Karlsruhe Institute of Technology

Shortest Feasible Paths with Charging Stops for Battery Electric Vehicles

1st MOVESMART Workshop · 15 October 2015, Bilbao

Moritz Baum, Julian Dibbelt, Andreas Gemsa, Dorothea Wagner, Tobias Zündorf

INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP

Route Planning for Electric Vehicles

Electric vehicles:

- Future means of transportation
- Run on regenerative energy sources

But:

- Restricted battery capacity
- Long recharging times

Therefore : Route planning applications have to consider:

- Energy consumption
- Charging stops
- . . . but we would still like to travel fast

Route Planning with Charging Stops

Task: Given some source s and target t in a road network

- Find the fastest route from s to t,
- such that battery does not deplete
- NP-hard problem (Constrained Shortest Path)

Institute of Theoretical Informatics
Algorithmics Group

Route Planning with Charging Stops

Task: Given some source s and target t in a road network

- Find the fastest route from s to t,
- such that battery does not deplete
- NP-hard problem (Constrained Shortest Path)

Challenges:

1. Recuperation
   - Vehicles recuperate energy (braking or going downhill)
   - Results in negative edge weights
   - Negative cycles are physically ruled out

Institute of Theoretical Informatics
Algorithmics Group

Route Planning with Charging Stops

Task: Given some source *s* and target *t* in a road network

- Find the fastest route from *s* to *t*,
- such that battery does not deplete
- NP-hard problem (Constrained Shortest Path)

Challenges:

1. Recuperation
   - Vehicles recuperate energy (braking or going downhill)
   - Results in negative edge weights
   - Negative cycles are physically ruled out
2. Battery Constraints
   - Battery must never be depleted (otherwise route is infeasible)
   - Battery cannot be overcharged (additional energy is lost)
   - Applies at every point of the route

Institute of Theoretical Informatics
Algorithmics Group

Route Planning with Charging Stops

Task: Given some source *s* and target *t* in a road network

- Find the fastest route from *s* to *t*,
- such that battery does not deplete
- NP-hard problem (Constrained Shortest Path)

Challenges:

3. Speed vs energy-efficiency vs. time spent at charging station
   - Fastest = optimize overall trip time (driving time + recharging time)
   - E. g., detour for more efficient route to skip charging station
   - E. g., detour for faster charging station (normal vs. swapping station)

Institute of Theoretical Informatics
Algorithmics Group

Route Planning with Charging Stops

Task: Given some source *s* and target *t* in a road network
- Find the fastest route from *s* to *t*,
- such that battery does not deplete
- NP-hard problem (Constrained Shortest Path)

Challenges:

3. Speed vs energy-efficiency vs. time spent at charging station
   - Fastest = optimize overall trip time (driving time + recharging time)
   - E. g., detour for more efficient route to skip charging station
   - E. g., detour for faster charging station (normal vs. swapping station)

4. Charging is not linear
   - Faster for lower battery SoC; do not charge fully if unnecessary
   - Short charging stops in order to reach faster charging stations

Institute of Theoretical Informatics
Algorithmics Group

Challenges

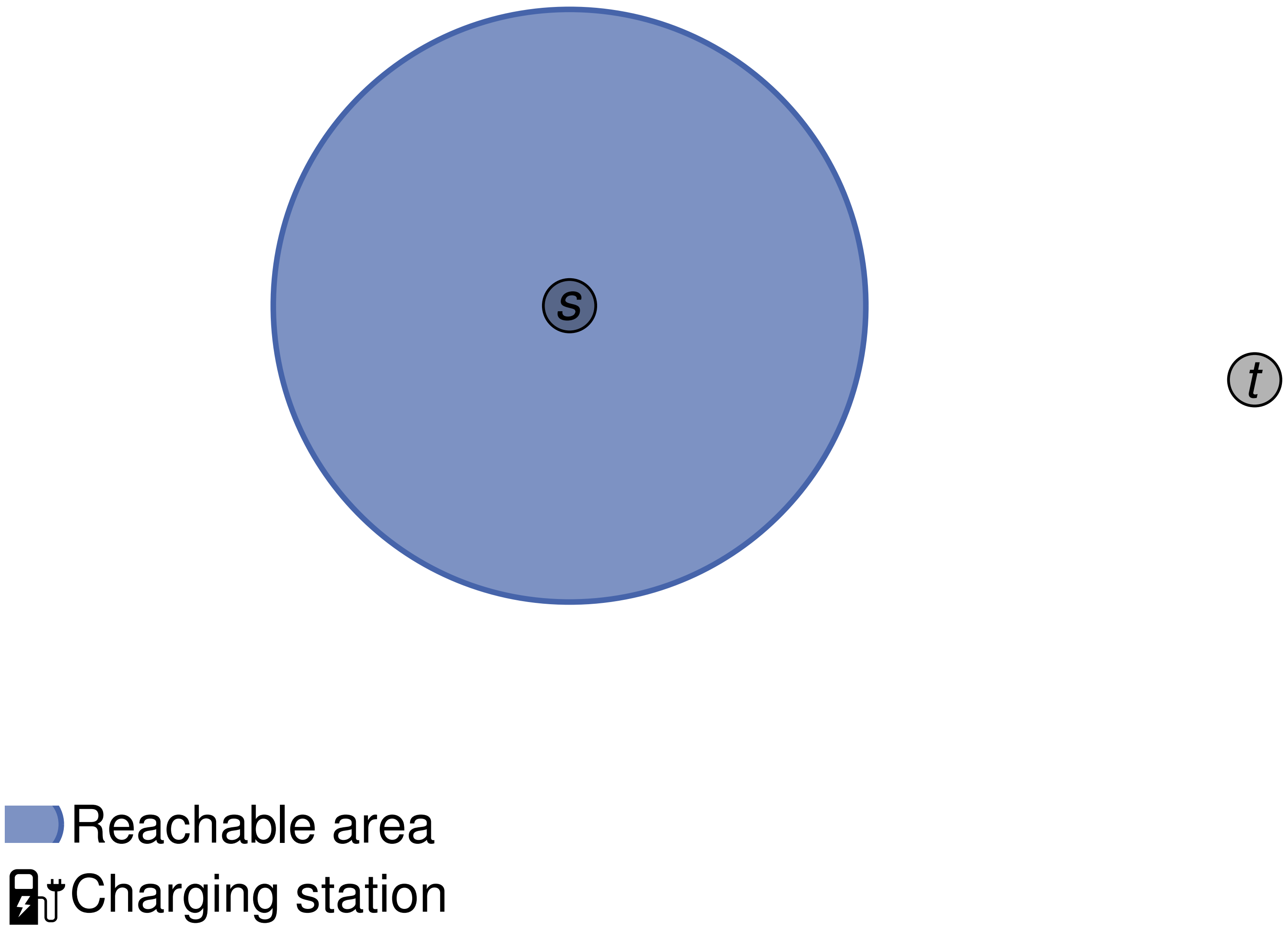
Find the fastest route from *s* to *t*:

Reachable area

Charging station

Challenges

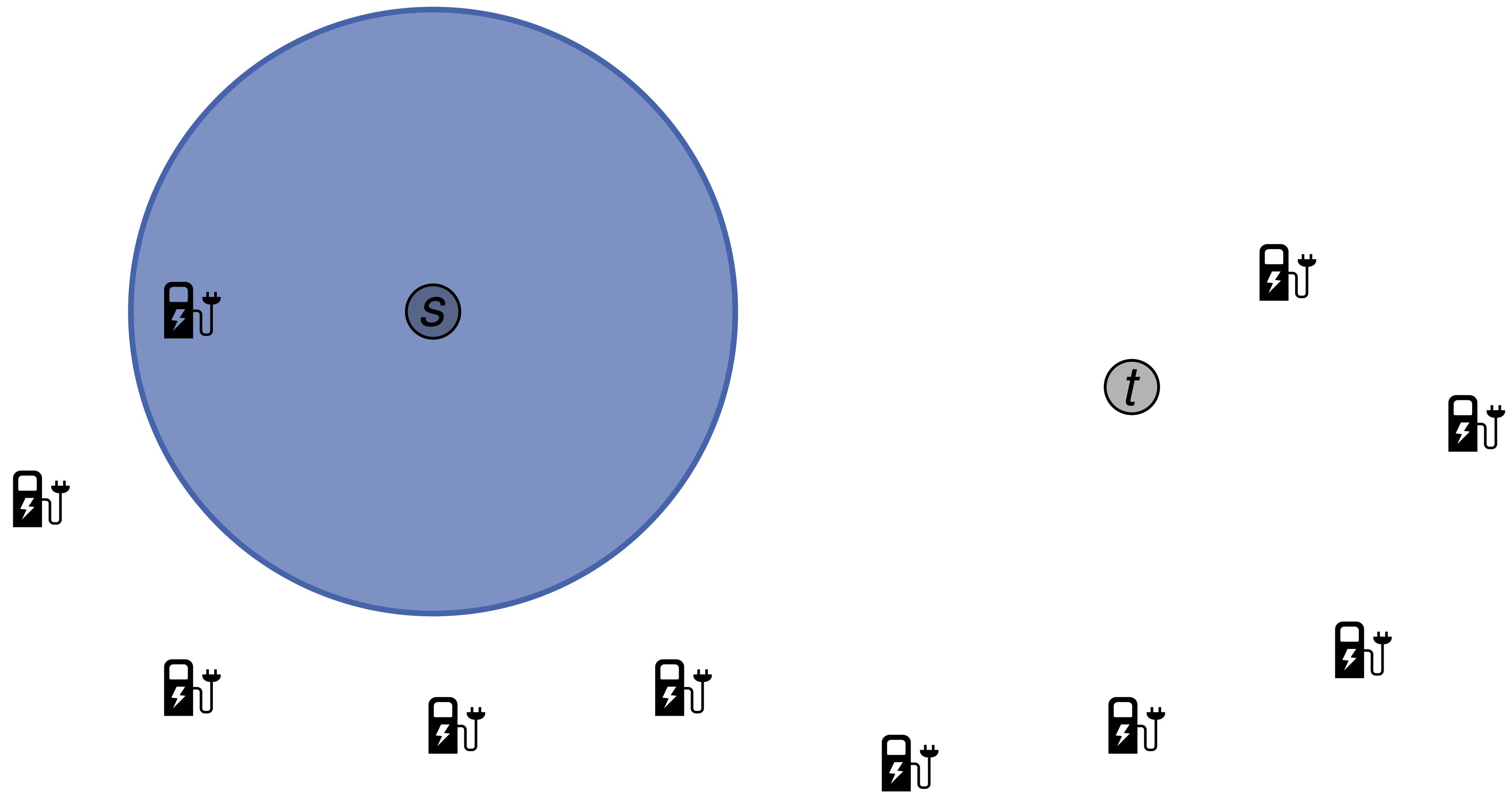
Find the fastest route from *s* to *t*:



Institute of Theoretical Informatics
Algorithmics Group

Challenges

Find the fastest route from *s* to *t*:



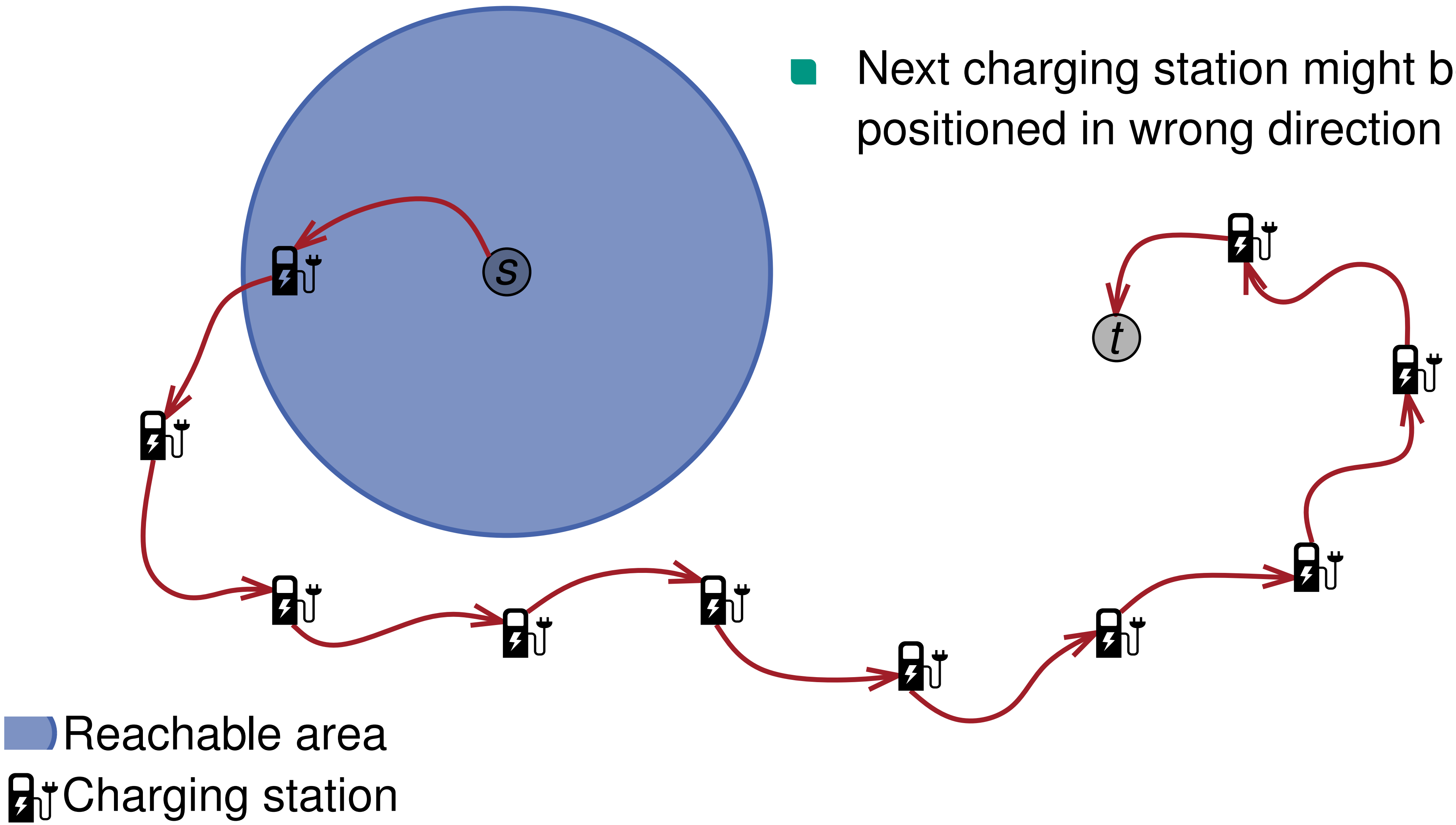
Reachable area

Charging station

Institute of Theoretical Informatics
Algorithmics Group

Challenges

Find the fastest route from s to t:

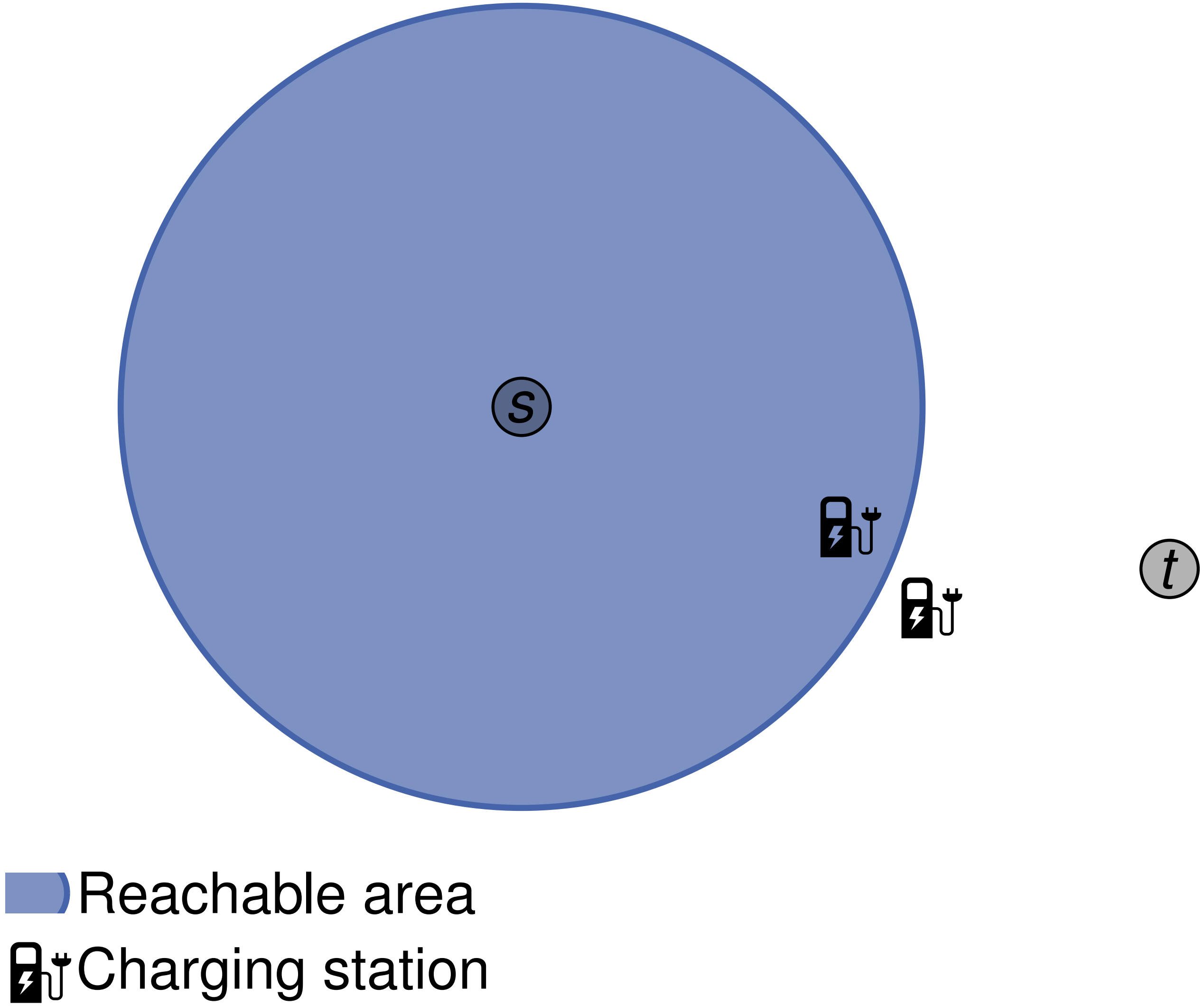


- Next charging station might be positioned in wrong direction

Institute of Theoretical Informatics
Algorithmics Group

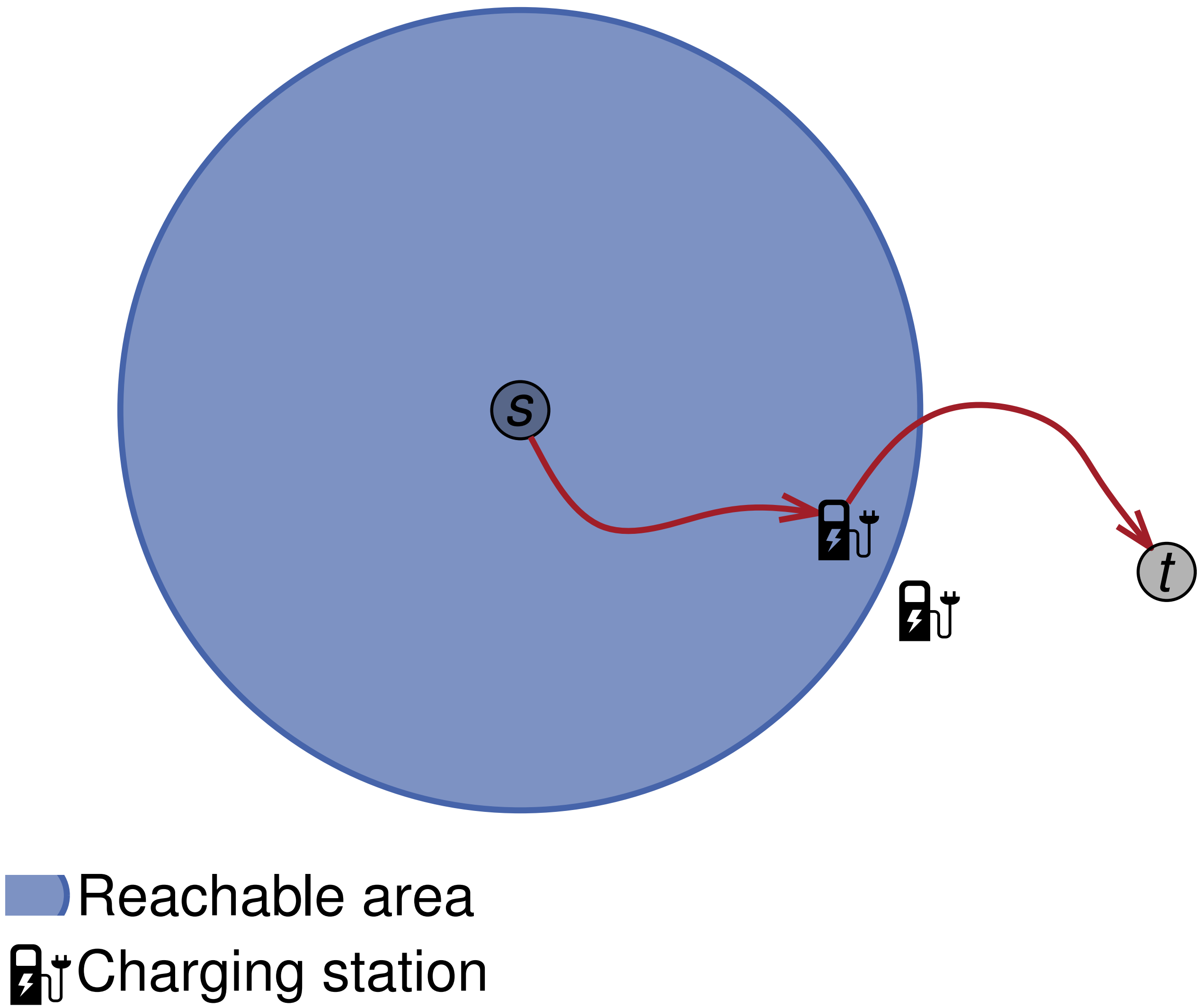
Challenges

Find the fastest route from s to t:



Challenges

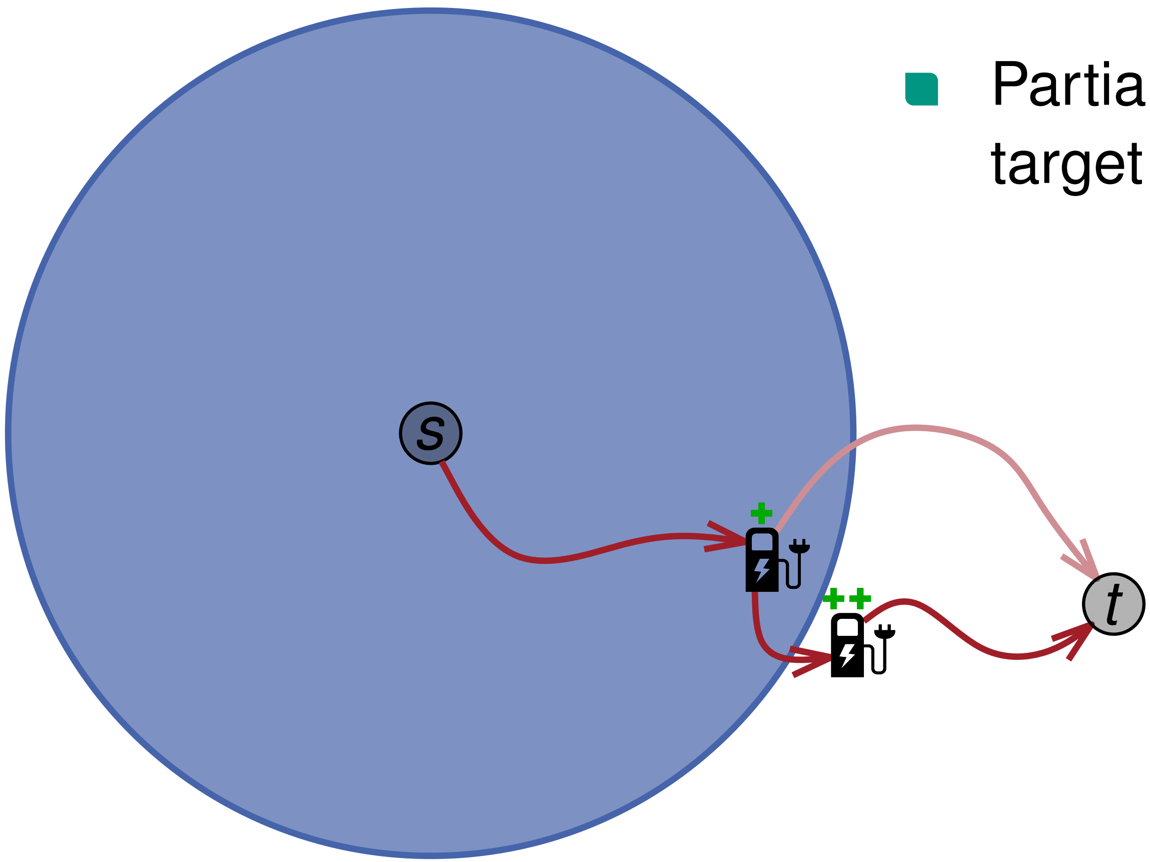
Find the fastest route from s to t:



Institute of Theoretical Informatics
Algorithmics Group

Challenges

Find the fastest route from *s* to *t*:



- Partial recharging, even if the target is already reachable

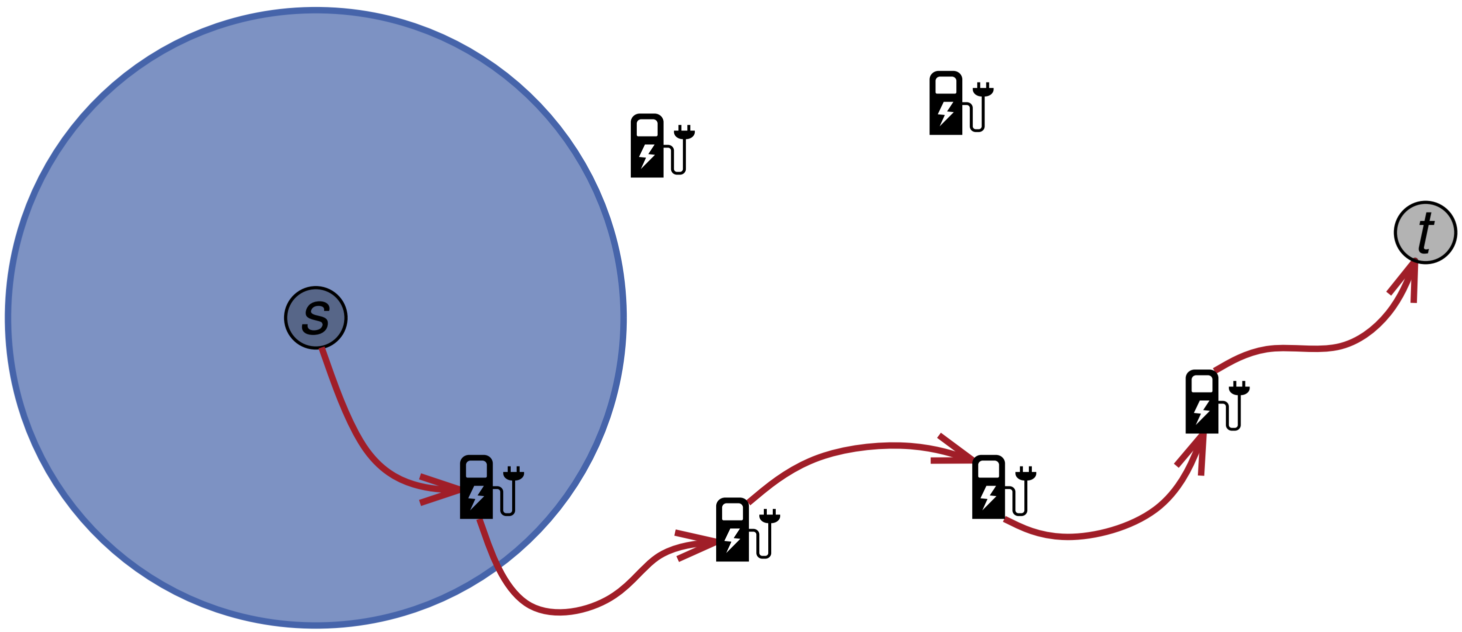
Reachable area

Charging station

Fast charging station / swapping station

Challenges

Find the fastest route from *s* to *t*:



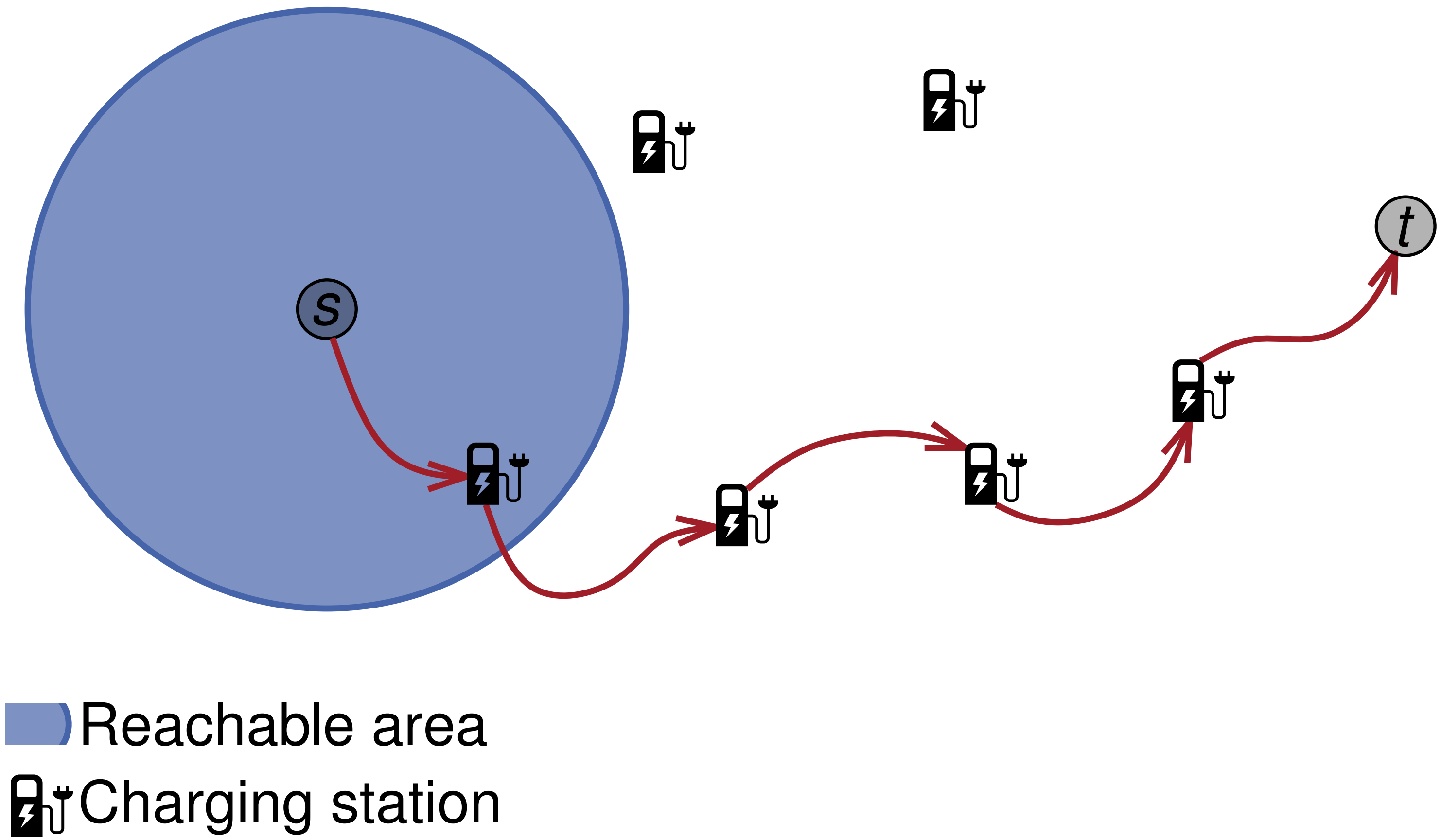
Reachable area
Charging station

Institute of Theoretical Informatics
Algorithmics Group

Challenges

Find the fastest route from *s* to *t*:

- Bigger battery $\Rightarrow$ simpler problem ?

