Optimal service selection policies for dynamic service composition

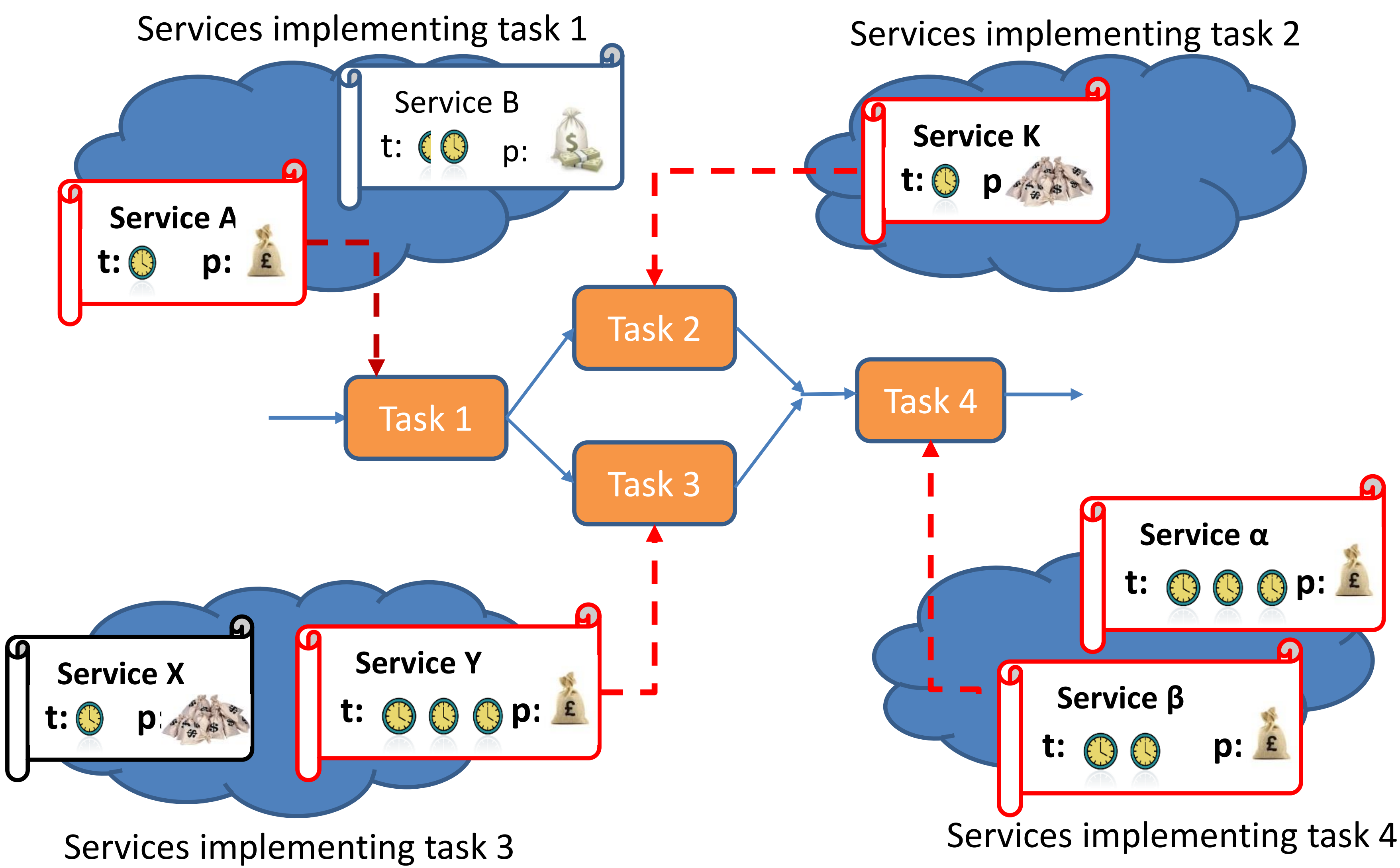
Miroslav Živković
University of Amsterdam (UvA)
System and Network Engineering Group
m.zivkovic@uva.nl

