Journey planning in uncertain environments, the multi-objective way

Mickael Randour

UMONS - Université de Mons & F.R.S.-FNRS, Belgium

January 15, 2019

Think tank "Systèmes complexes"

Aim of this talk

Flavor of $\neq$ types of **useful strategies** in stochastic environments.

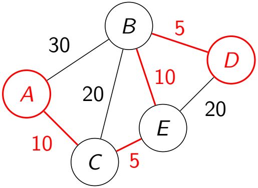
- ▷ Loosely based on [RRS15] (on arXiv: abs/1411.0835).

Aim of this talk

Flavor of $\neq$ types of **useful strategies** in stochastic environments.

▷ Loosely based on [RRS15] (on arXiv: abs/1411.0835).

Applications to the **shortest path problem**.



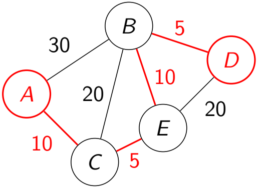
↪ Find a **path of minimal length** in a weighted graph (Dijkstra, Bellman-Ford, etc) [CGR96].

Aim of this talk

Flavor of ≠ types of **useful strategies** in stochastic environments.

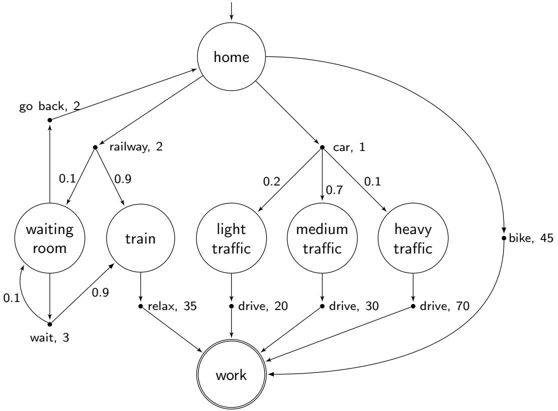
- ▷ Loosely based on [RRS15] (on arXiv: abs/1411.0835).

Applications to the **shortest path problem**.



What if the environment is **uncertain**? E.g., in case of heavy traffic, some roads may be crowded.

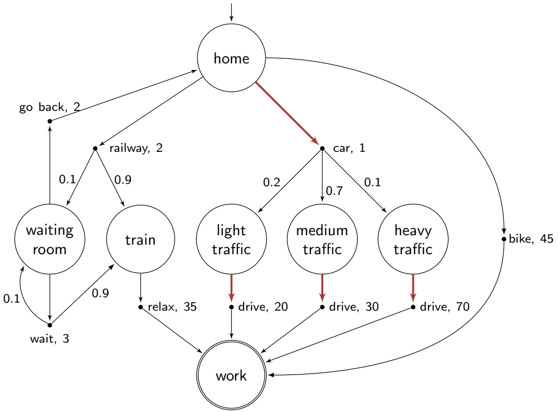
Planning a journey in an uncertain environment



Each action takes time, target = work.

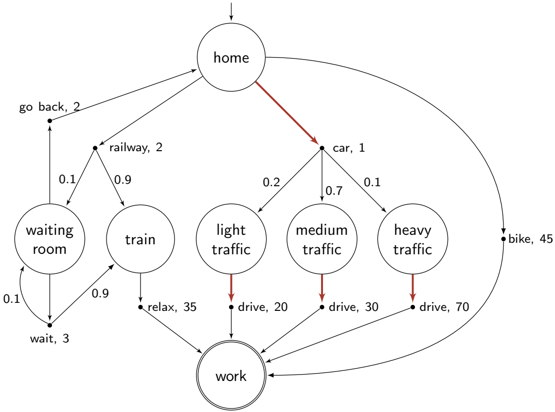
- What kind of strategies are we looking for when the environment is **stochastic** (Markov decision process)?

