Minimizing the cardinality of a real-time task set by automated task clustering

Antoine Bertout, Julien Forget and Richard Olejnik

LIFL, University Lille1, France

Context

- **Motivation:** High-level functionalities are numerous but real-time OS are limited to several tens of tasks
- **Objective:** Grouping tasks together in order to reduce their number
  - while keeping the system functionally equivalent
  - while preserving schedulability
- **Model:** Uniprocessor and independent tasks

Clustering

Cluster (or task) τ_{ij} from clustering of τ_i and τ_j :

- $C_{ij} = C_i + C_j$
- $\mathbf{D_{ij} = \min(D_i, D_j)}$
- $T_{ij} = T_i = T_j$

Validity conditions

- **A clustering is valid if**

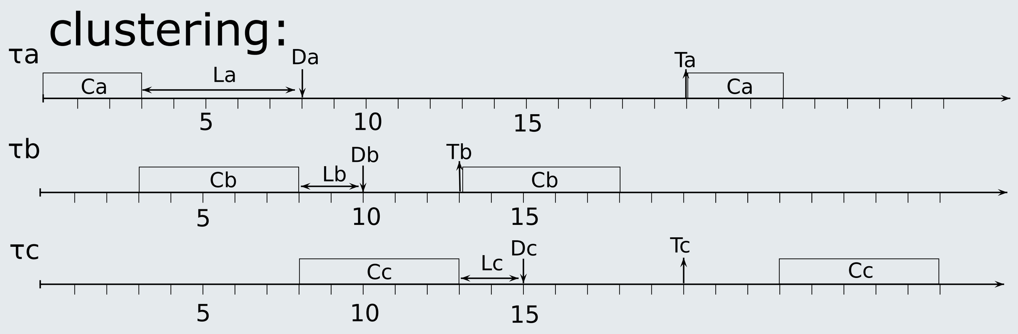
1. Tasks periods of the two tasks are equal.

2. Host task (that one with the shortest deadline to ensure initial constraints) has sufficient laxity L to incorporate the other one ($L_i = D_i - C_i$).

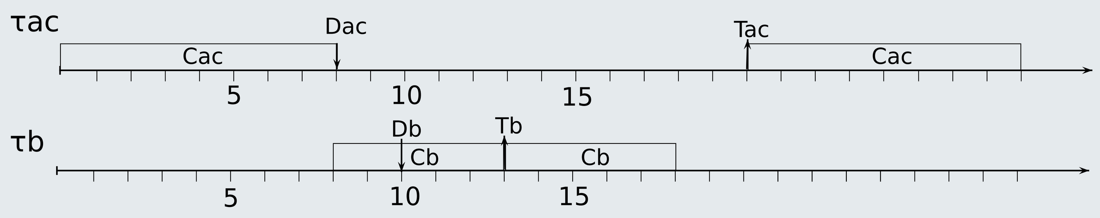
3. Whole system is still schedulable after clustering.

Schedulability

- **Clustering's impact on schedulability**
  - A system may not be still schedulable after clustering:

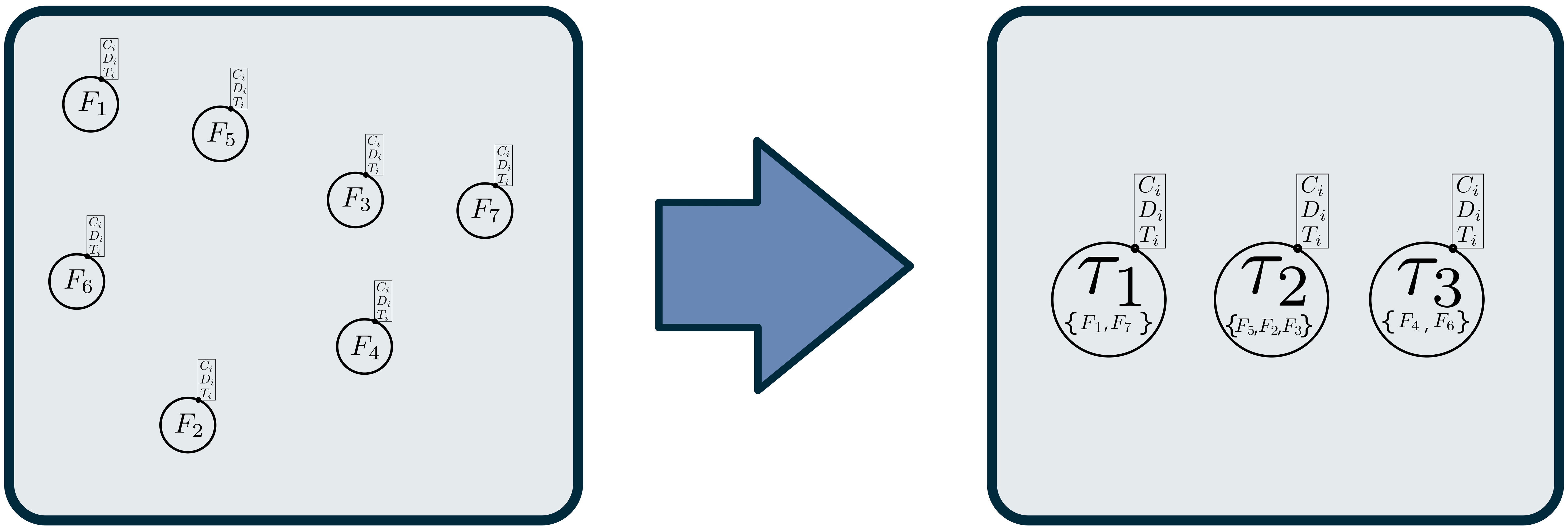


In the second diagram, τ_b missed its first deadline after clustering of τ_a and τ_c



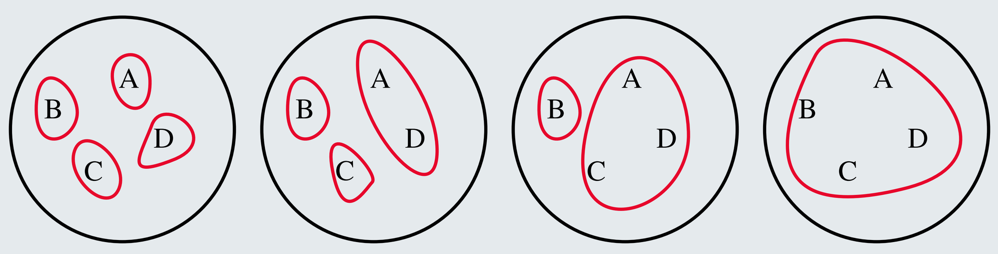
- Schedulability must be checked after each clustering

- **Schedulability tests**
  - Use of sufficient or exact tests ensuring that a task set considered schedulable by the tests is schedulable, even if the sufficient tests are pessimistic
  - *Exact tests* considered are boolean tests or response time analysis (RTA) tests with pseudo-polynomial complexity
  - *Sufficient tests* considered give an approximation of the RTA with often linear complexity



Search space

- **Partitioning problem**
  - Example: a few possibilities to group 4 tasks



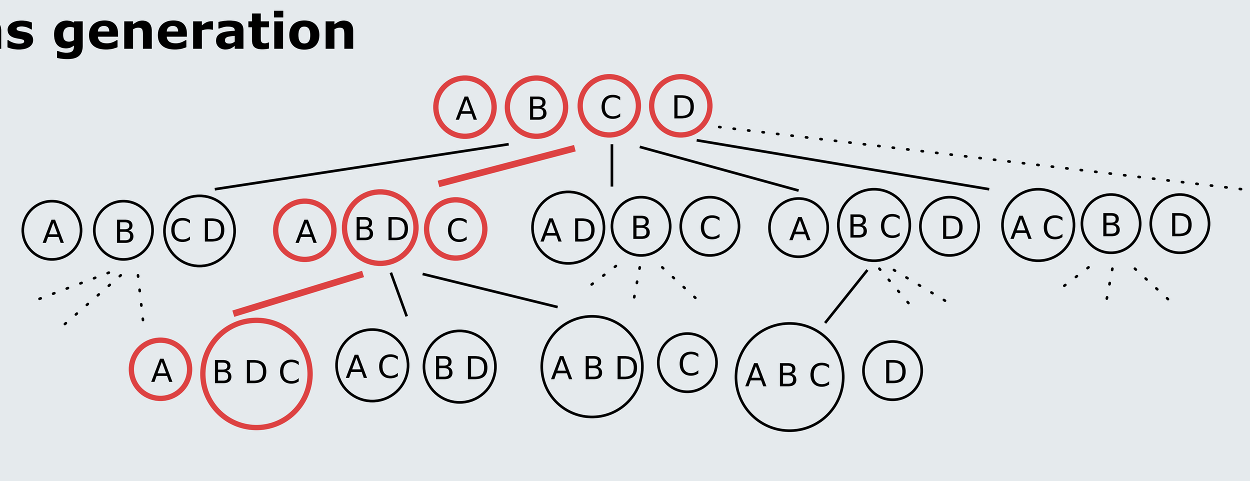
- **Combinatorial explosion**
  - Number of possibilities in the Bell number range:

$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$ with $\mathbf{B_{500} = 10^{844}}$

  - Expermiments show that exhaustive search through partitions is not practicable even with linear tests

Heuristic approach

- **Partitions generation**



- **Heuristic cost function**
  - We need a heuristic function to select the best local candidate at each partitions generation
  - Following RTA test or its approximation with worst response time of τ_k denoted R_k , the closer to 1 $\frac{R_k}{D_k}$ is, the less we have margin to group τ_k with another task (or cluster)
  - We have a heuristic cost function $h(\mathcal{S})$, such that $h(\mathcal{S}) = \sum_{k=0}^{|\mathcal{S}|} \frac{R_k}{D_k}$