Joost Bosman, Hans van den Berg, Rob van der Mei,
Erik Meeuwissen, Rudesindo Núñez-Queija

- M. Živković, J. Bosman, H. van den Berg, R. van der Mei, H. Meeuwissen, R. Núñez-Queija: Run-time Revenue Maximization for Composite Web Services with Response Time Commitments (AINA 2012)
- M. Živković, H. van den Berg: Revenue Optimization of Service Compositions using Conditional Request Retries (IJWSR 2013)

Service Composition problem

Given abstract workflow select concrete services to execute it



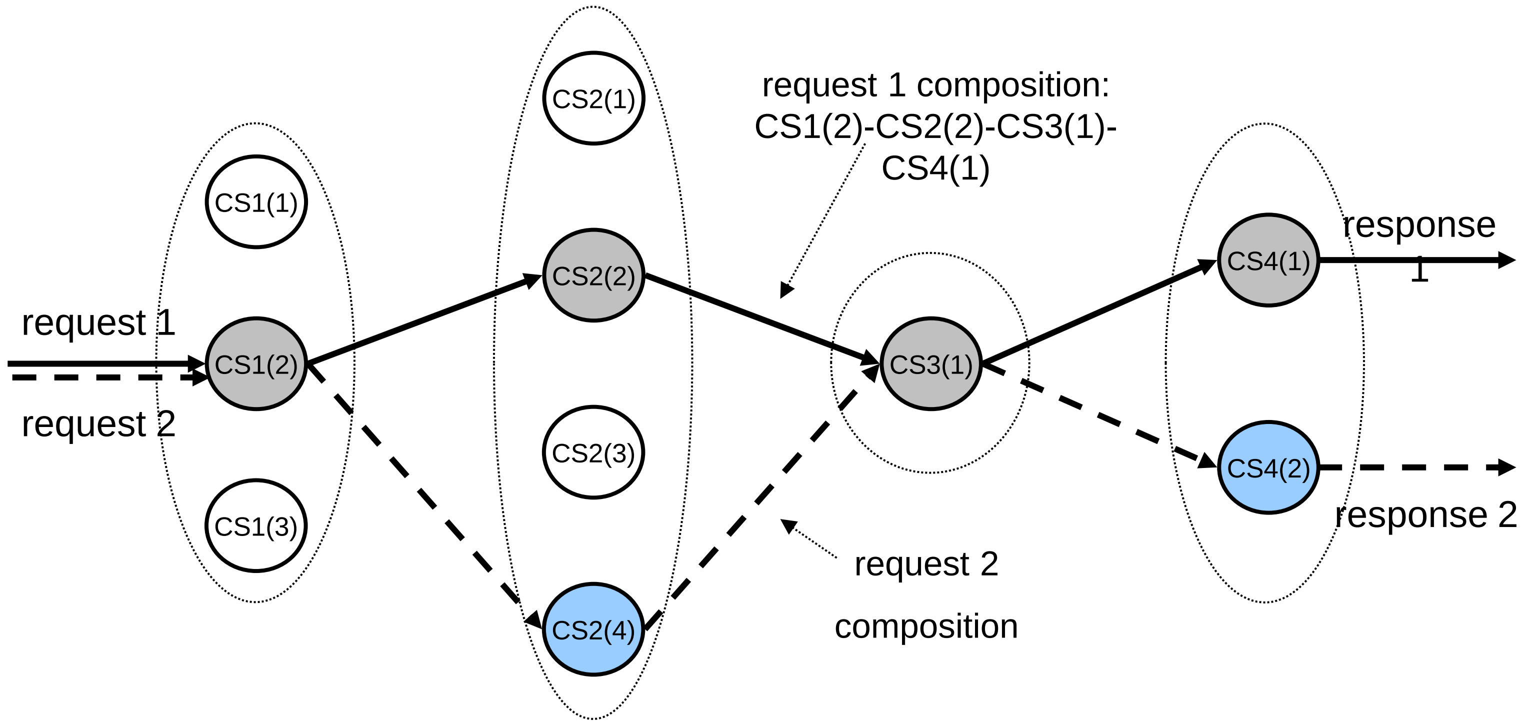
Service composition

- We focus on orchestration
- At design time (multi-objective problem)
  - Choices made upfront; non-dominated, Pareto-optimal solutions
  - Inflexible; impossible to modify on-the-fly
- At execution time:
  - Composition choices are made on-the-fly, flexible

What if performance worsens?

