Optimizing Vehicle-to-Cloud Data Transfers using Soft Real-Time Scheduling Concepts

Jean IBARZ [1,2] Olivier Flébus [1]

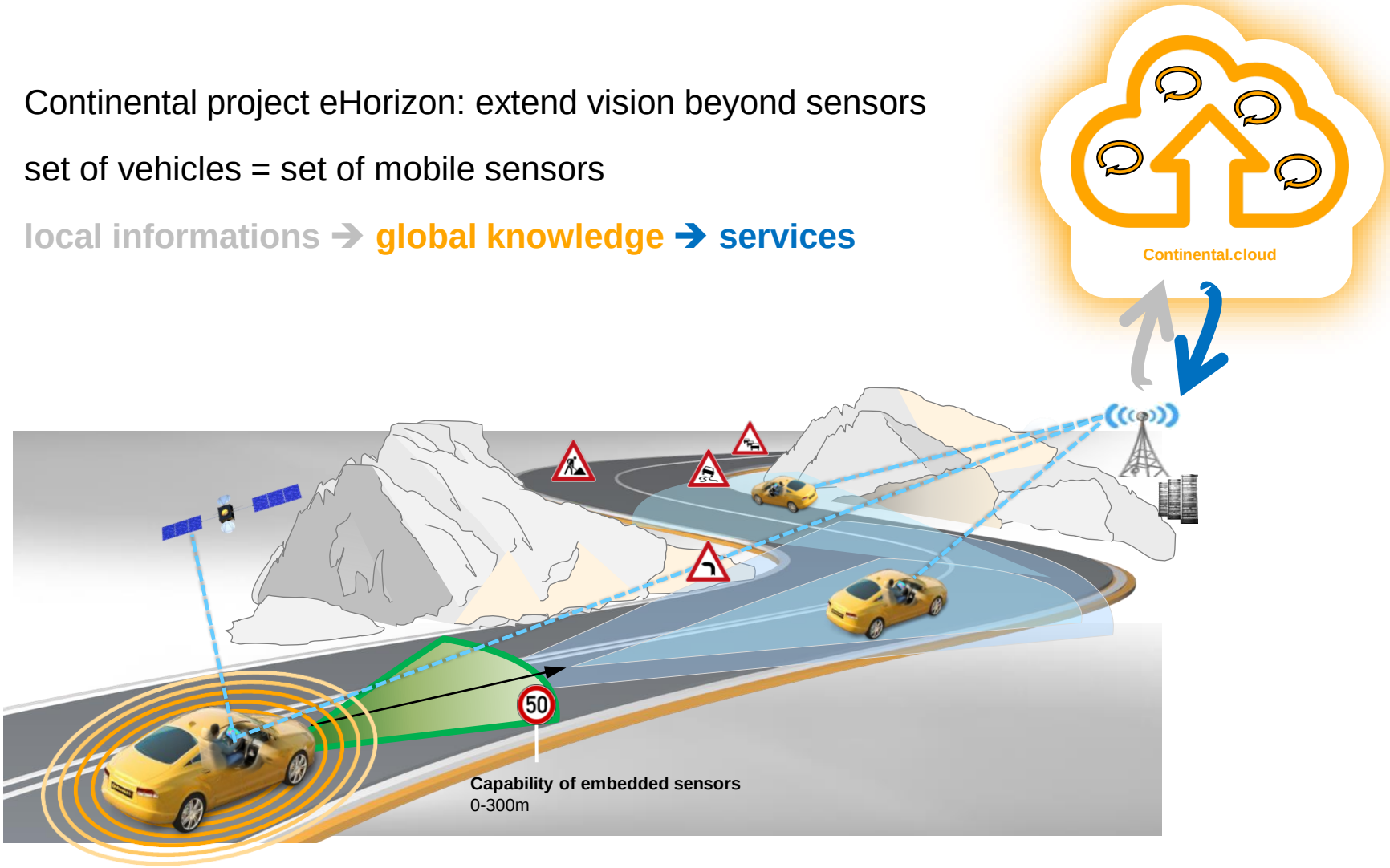
Matthieu ROY [2], Michaël LAUER [2], Jean-Charles FABRE [2]

[1] Continental Digital Services, Toulouse, France

[2] CNRS-LAAS, Toulouse, France

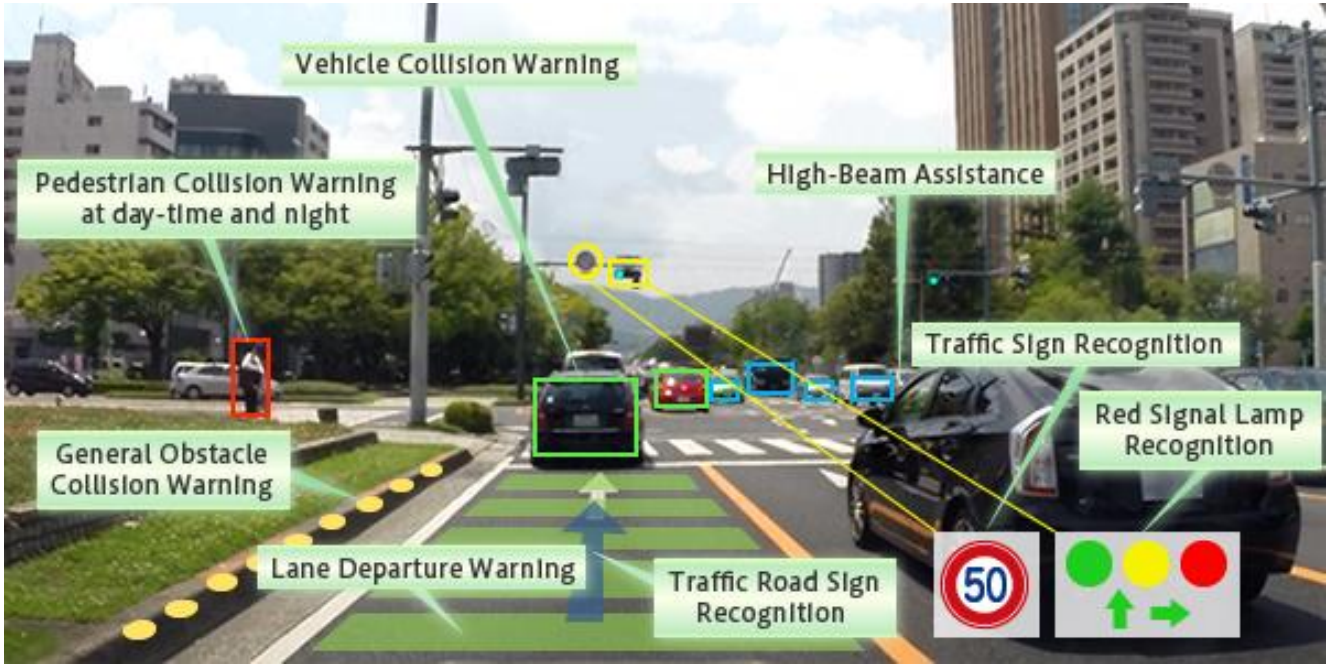
Context

- Continental project eHorizon: extend vision beyond sensors
- set of vehicles = set of mobile sensors
- local informations ➔ **global knowledge** ➔ **services**