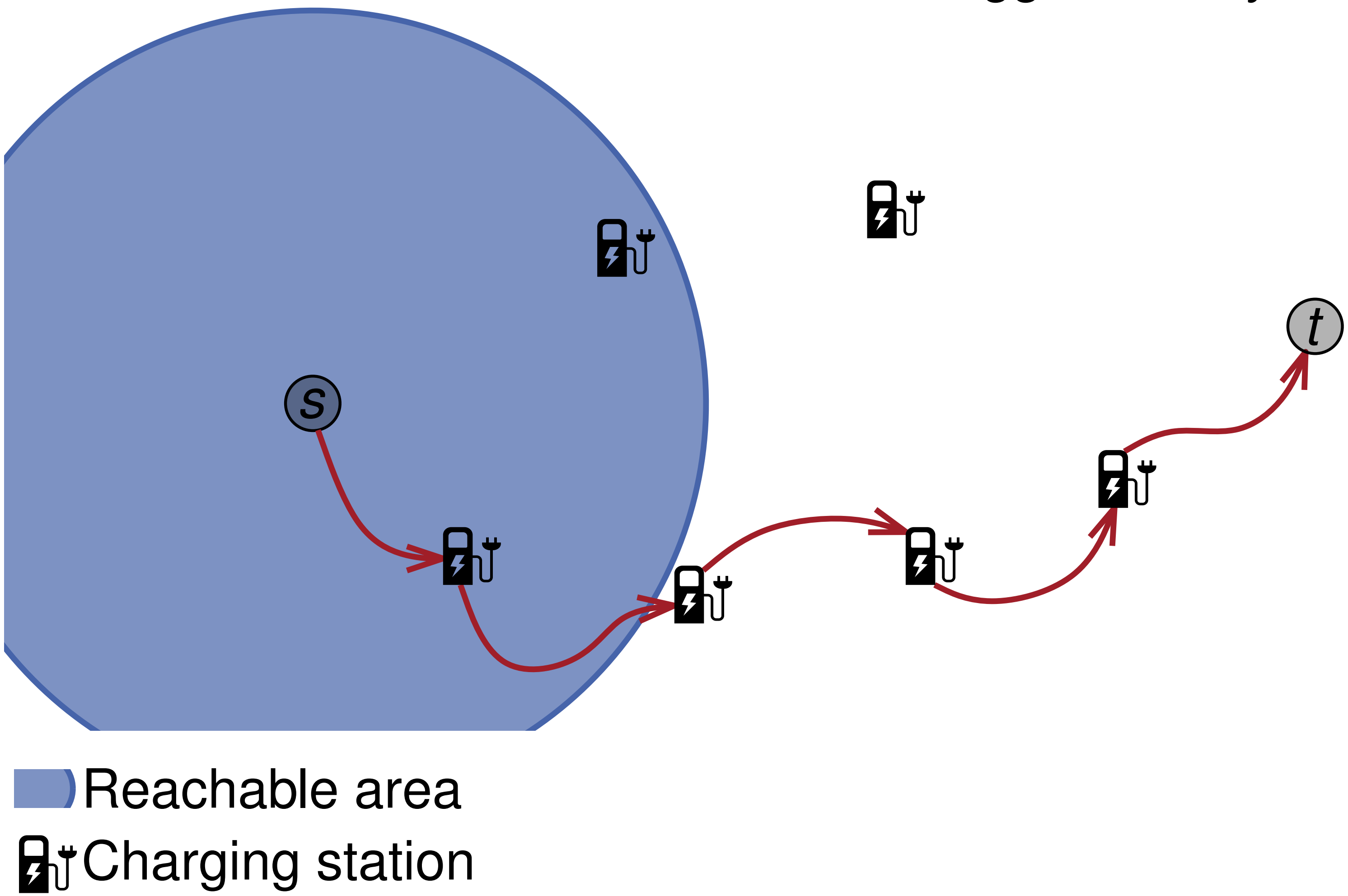


Institute of Theoretical Informatics
Algorithmics Group

Challenges

Find the fastest route from s to t:

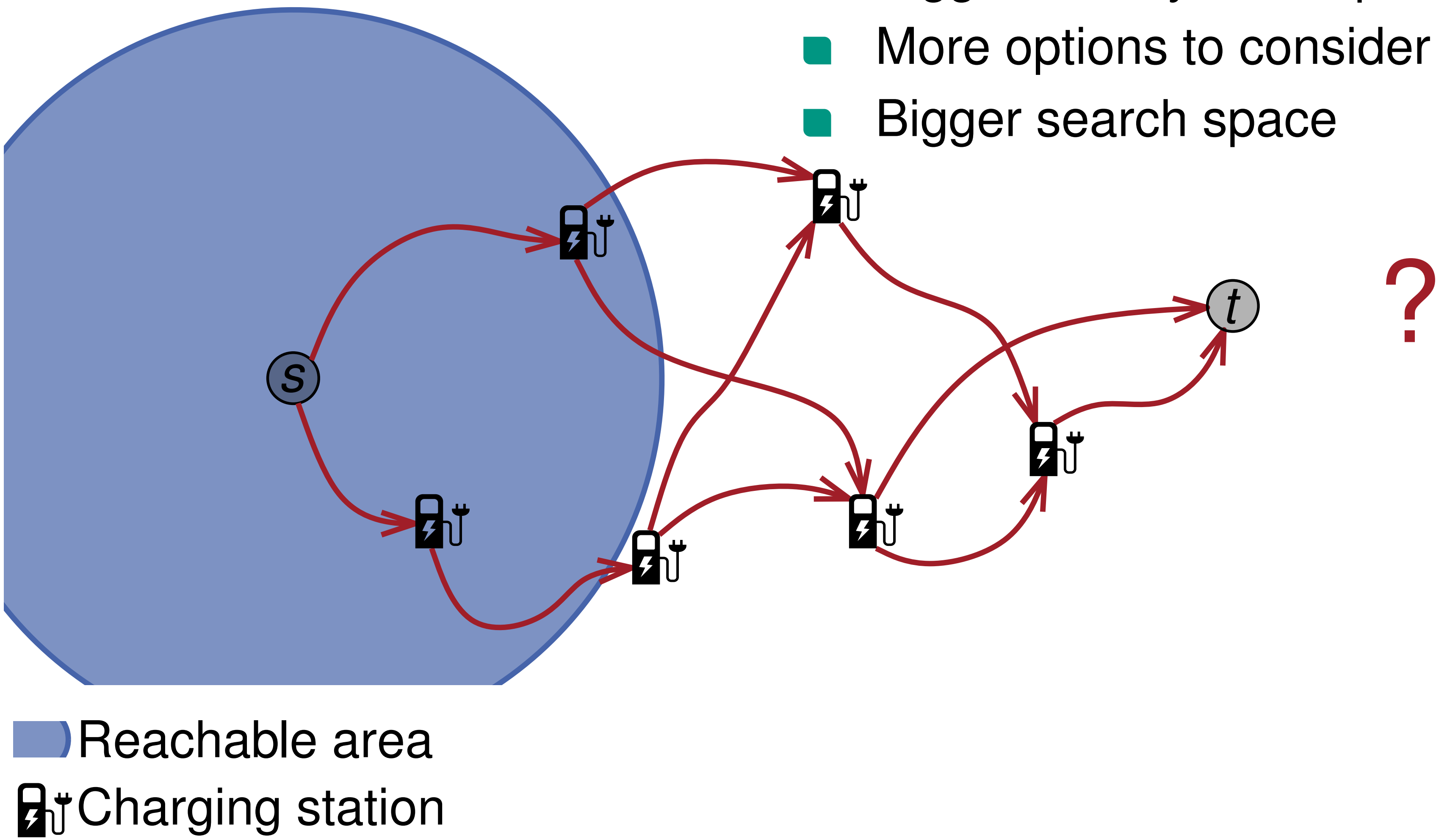
- Bigger battery $\Rightarrow$ simpler problem ?



Challenges

Find the fastest route from *s* to *t*:

- Bigger battery ⇒ simpler problem ?
- More options to consider
- Bigger search space



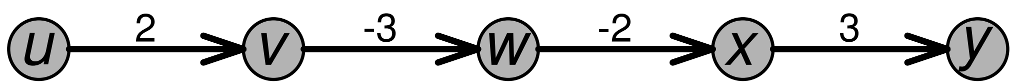
Related Work – Consumption Profiles

[Eisner et al. ’11], [Baum et al. ’13]

- Incorporate battery constraints in one single function
  - Maps State of Charge (SoC) onto energy consumption
  - Value ∞ indicates an infeasible SoC
- For a single edge or path:
  - Representable using maximal 3 values

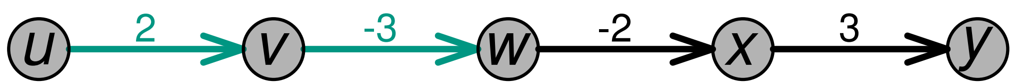
Related Work – Consumption Profiles
[Eisner et al. ’11], [Baum et al. ’13]

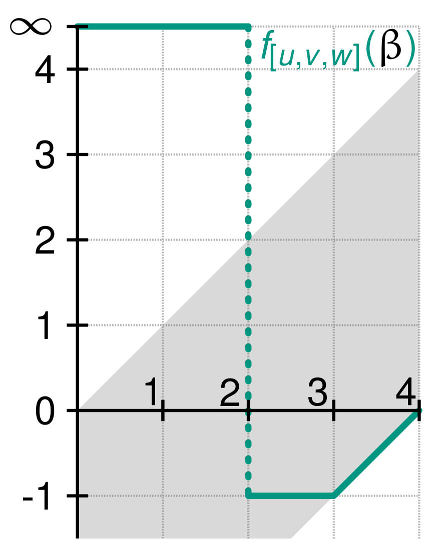
- Incorporate battery constraints in one single function
  - Maps State of Charge (SoC) onto energy consumption
  - Value ∞ indicates an infeasible SoC
- For a single edge or path:
  - Representable using maximal 3 values