- **Runtime service selection**
- **Runtime service substitution**

Runtime service selection: Model



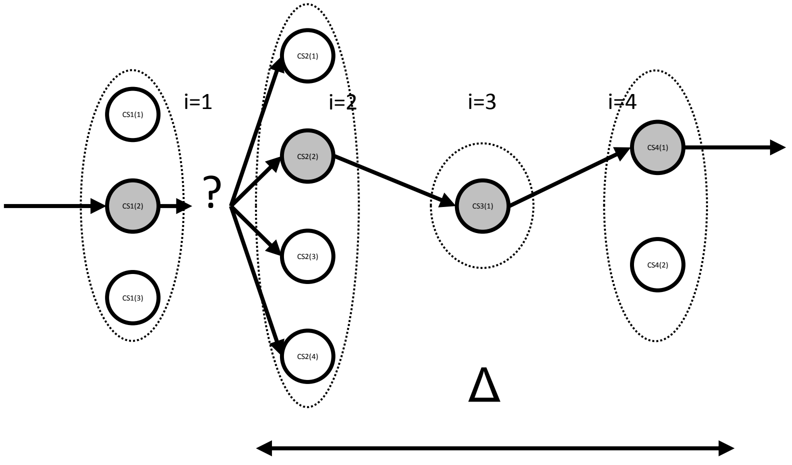
Runtime service selection

- Sequential workflow
- Composition may be adapted during execution (per request/task)
- Use of elapsed time info

The orchestrator

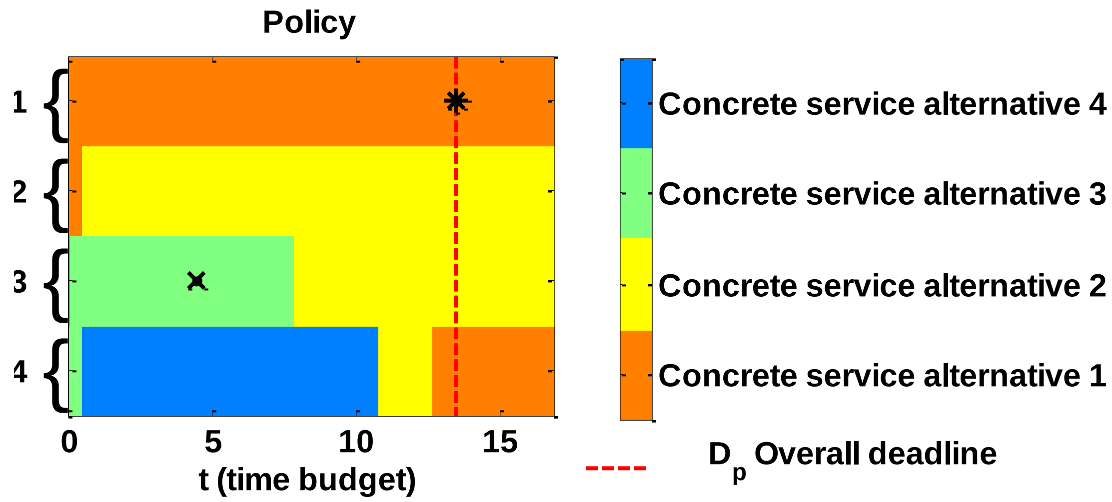
- knows the workflow
- selects appropriate services
- makes appropriate Service Level Agreements (SLAs) with 3rd party providers and its clients
- has no impact to or control of 3rd party domains
- End-to-end SLA
  - response time deadline (δ_p)
  - Reward when response time $\leq \delta_p$, penalty otherwise
- Single service SLA
  - Execution cost, response time distribution