Sensing ➔ events

- vehicles generate events
- multiple events are packed into a message
- an event characterize the vehicle surroundings in space-time



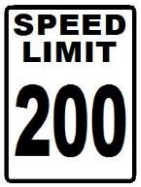
source: Toshiba image-recognition for ADAS, Green Car Congress

Scalability

- millions of connected vehicles ➔ transmission of all messages not sustainable:
  - technical constraints (cellular network capacity)
  - economical constraints (storage, processing, etc.)

Scalability

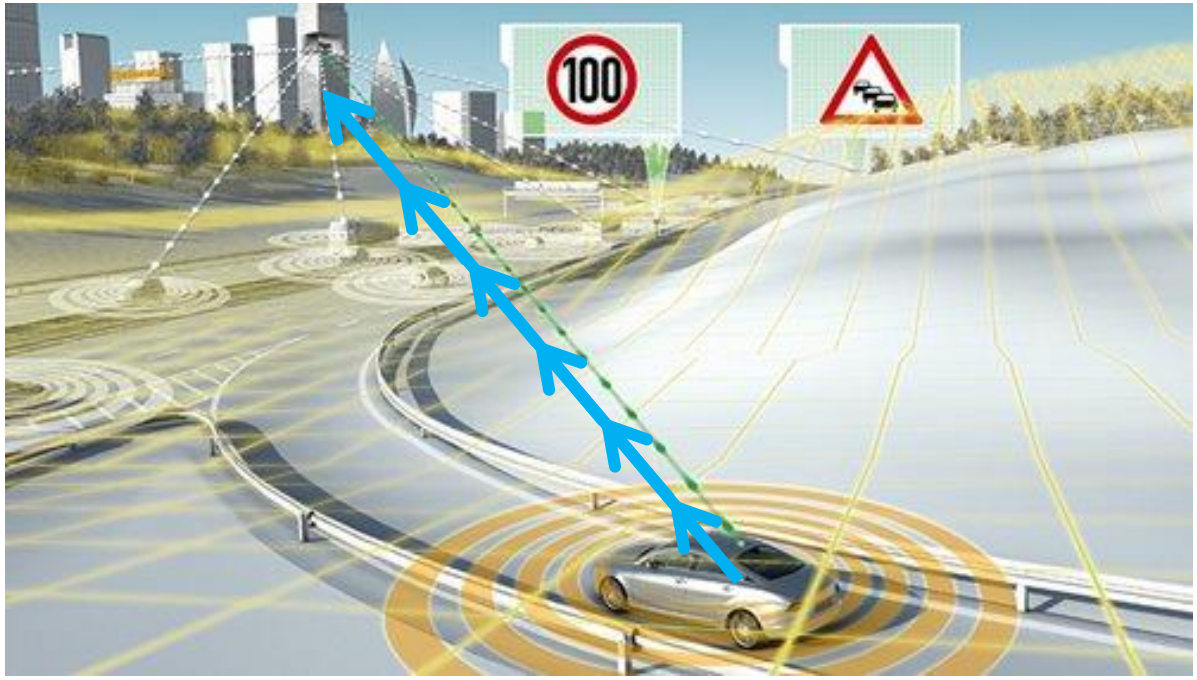
- millions of connected vehicles ➔ transmission of all messages not sustainable:
  - technical constraints (cellular network capacity)
  - economical constraints (storage, processing, etc.)



problem: optimize data flow under finite resources constraints

Our problem

- V2C data flow only
- cellular network highly overloaded ➔ schedule messages to be uploaded to the cloud
- system purpose is to improve transportation efficiency ➔ non safety critical



Approach

- optimization locally to each vehicle (scalability)
- message value: explicit choice criteria for scheduling decisions
- analogy with soft real-time formalism to solve the problem
- analysis of existing algorithms

Approach

- optimization locally to each vehicle (scalability)
- message value: explicit choice criteria for scheduling decisions
- analogy with soft real-time formalism to solve the problem
- analysis of existing algorithms

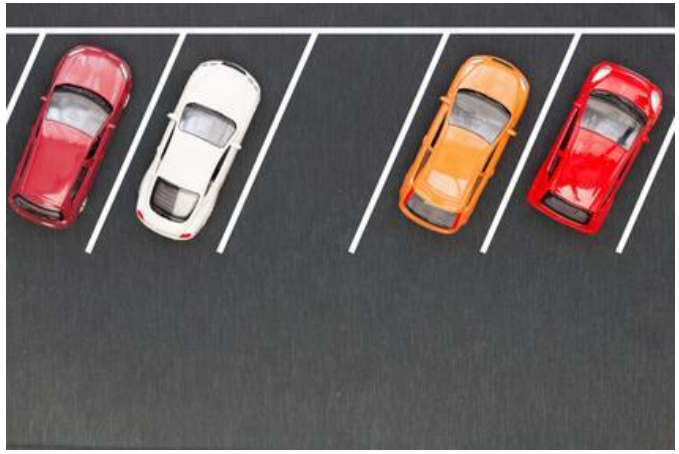
➔ **extension of analysis methods**

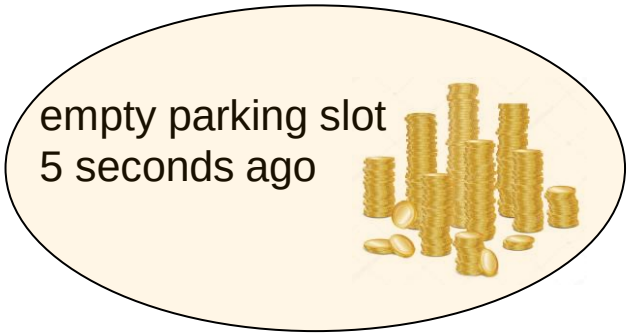
Outline

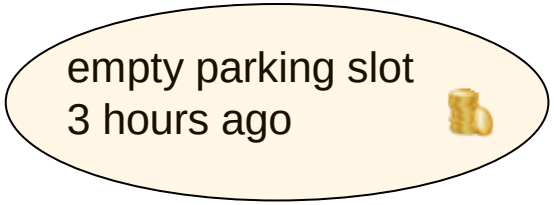
- value-based scheduling problem
- performance evaluation & results
- conclusion & future work