Solution 1: minimize the *expected* time to work



- "Average" performance: meaningful when you journey often.
- **Simple strategies** suffice: no memory, no randomness.
- Taking the **car** is optimal: $\mathbb{E}_D^{\sigma}(\mathsf{TS}^{\mathsf{work}}) = 33$.

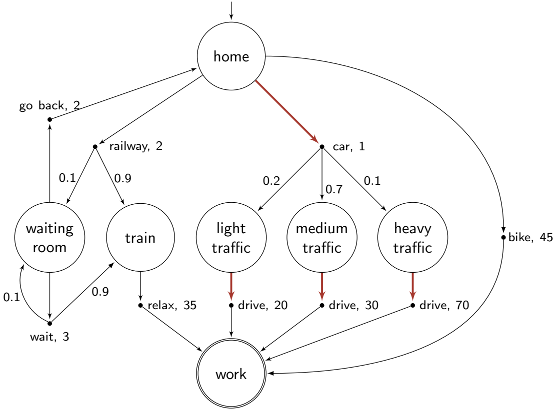
Solution 2: traveling without taking too many risks



Minimizing the *expected time* to destination makes sense **if** we travel often and **it is not a problem to be late**.

With car, in 10% of the cases, the journey takes 71 minutes.

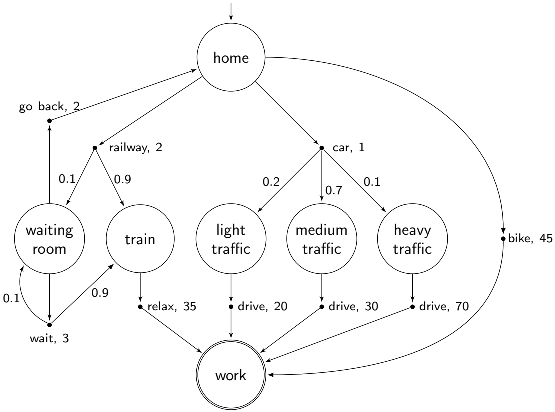
Solution 2: traveling without taking too many risks



Most bosses will not be happy if we are late too often...

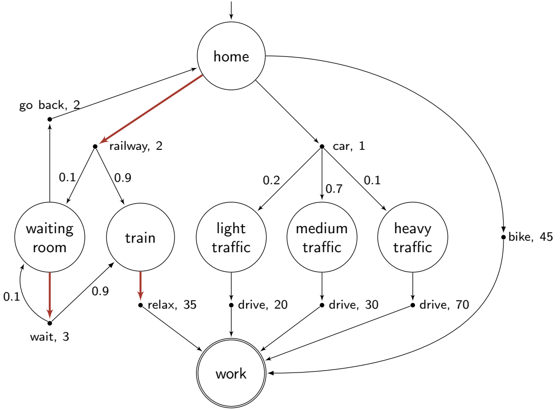
$\rightsquigarrow$ what if we are risk-averse and want to avoid that?

Solution 2: maximize the *probability* to be on time



Specification: reach work within 40 minutes with 0.95 probability

Solution 2: maximize the *probability* to be on time

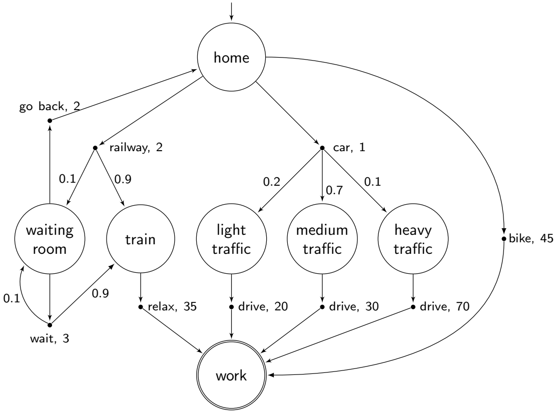


Specification: reach work within 40 minutes with 0.95 probability

Sample strategy: take the **train** $\rightsquigarrow \mathbb{P}^{\sigma}_{D}\left[\mathsf{TS}^{\mathsf{work}} \leq 40\right] = 0.99$