Optimized dynamic decisions



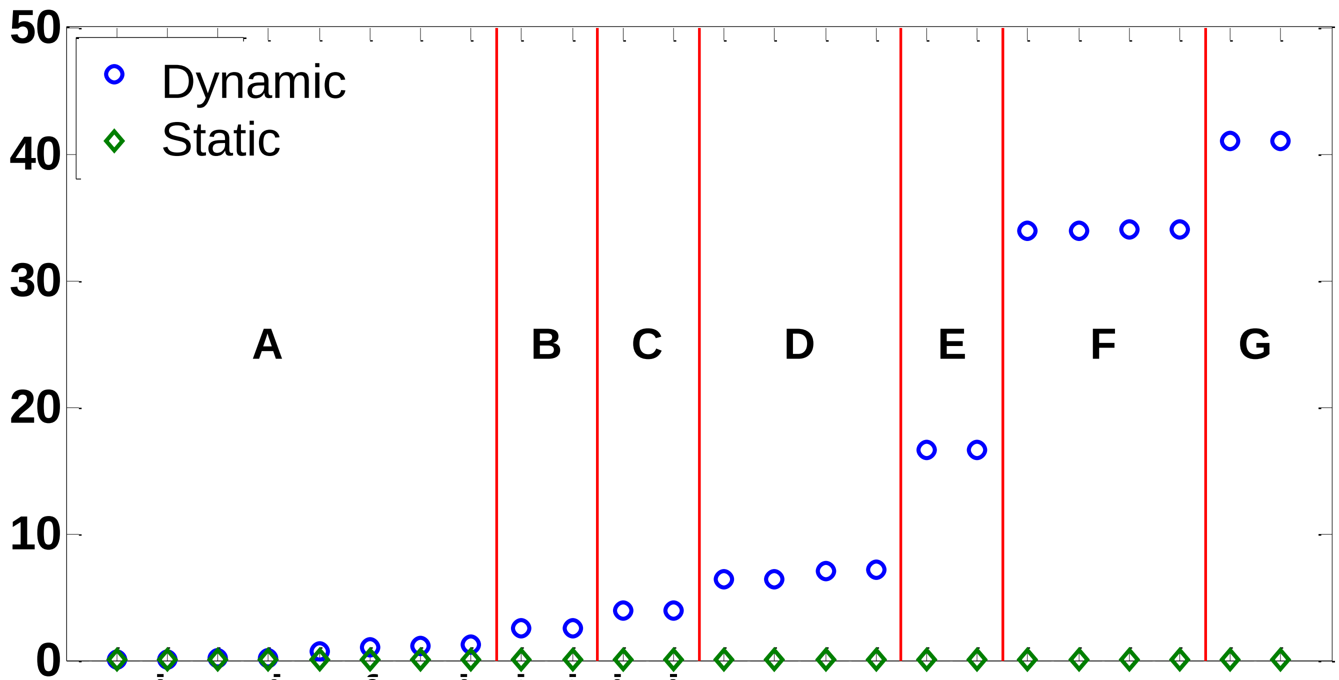
- Given
  - Position in workflow
  - Remaining time until deadline
  - Response time distributions
  - Costs, reward and penalty
- Decision
  - what service alternative to select based on the elapsed time?
- Goal
  - Optimize expected revenue
- Solution Apply Dynamic Programming