Value-based scheduling problem

- goal: maximize the cumulative value of all transmitted messages
- value of a message = contribution to the quality of service granted to the end-user
- **the challenge:** value of an event evolves over time







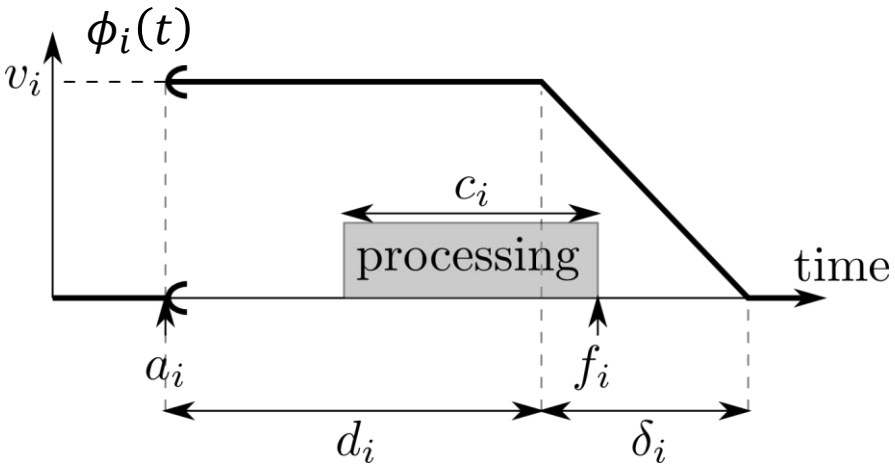
Our problem // real-time scheduling problem

our scheduling problem	a real-time scheduling problem
a message	a real-time job
communication interface	processor
bandwidth	processor’s speed
transmission of a message	processing of a job

maximize the cumulative value of
all transmitted messages // all completed jobs

Modelization: soft real-time (Buttazzo[1])

- model is simple and memory efficient ➔ appropriate for embedded systems
- job J_i: $< a_i, c_i, v_i, d_i, \delta_i >$
- utility function ϕ_i of a job : completion date ↦ utility granted to the system



a_i: arrival date
v_i : initial value
c_i : execution units
d_i : firm deadline
δ_i : lateness limit
f_i : completion date

[1] Giorgio C Buttazzo. 2011. Hard real-time computing systems: predictable scheduling algorithms and applications.

Performance evaluation

- goal: maximize cumulative value of all completed jobs

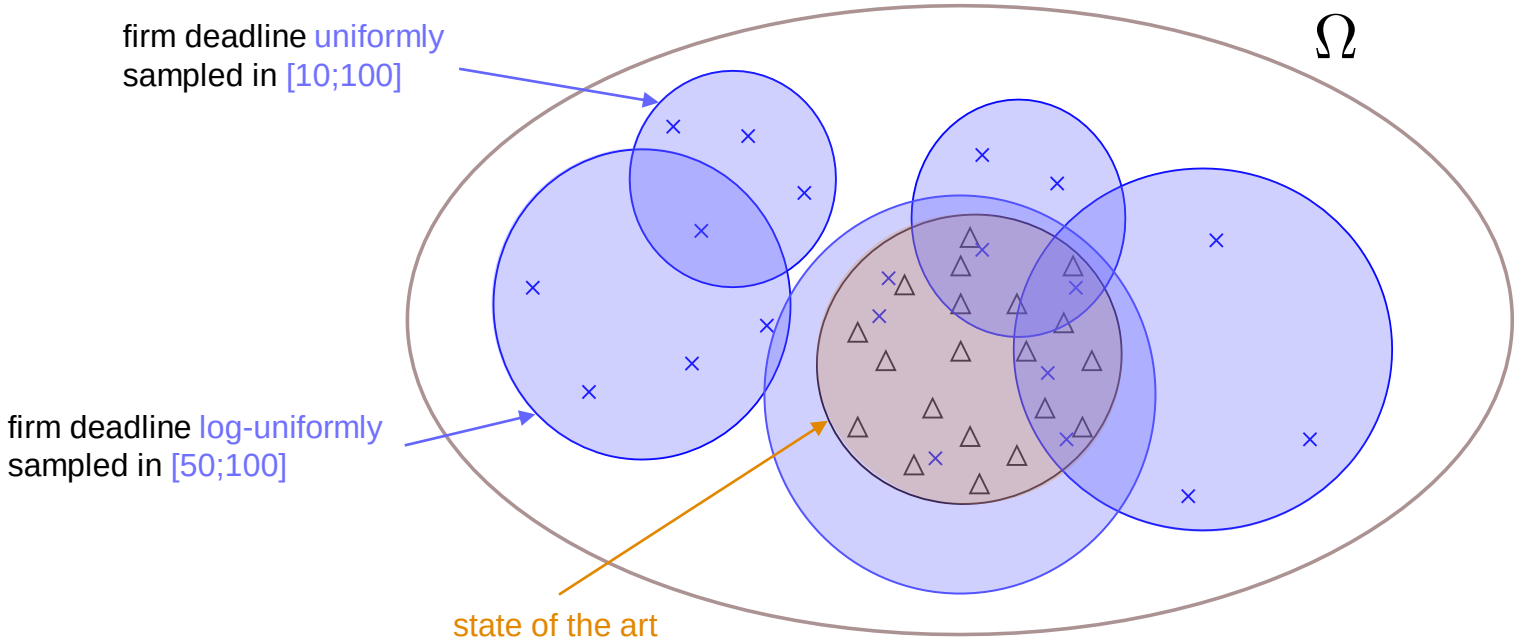
cumulative value: $\sum_i \phi_i(f_i)$

where $\phi_i(f_i)$ is the utility value granted to the system by the completion of job i at date f_i

- optimal algorithm: maximize cumulative value under any scenario
- **but** requires knowledge of arrival date of future jobs ➔ not implementable
- lot of (non-optimal) on-line algorithms already exists: which one to choose ?
  - worst case evaluation: not the most relevant because the system is not critical
  - experimental evaluation: real scenarios not available yet ➔ synthetic scenarios