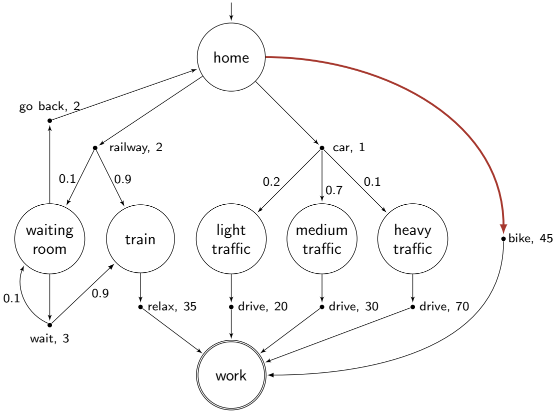
Bad choices: car (0.9) and bike (0.0)

Solution 3: strict worst-case guarantees



Specification: *guarantee* that work is reached within 60 minutes (to avoid missing an important meeting)

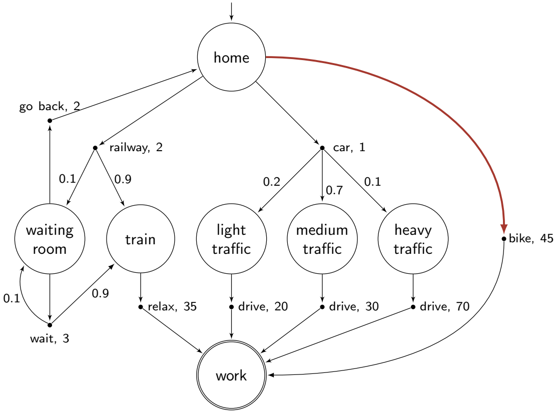
Solution 3: strict worst-case guarantees



Specification: *guarantee* that work is reached within 60 minutes (to avoid missing an important meeting)

Sample strategy: **bike** $\rightsquigarrow$ worst-case reaching time $= 45$ minutes.
Bad choices: train ($wc = \infty$) and car ($wc = 71$)

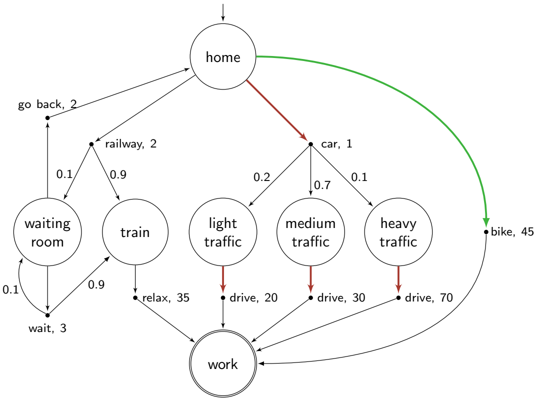
Solution 3: strict worst-case guarantees



Worst-case analysis ⇝ **two-player game** against an antagonistic adversary (*bad guy*)

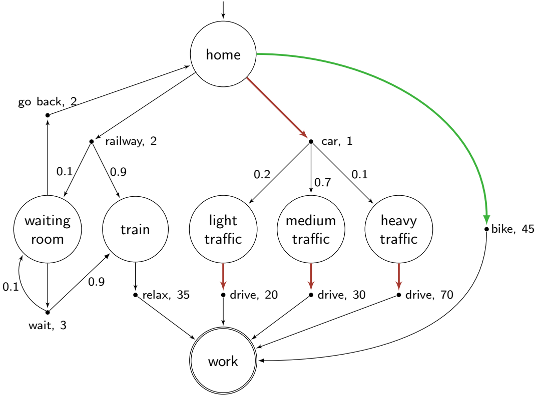
- ▷ forget about probabilities and give the choice of transitions to the adversary