Lookup Table



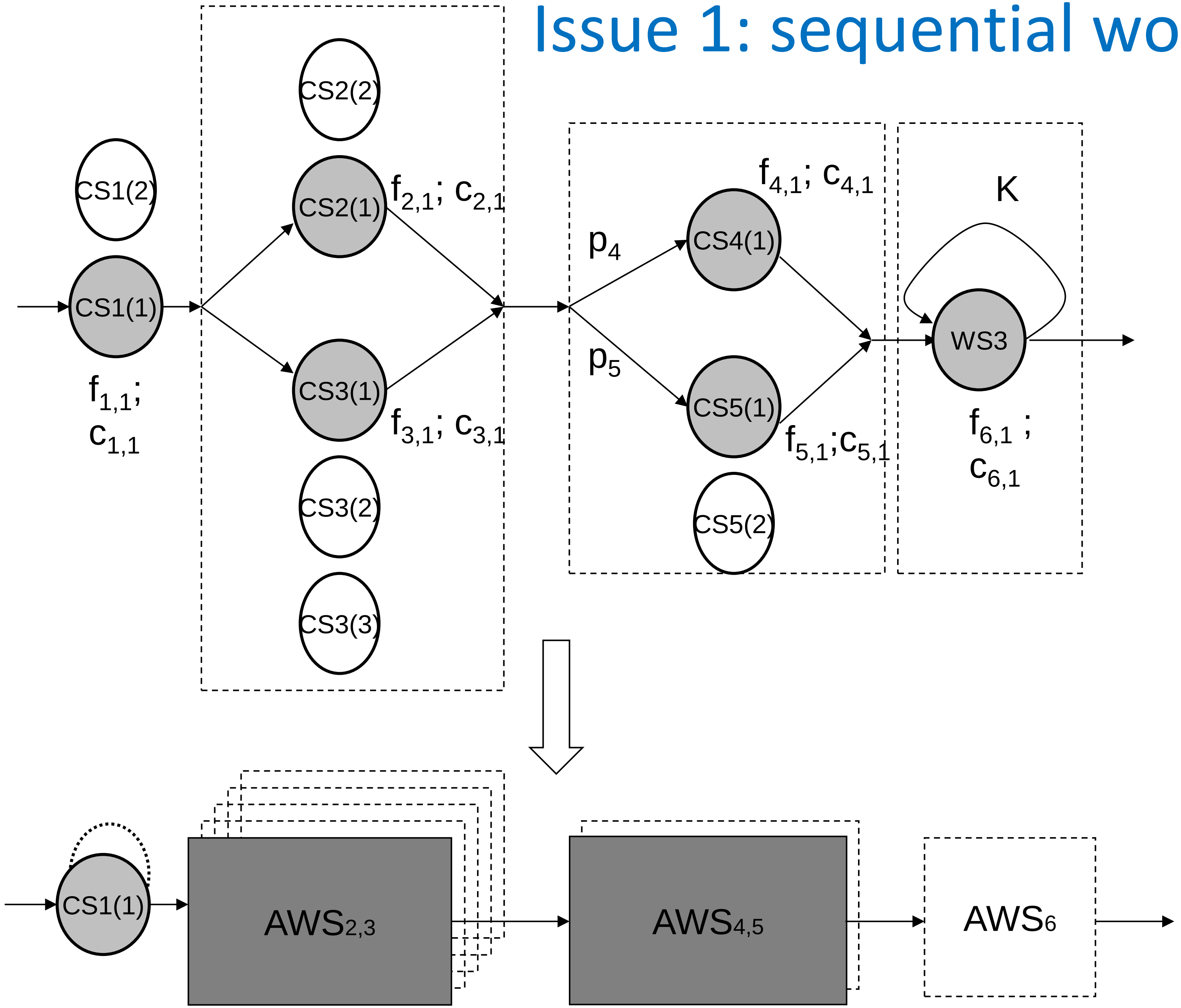
- <u>Simple</u> solution
- Calculate lookup table <u>off-line</u>
- Apply lookup table <u>on-line</u> (no computing)