Performance evaluation: scenarios generation

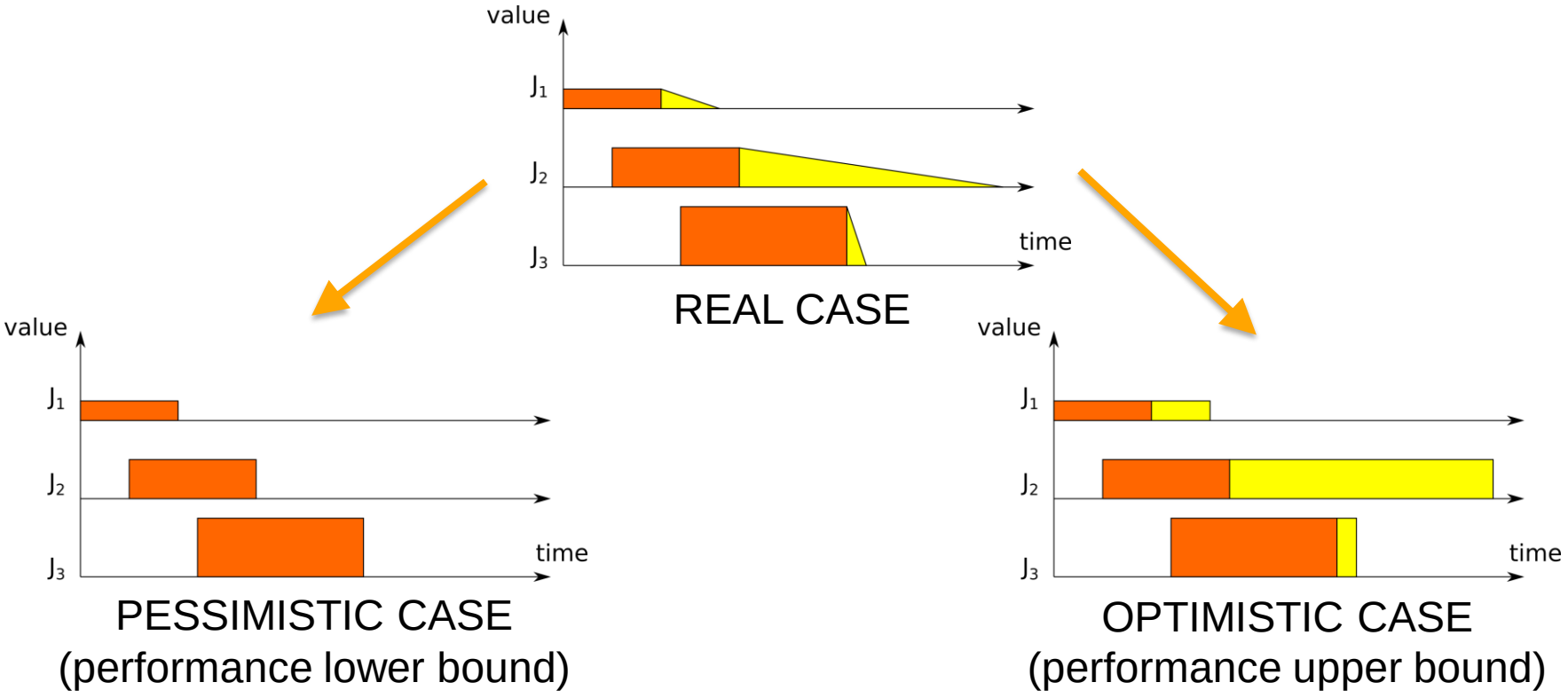
- synthetic scenarios generation method based on Aldarmi & Burns [2]
- **base method:** sample jobs from random variables (arbitrarily fixed)
- **proposed extension:** increase genericity by varying the random variables



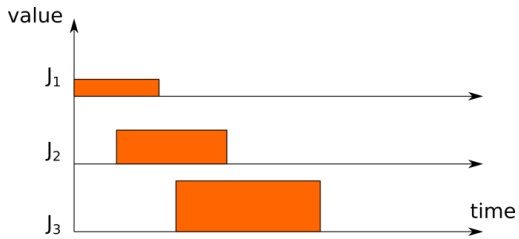
[2] Saud Ahmed Aldarmi and Alan Burns. 1999. Dynamic value-density for scheduling real-time systems.

Performance evaluation: baseline

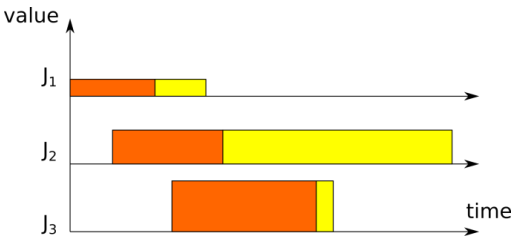
- baseline for comparison: optimal clairvoyant algorithm (knowing the future)
- optimal scheduling too complex to find: search for performance bounds instead



Performance evaluation: baseline



PESSIMISTIC CASE



OPTIMISTIC CASE

› problem with firm jobs: find a **schedulable subset of jobs** that maximizes **cumulative value**

↕

exact schedulability test

› MILP formulation:

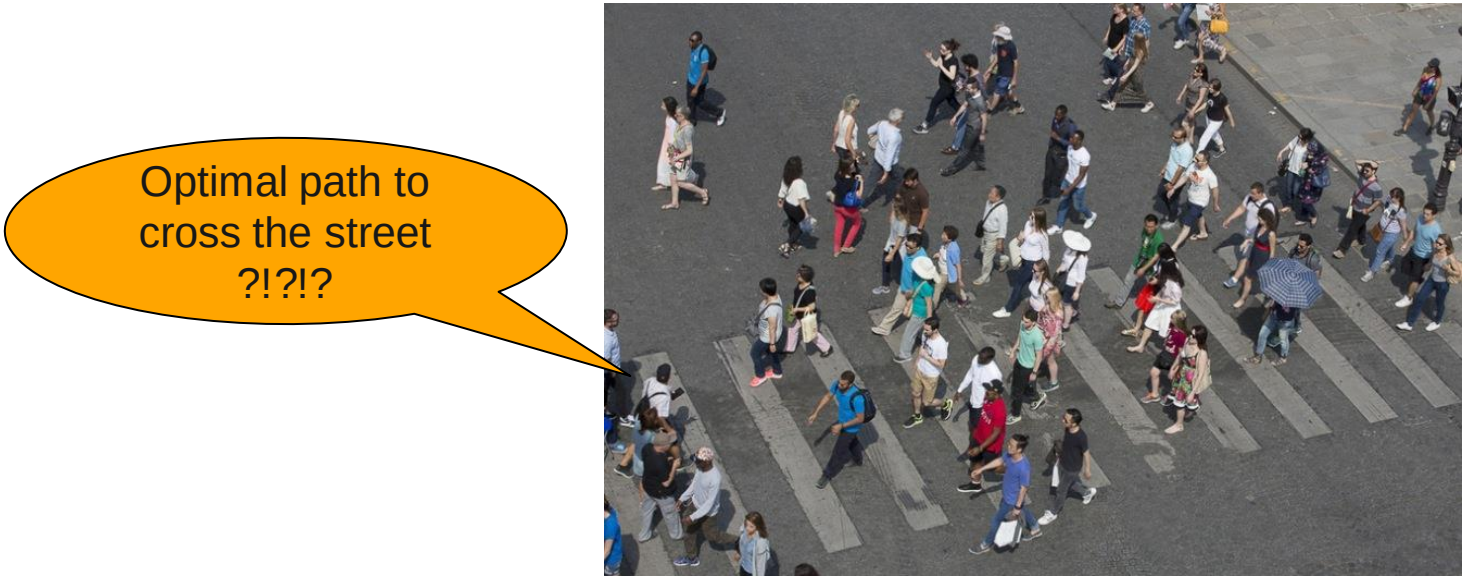
$$maximize \sum_i y_i v_i$$

$$subject\ to\ \forall (i,k), a_i < d_k, \sum_{j:[a_j,d_j]\subseteq[a_i,d_k]} y_i c_j \leq d_k - a_i$$

› GLPK solver ➔ (pseudo-)optimal solution

Performance evaluation: evaluated algorithms

- greedy algorithms are good candidates:
  - low calculus complexity ➔ adapted to embedded systems
  - short sighted decisions ➔ adapted to high uncertainty about future evolution (job arrivals, variation of processing speed, transmission failure, etc.)



Performance evaluation: evaluated algorithms

- greedy algorithms are good candidates:
  - low calculus complexity ➔ adapted to embedded systems
  - short sighted decisions ➔ adapted to high uncertainty about future evolution (job arrivals, variation of processing speed, transmission failure, etc.)

Performance evaluation: evaluated algorithms

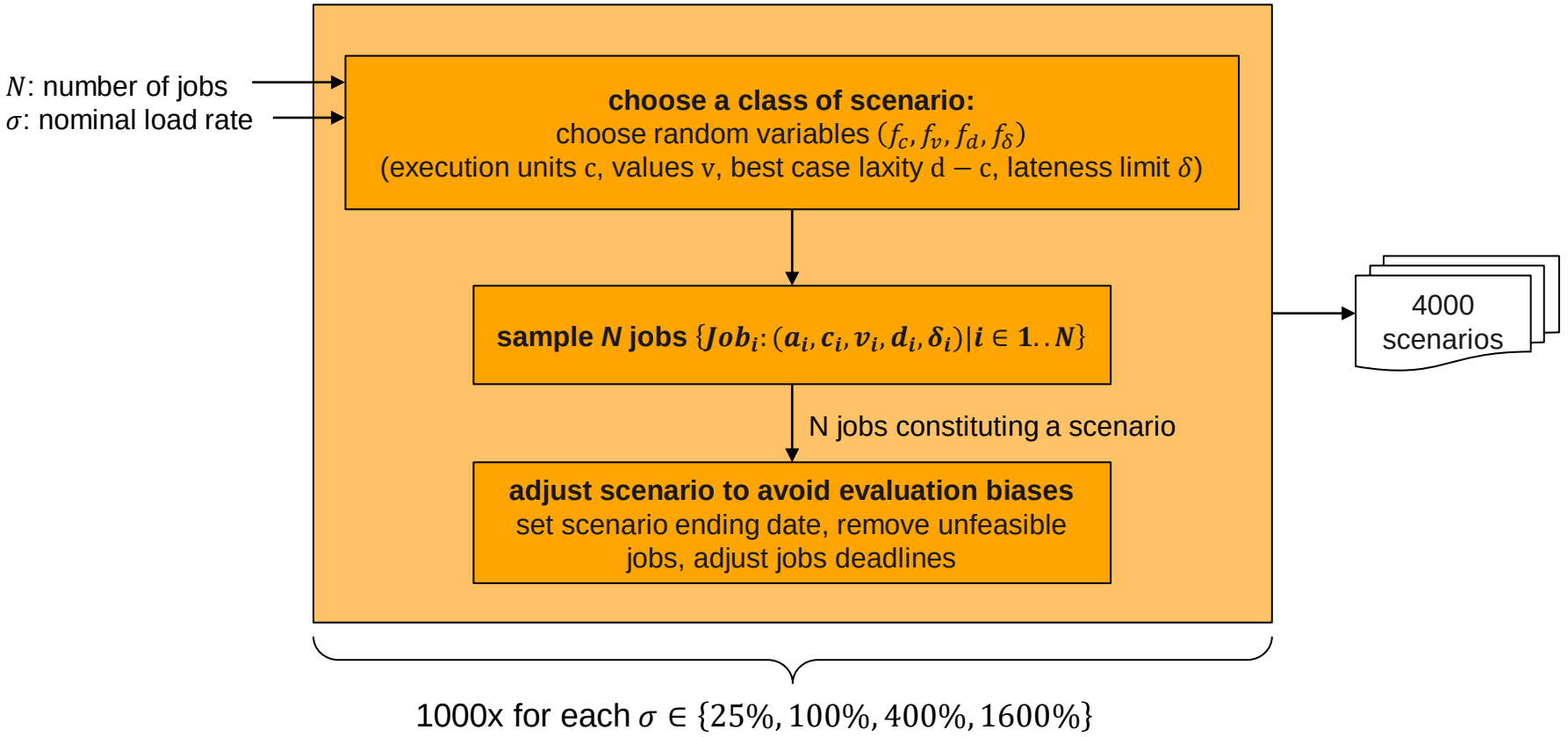
Algorithm	Heuristic (priority ∝ heuristic score)
SVD[1]	base value / base cost
SDVD[2]	current value / base cost
DVD1[3]	current value / remaining cost
DVD2[4]	current value / remaining cost squared
DTD1[5]	value at current best case completion date / remaining cost
DTD2[6]	value at current best case completion date / remaining cost squared

color code: static parameter, dynamic parameter, dynamic parameter with foreseeing future