Related Work – Consumption Profiles
[Eisner et al. ’11], [Baum et al. ’13]

- Incorporate battery constraints in one single function
  - Maps State of Charge (SoC) onto energy consumption
  - Value ∞ indicates an infeasible SoC
- For a single edge or path:
  - Representable using maximal 3 values

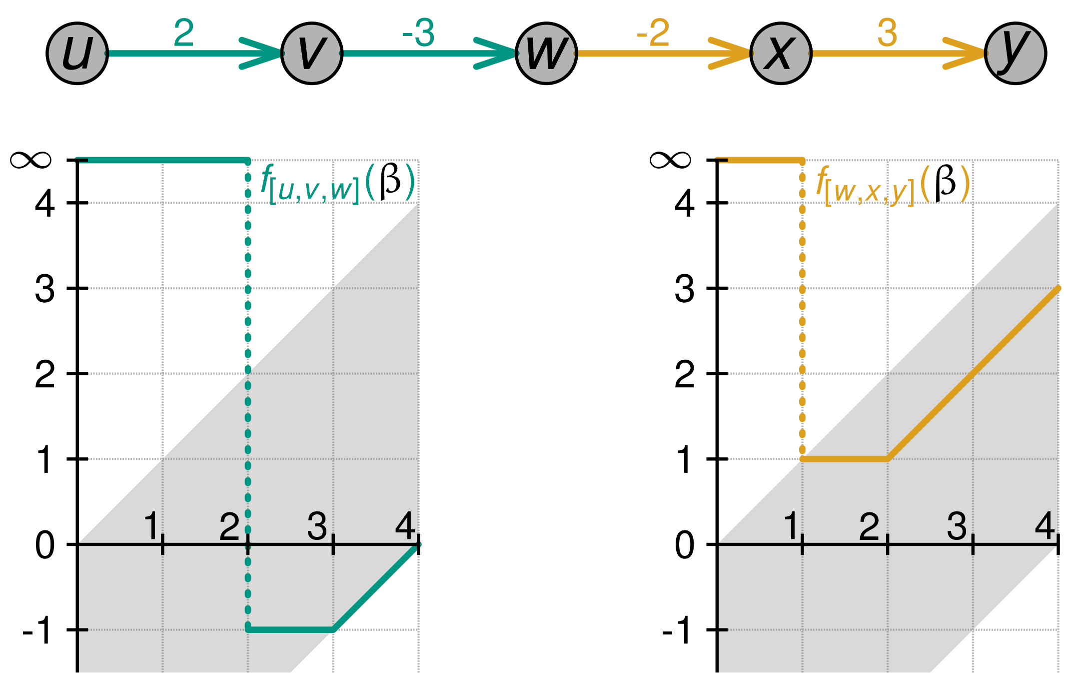




Institute of Theoretical Informatics
Algorithmics Group

Related Work – Consumption Profiles
[Eisner et al. ’11], [Baum et al. ’13]

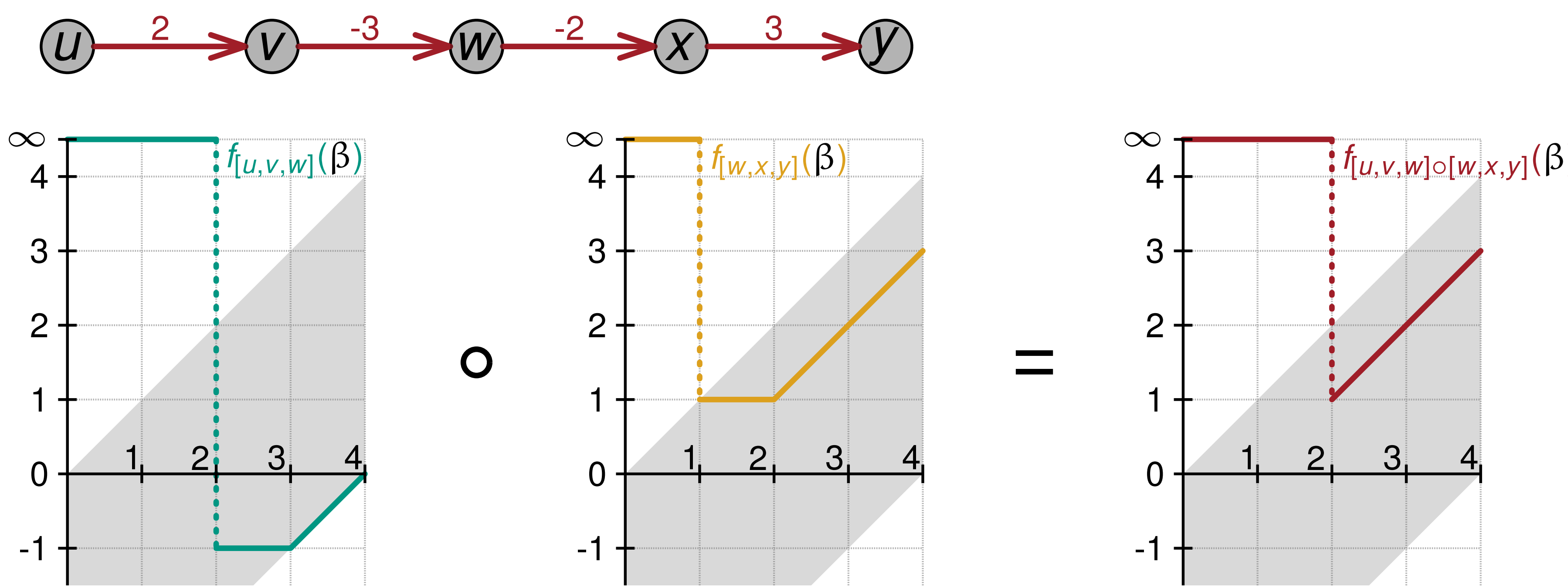
- Incorporate battery constraints in one single function
  - Maps State of Charge (SoC) onto energy consumption
  - Value ∞ indicates an infeasible SoC
- For a single edge or path:
  - Representable using maximal 3 values



Institute of Theoretical Informatics
Algorithmics Group

Related Work – Consumption Profiles
[Eisner et al. ’11], [Baum et al. ’13]

- Incorporate battery constraints in one single function
  - Maps State of Charge (SoC) onto energy consumption
  - Value ∞ indicates an infeasible SoC
- For a single edge or path:
  - Representable using maximal 3 values



Institute of Theoretical Informatics
Algorithmics Group

Related Work – Charging Stations

Quick and Energy-Efficient Routes: Computing Constrained Shortest Paths for Electric Vehicles [Storandt '12]

- Supports only battery swapping stations (BSS)
- Computes overlay graph of directly reachable BSS
- BSS fully recharges in constant time $\Rightarrow$ simple scalar overlay graph

$\Rightarrow$ Not applicable / sub-optimal in our scenario

Related Work – Charging Stations

Quick and Energy-Efficient Routes: Computing Constrained Shortest Paths for Electric Vehicles [Storandt ’12]