Example workflow, 4 tasks

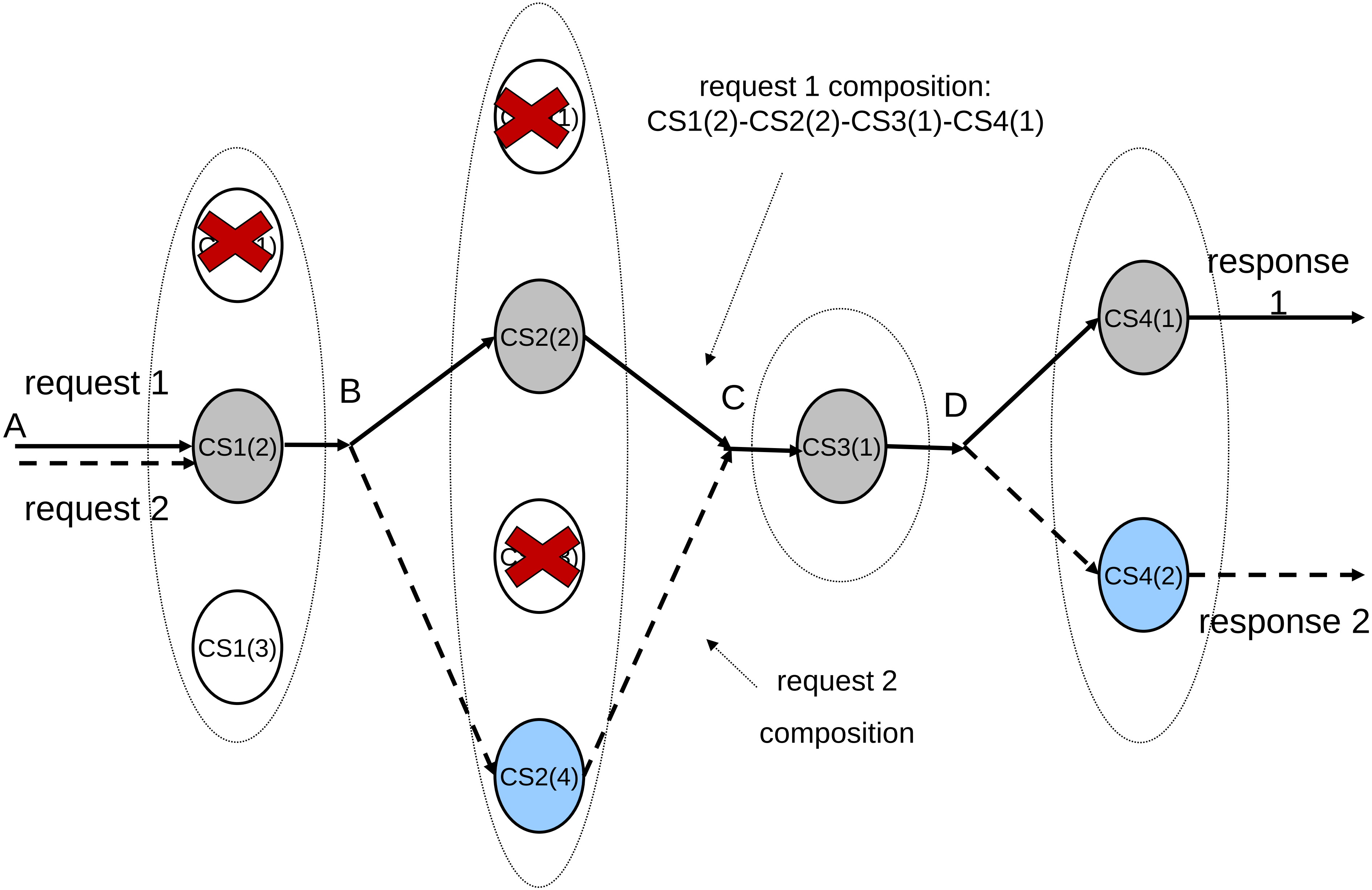


	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
Position 1	4	4	4	4	2	1	3	3	2	3	3	1	4	4	2	1	2	1	2	3	3	1	2	1
Position 2	2	3	3	1	4	4	4	4	3	2	1	3	2	1	4	4	1	2	3	2	1	3	1	2
Position 3	3	2	1	3	3	3	2	1	4	4	4	4	1	2	1	2	4	4	1	1	2	2	3	3
Position 4	1	1	2	2	1	2	1	2	1	1	2	2	3	3	3	3	3	3	4	4	4	4	4	4