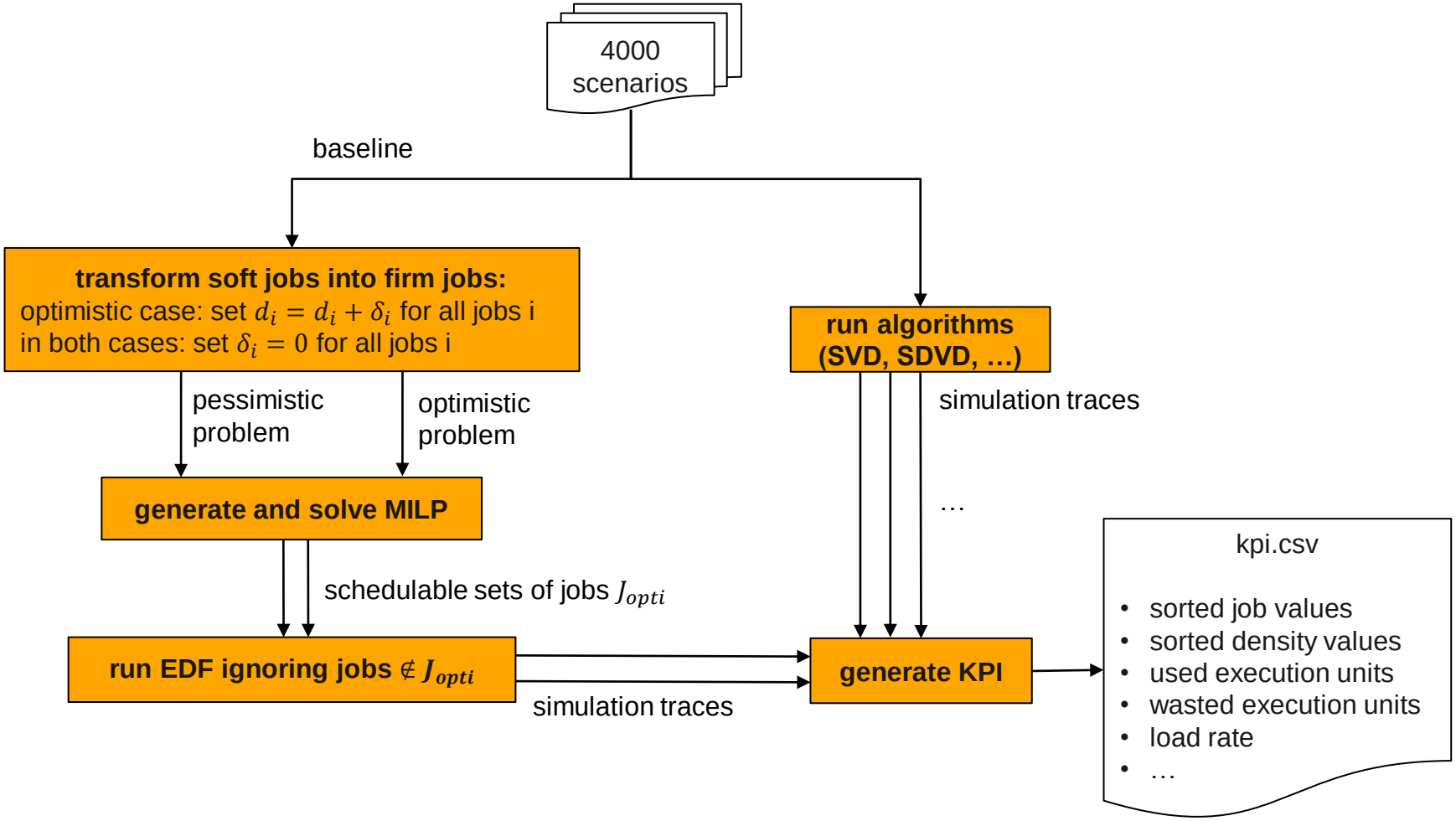
cost: execution units to be processed to complete the job

[1] Static Value Density, [2] Semi Dynamic Value Density, [3] Dynamic Value Density,
[4] Dynamic Value Density Squared, [5] Dynamic Timeliness Deadline,
[6] Dynamic Timeliness Deadline Squared

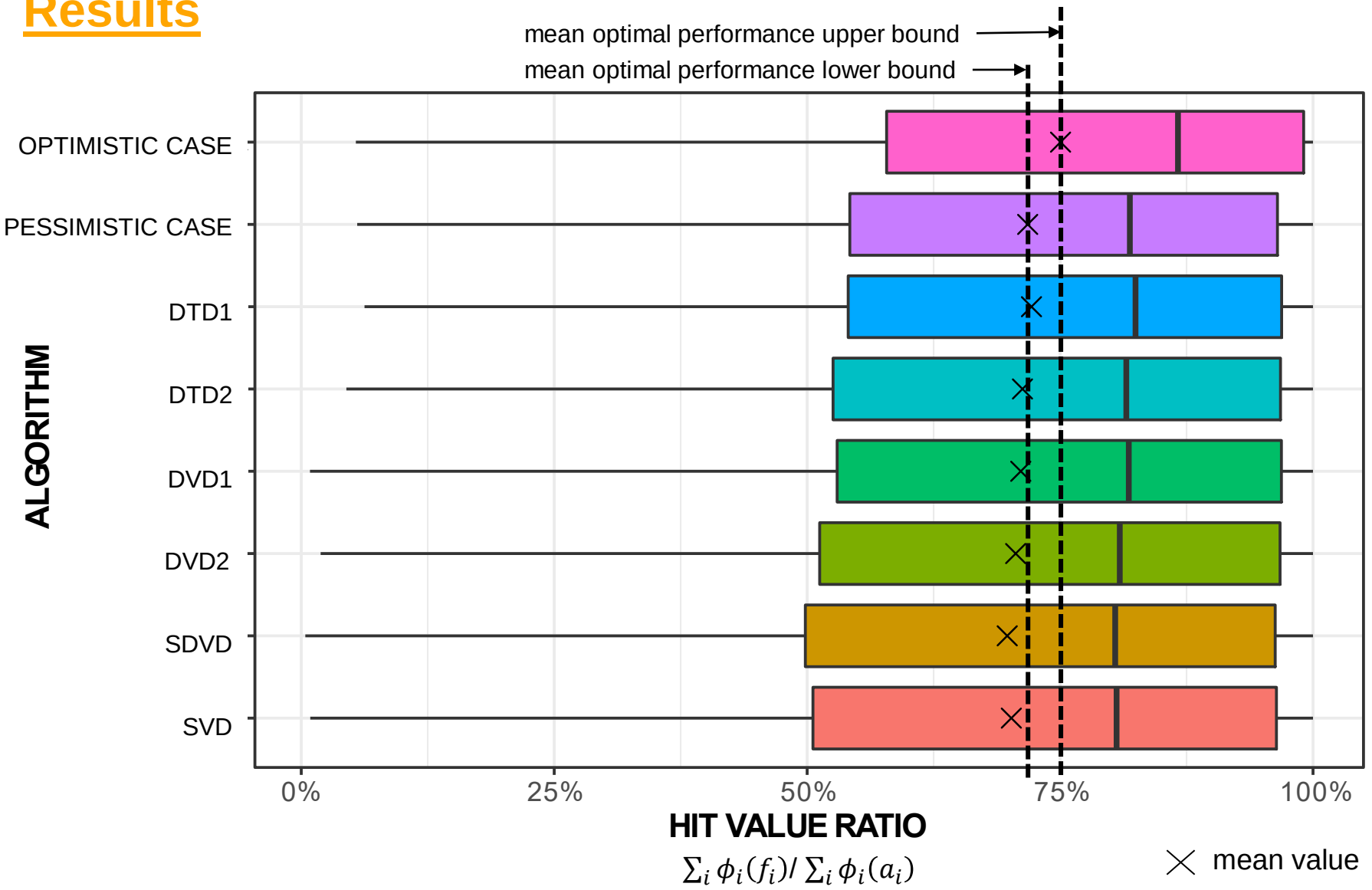
Experimental evaluation: scenario generation



