP Model

- Greedy approach is the fastest and with the local search approximate quite well the optimum.
- GRASP can find better solutions, but takes longer
- ILP always find the best solution and is the slowest if we include every step of the execution

Thank you!