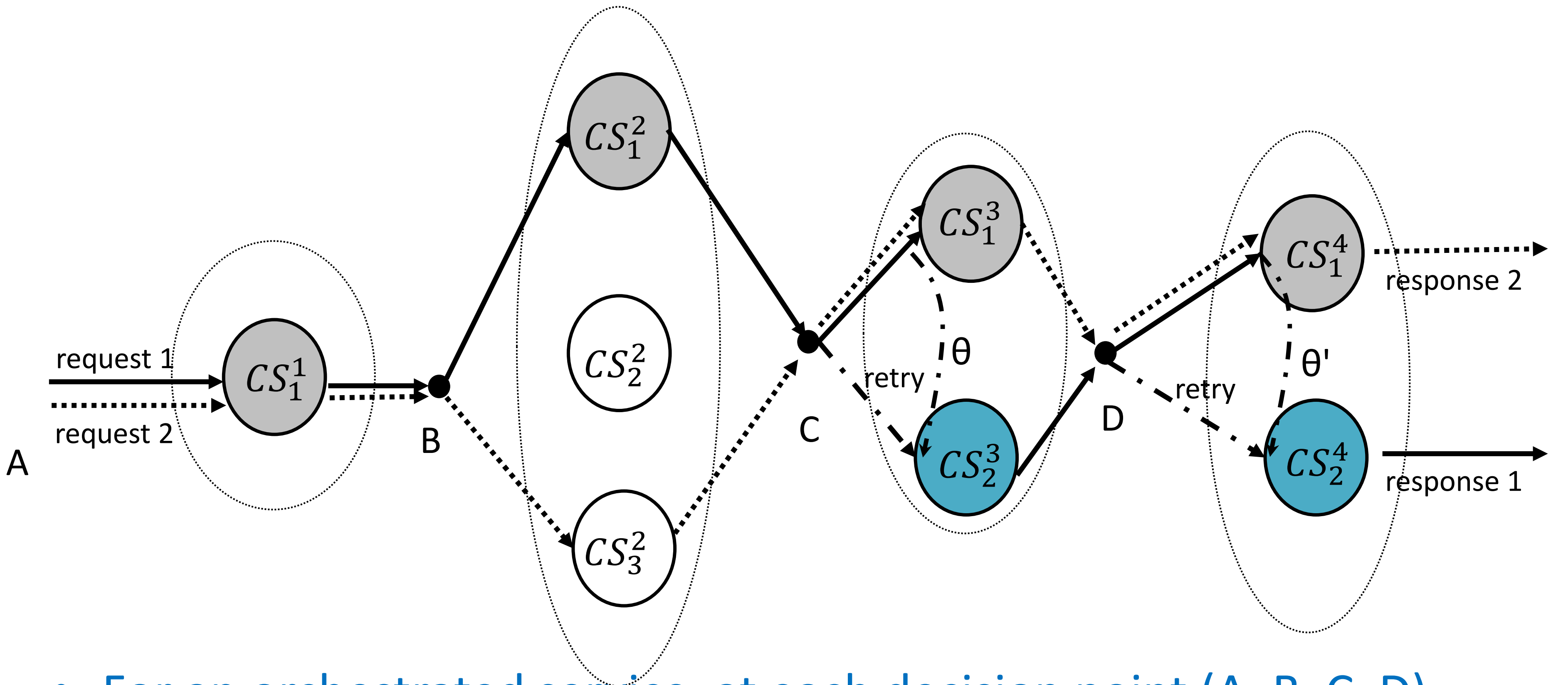
Issue 1: sequential workflow



Issue 2 - availability



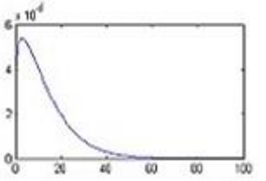
Runtime service substitution: Model



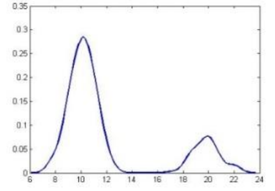
- For an orchestrated service, at each decision point (A, B, C, D)
  - a) which service should be selected?
  - b) when is it optimal to perform substitution
  - c) which service should be selected for a retry, same or some other?
- with the goal to maximize end-to-end expected revenue for given deadline

Response-time: when does a substitution make sense?