Solution 4: minimize the *expected* time under strict worst-case guarantees



- Expected time: car $\rightsquigarrow \mathbb{E} = 33$ but $wc = 71 > 60$
- Worst-case: bike $\rightsquigarrow wc = 45 < 60$ but $\mathbb{E} = 45 >>> 33$

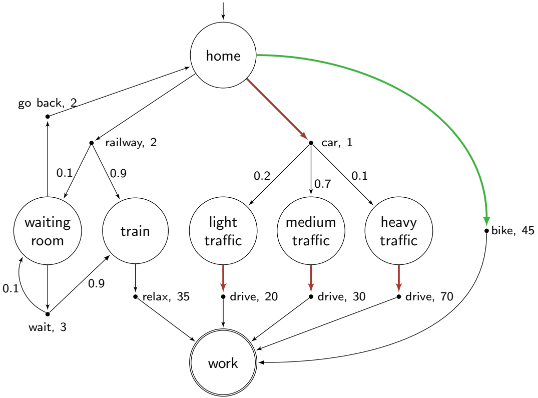
Solution 4: minimize the *expected* time under strict worst-case guarantees



In practice, we want both! Can we do better?

▷ **Beyond worst-case synthesis** [BFRR17]: minimize the expected time under the worst-case constraint.

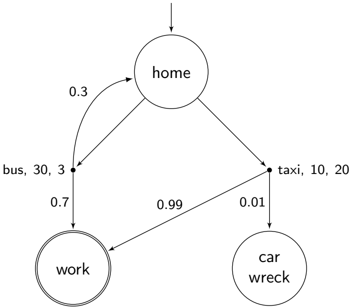
Solution 4: minimize the *expected* time under strict worst-case guarantees



Sample strategy: try train up to 3 delays then switch to bike.

- $\rightsquigarrow$ $wc = 58 < 60$ and $\mathbb{E} \approx 37.34 << 45$
- $\rightsquigarrow$ Strategies need **memory** $\rightsquigarrow$ more complex!

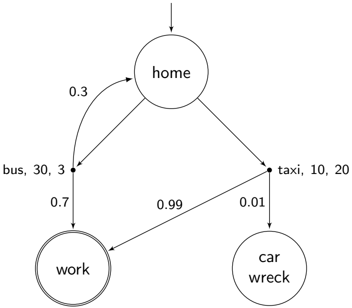
Solution 5: multiple objectives ⇒ trade-offs



Two-dimensional weights on actions: *time* and *cost*.

Often necessary to consider trade-offs: e.g., between the probability to reach work in due time and the risks of an expensive journey.

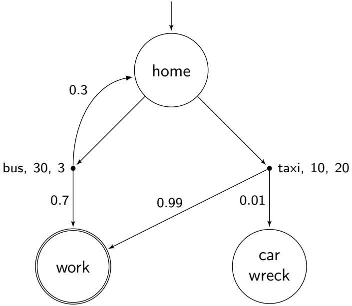
Solution 5: multiple objectives $\Rightarrow$ trade-offs



Solution 2 (probability) can only ensure a **single constraint**.

- **C1**: 80% of runs reach work in at most 40 minutes.
  - Taxi $\leadsto \leq 10$ minutes with probability $0.99 > 0.8$.

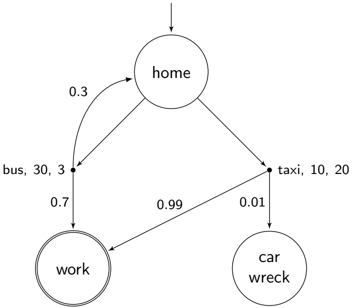
Solution 5: multiple objectives ⇒ trade-offs



Solution 2 (probability) can only ensure a **single constraint**.

- **C1**: 80% of runs reach work in at most 40 minutes.
  - ▷ Taxi $\rightsquigarrow \leq 10$ minutes with probability $0.99 > 0.8$.
- **C2**: 50% of them cost at most 10$ to reach work.
  - ▷ Bus $\rightsquigarrow \geq 70\%$ of the runs reach work for 3$.