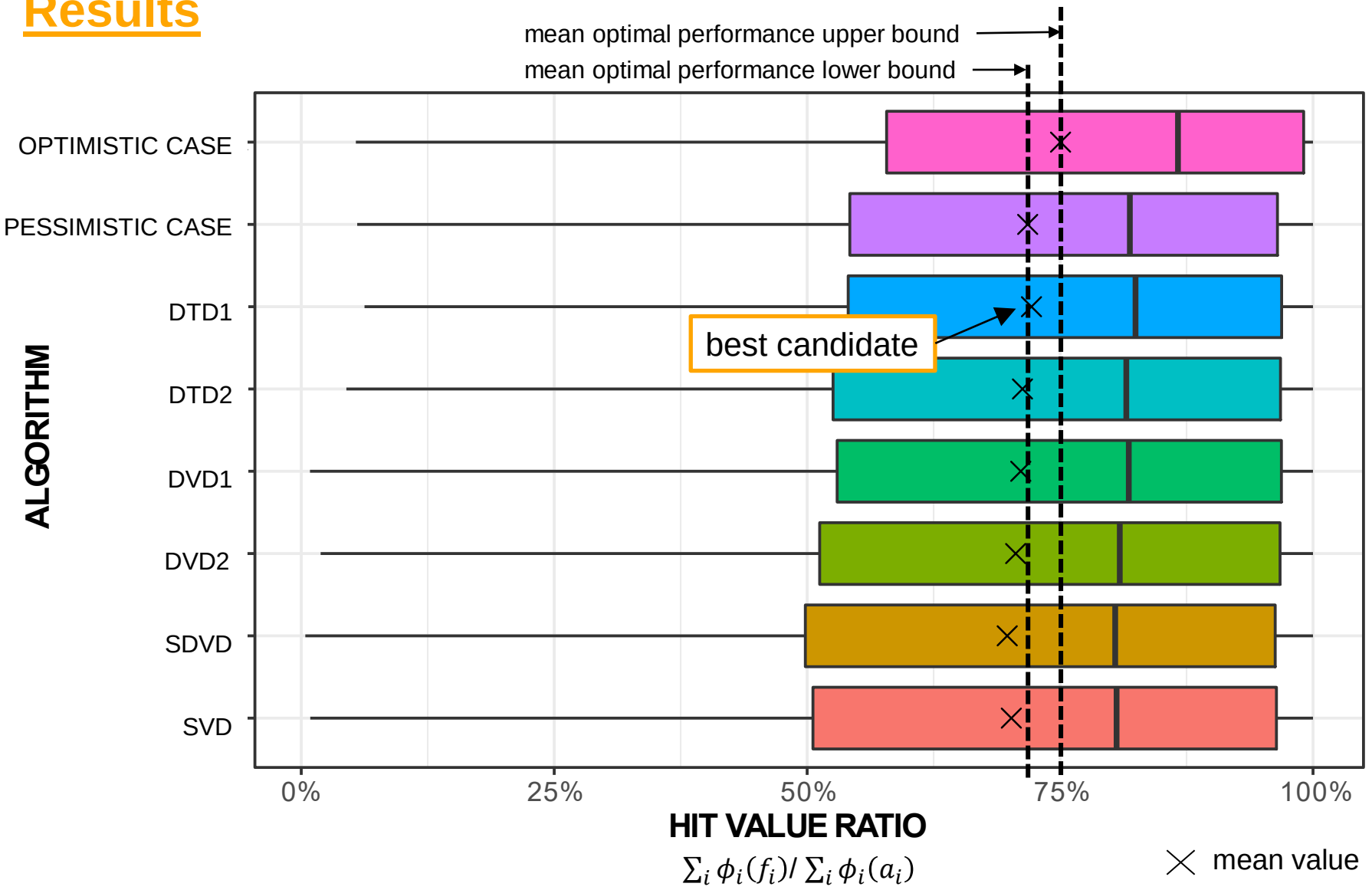
Experimental evaluation: KPI generation



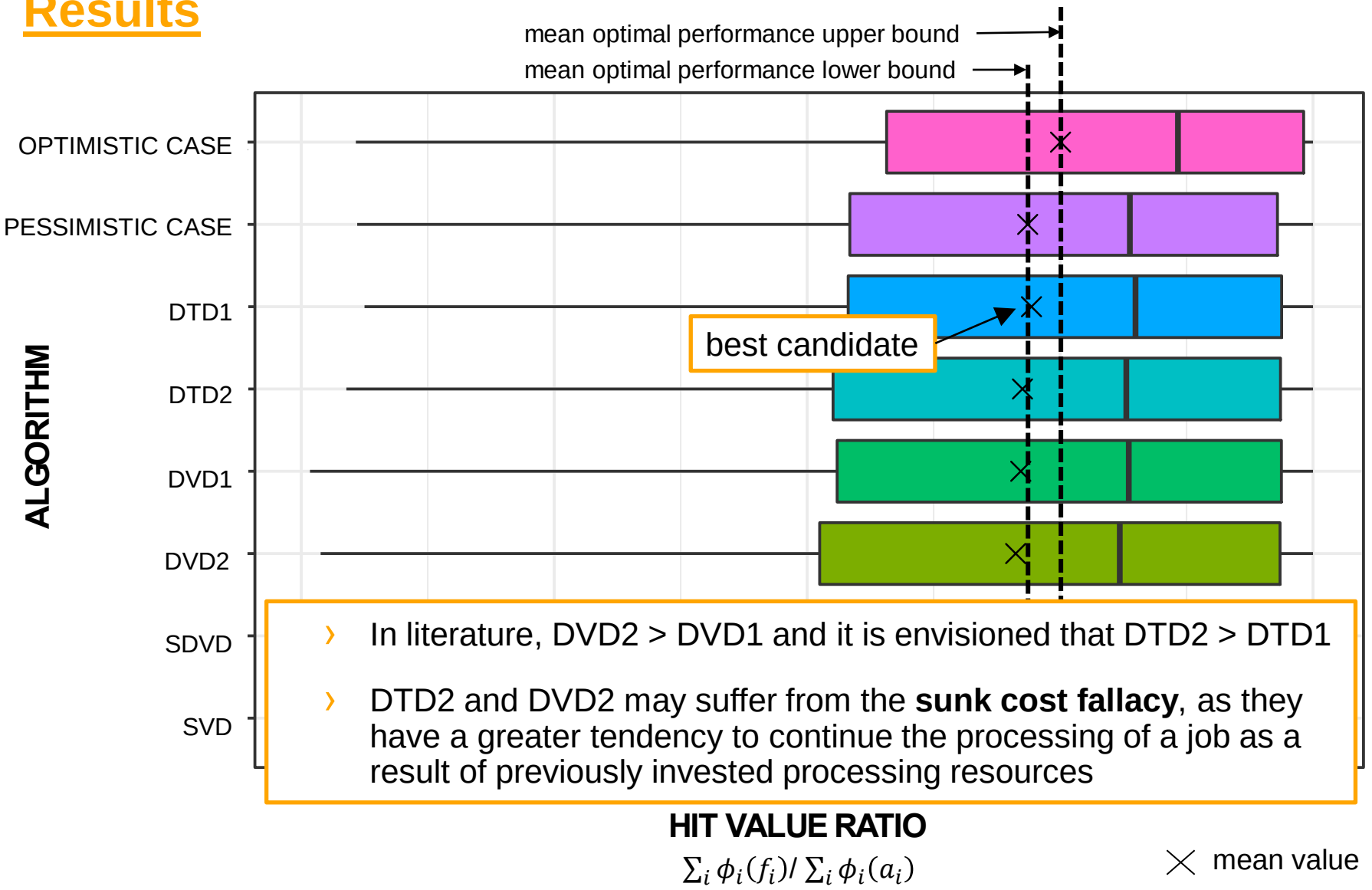
Results


mean optimal performance upper bound
mean optimal performance lower bound
OPTIMISTIC CASE
PESSIMISTIC CASE
DTD1
DTD2
DVD1
DVD2
SDVD
SVD
ALGORITHM
0%
25%
50%
75%
100%
HIT VALUE RATIO
$\sum_i \phi_i(f_i) / \sum_i \phi_i(a_i)$
mean value

Results


mean optimal performance upper bound
mean optimal performance lower bound
OPTIMISTIC CASE
PESSIMISTIC CASE
DTD1
DTD2
DVD1
DVD2
SDVD
SVD
best candidate
ALGORITHM
0%
25%
50%
75%
100%
HIT VALUE RATIO
$\sum_i \phi_i(f_i) / \sum_i \phi_i(a_i)$
mean value

Results



- In literature, DVD2 > DVD1 and it is envisioned that DTD2 > DTD1
- DTD2 and DVD2 may suffer from the **sunk cost fallacy**, as they have a greater tendency to continue the processing of a job as a result of previously invested processing resources