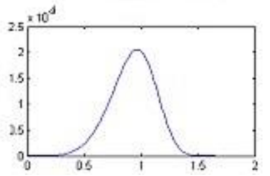
- Heavy-tailed response time distributions



  - <u>Expectation paradox</u>: “the longer we have waited, the longer we should expect to wait”



- Bimodal/Multimodal distributions



- Substitute by any given service

- This <u>does not involve</u> the <u>costs</u>

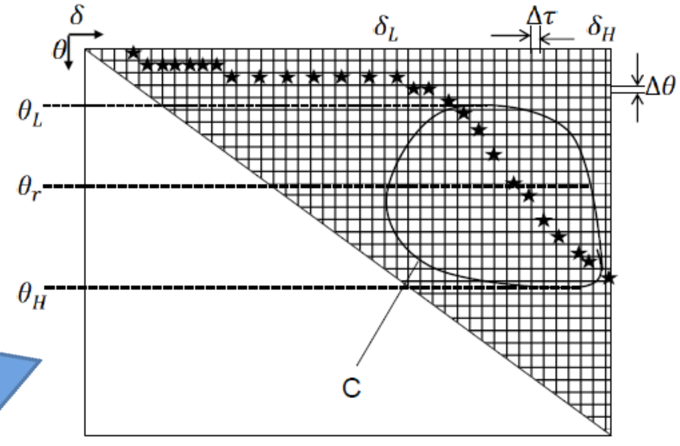
Optimal solution

- Use <u>dynamic programming</u> to calculate the <u>policy:</u>
  - Compare the expected revenues with and without retry
  - Formulae for case when <u>single retry</u> per <u>each task</u> is allowed
  - Task i, service j, deadline δ, retry moment θ, response time distribution *f, F,* revenue *W*

$$\mathbb{E}[W_j^i(\delta)] = -c_j^i + \int_0^{\delta} f_j^i(\tau) W^{(i+1)*}(\delta-\tau)\,d\tau + (1-F_j^i(\delta))\cdot W^{(i+1)*}(0).$$

$$\mathbb{E}[W_{j\to k}^i(\delta,\theta_{j\to k}^i)] = \underbrace{-c_j^i}_{\text{term 1}} + \underbrace{\int_0^{\theta_{j\to k}^i} f_j^i(\tau) W^{(i+1)*}(\theta_{j\to k}^i-\tau)\,d\tau}_{\text{term 2}}$$

$$+\underbrace{(1-F_j^i(\theta_{j\to k}^i))\cdot \mathbb{E}[W_k^i(\delta-\theta_{j\to k}^i)]}_{\text{term 3}}.$$



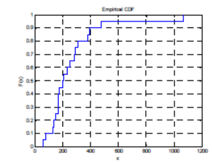
- term 1: execution cost; term 2 – no retry needed; term 3 – retry made

Runtime service substitution - conclusions

- For larger values of deadlines, policies with or without retries are close
- Perform substitution for the last tasks
  - Cost plays a role: the more you pay the less substitutions you should perform

Issue 3 – time invariant PDF: Closed loop control

- For each alternative response time distribution: keep last n samples
- Calculate empirical distribution(s)
- Apply DP on empirical distributions



<u>Challenges</u>

1. We do <u>not</u> prefer updating the policy after <u>each</u> realization
2. When a certain alternative is never selected we don't observe changes

<u>Solutions</u>

1. Apply <u>statistical test</u> to see weather an empirical distribution has changed significantly
2. If a service is not used for (specified) time interval send a probe request (and pay corresponding cost);
   - Tradeoff: short interval (good + expensive) vs. large interval

Thank you!