Solution 5: multiple objectives ⇒ trade-offs

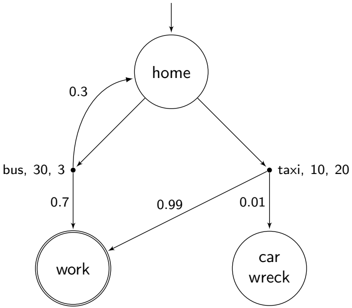


Solution 2 (probability) can only ensure a **single constraint**.

- **C1**: 80% of runs reach work in at most 40 minutes.
  - ▷ Taxi $\rightsquigarrow \leq 10$ minutes with probability $0.99 > 0.8$.
- **C2**: 50% of them cost at most 10$ to reach work.
  - ▷ Bus $\rightsquigarrow \geq 70\%$ of the runs reach work for 3$.

Taxi $\not\models$ C2, bus $\not\models$ C1. What if we want C1 $\wedge$ C2?

Solution 5: multiple objectives ⇒ trade-offs

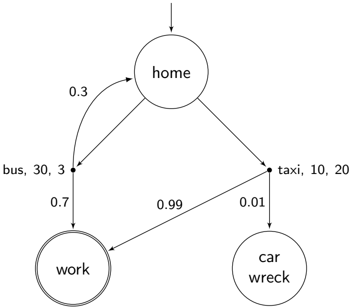


- **C1**: 80% of runs reach work in at most 40 minutes.
- **C2**: 50% of them cost at most 10$ to reach work.

Study of **multi-constraint percentile queries** [RRS17].

- ▷ Sample strategy: bus once, then taxi. Requires *memory*.
- ▷ Another strategy: bus with probability 3/5, taxi with probability 2/5. Requires *randomness*.

Solution 5: multiple objectives ⇒ trade-offs



- **C1**: 80% of runs reach work in at most 40 minutes.
- **C2**: 50% of them cost at most 10$ to reach work.

Study of **multi-constraint percentile queries** [RRS17].

In general, *both* memory *and* randomness are required.

$\neq$ previous problems $\rightsquigarrow$ more complex!

Conclusion

Our research aims at:

- defining meaningful *strategy concepts* and *objectives*,,
- providing *algorithms* and *tools* to compute those strategies,
- classifying the *complexity* of the different problems from a theoretical standpoint.
  - $\hookrightarrow$ Is it mathematically possible to obtain efficient algorithms?

Conclusion

Our research aims at:

- defining meaningful *strategy concepts* and *objectives*,,
- providing *algorithms* and *tools* to compute those strategies,
- classifying the *complexity* of the different problems from a theoretical standpoint.
  - $\hookrightarrow$ Is it mathematically possible to obtain efficient algorithms?

Thank you! Any question?

References I

Véronique Bruyère, Emmanuel Filiot, Mickael Randour, and Jean-François Raskin.
Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games.
Inf. Comput., 254:259–295, 2017.

B.V. Cherkassky, A.V. Goldberg, and T. Radzik.
Shortest paths algorithms: Theory and experimental evaluation.
Math. programming, 73(2):129–174, 1996.

Mickael Randour, Jean-François Raskin, and Ocan Sankur.
Variations on the stochastic shortest path problem.
In Deepak D'Souza, Akash Lal, and Kim Guldstrand Larsen, editors, Verification, Model Checking, and Abstract Interpretation - 16th International Conference, VMCAI 2015, Mumbai, India, January 12-14, 2015. Proceedings, volume 8931 of Lecture Notes in Computer Science, pages 1–18. Springer, 2015.

Mickael Randour, Jean-François Raskin, and Ocan Sankur.
Percentile queries in multi-dimensional markov decision processes.
Formal Methods in System Design, 50(2-3):207–248, 2017.