Conclusion & future work

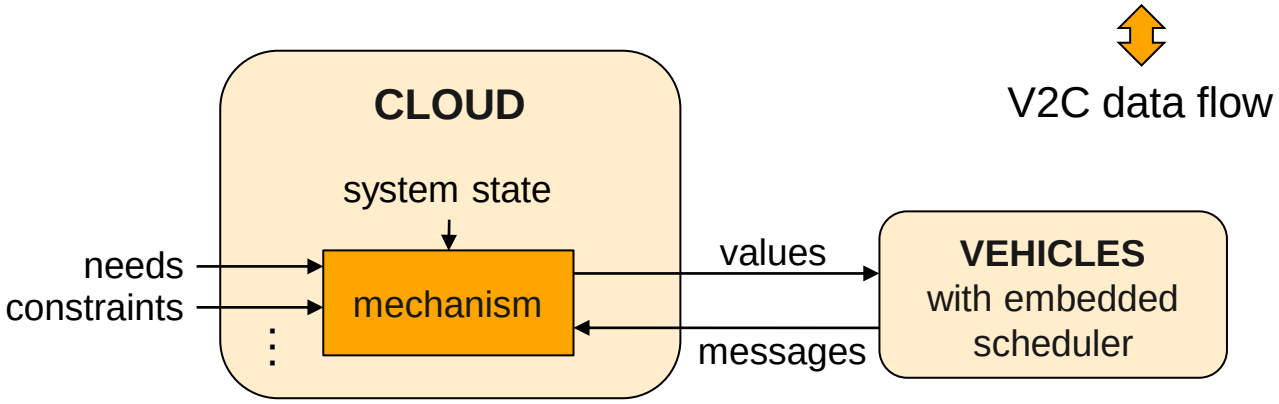
- approach: optimization local to each vehicle ➔ soft real-time scheduling problem
- extended analysis: generation of synthetic cases & comparison with baseline

Conclusion & future work

- approach: optimization local to each vehicle ➔ soft real-time scheduling problem
- extended analysis: generation of synthetic cases & comparison with baseline

Future work

- messages values influence V2C data flow
- design a mechanism in the Cloud to modulate messages values to react to system state



THANK YOU !

Feel free to contact me: jibarz@laas.fr

Optimizing Vehicle-to-Cloud Data Transfers using Soft Real-Time Scheduling Concepts

Jean IBARZ [1,2] Olivier Flébus [1]

Matthieu ROY [2], Michaël LAUER [2], Jean-Charles FABRE [2]

[1] Continental Digital Services, Toulouse, France

[2] CNRS-LAAS, Toulouse, France

Jean IBARZ © Continental AG