- Supports only battery swapping stations (BSS)
- Computes overlay graph of directly reachable BSS
- BSS fully recharges in constant time $\Rightarrow$ simple scalar overlay graph

$\Rightarrow$ Not applicable / sub-optimal in our scenario

Routing of Electric Vehicles: Constrained Shortest Path with Resource Recovering Nodes [Merting et al. ’15]

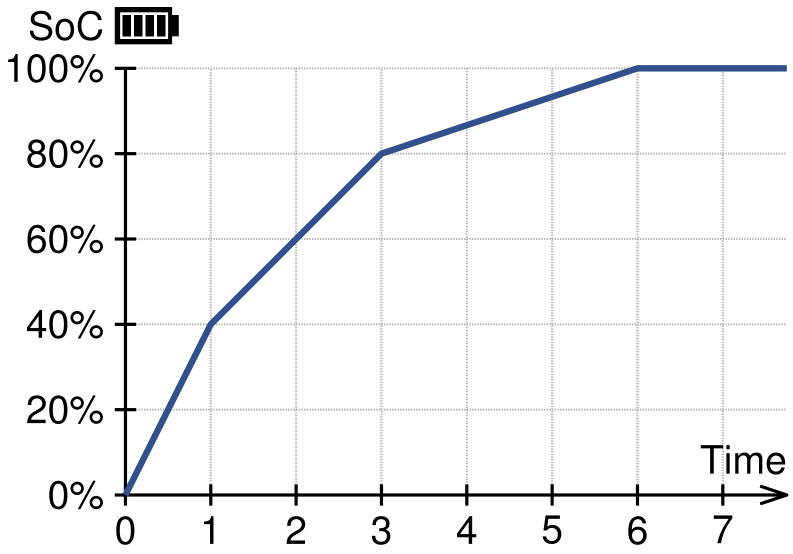
- Theoretical analysis of the problem
- Show missing sub-path property
- Optimal solutions can visit a charging station several times
- Developed a FPTAS

Institute of Theoretical Informatics
Algorithmics Group

Charging Functions

Charging station model:

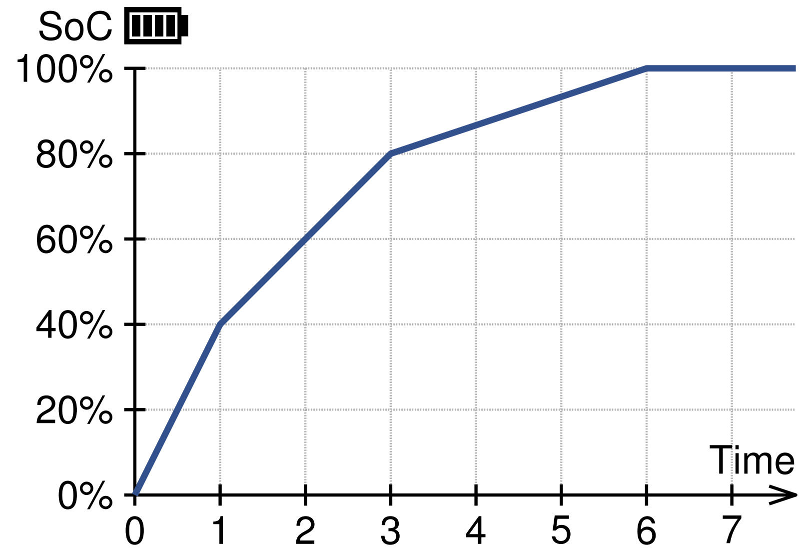
- A Function mapping charging time onto State of Charge (SoC)



Charging Functions

Charging station model:

- A Function mapping charging time onto State of Charge (SoC)



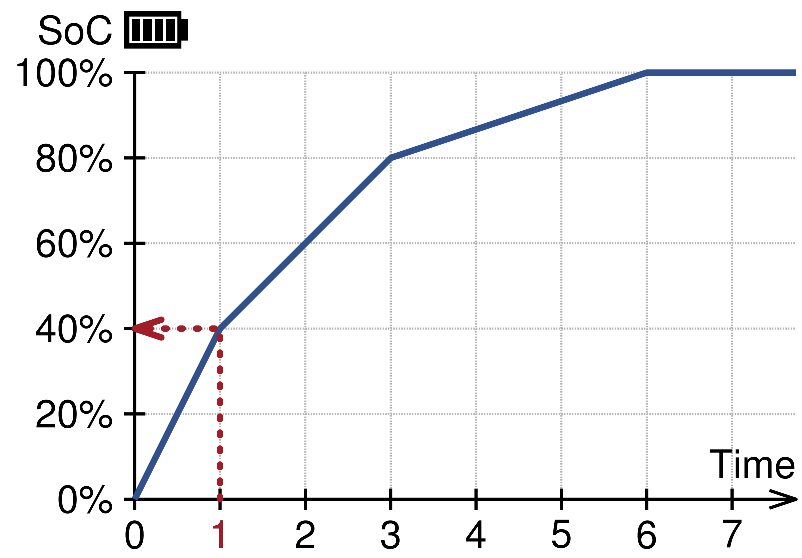
Properties:

- Monotonically increasing
  Spending time will not decrease SoC
- Concave
  As SoC rises, the charging rate may only decline

Charging Functions

Charging station model:

- A Function mapping charging time onto State of Charge (SoC)



Properties:

- Monotonically increasing
  Spending time will not decrease SoC
- Concave
  As SoC rises, the charging rate may only decline

Examples:

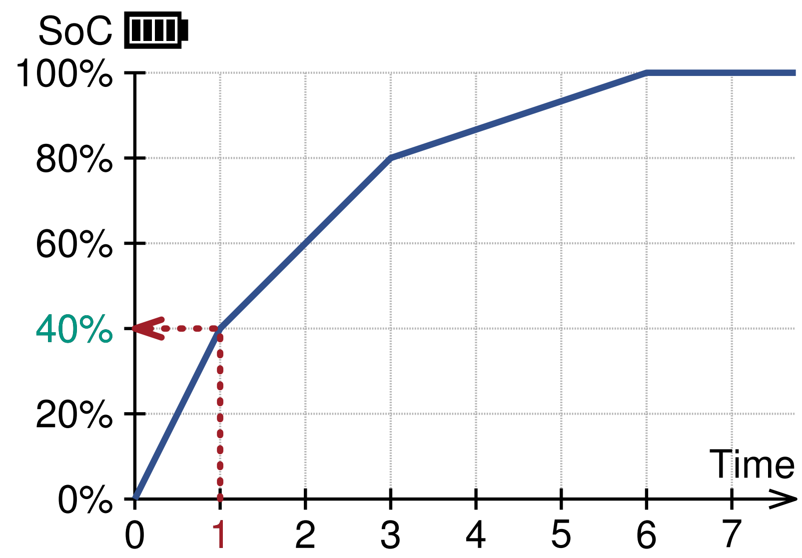
- Charging an empty battery for 1 time unit

Institute of Theoretical Informatics
Algorithmics Group

Charging Functions

Charging station model:

- A Function mapping charging time onto State of Charge (SoC)



Properties:

- Monotonically increasing
  Spending time will not decrease SoC
- Concave
  As SoC rises, the charging rate may only decline

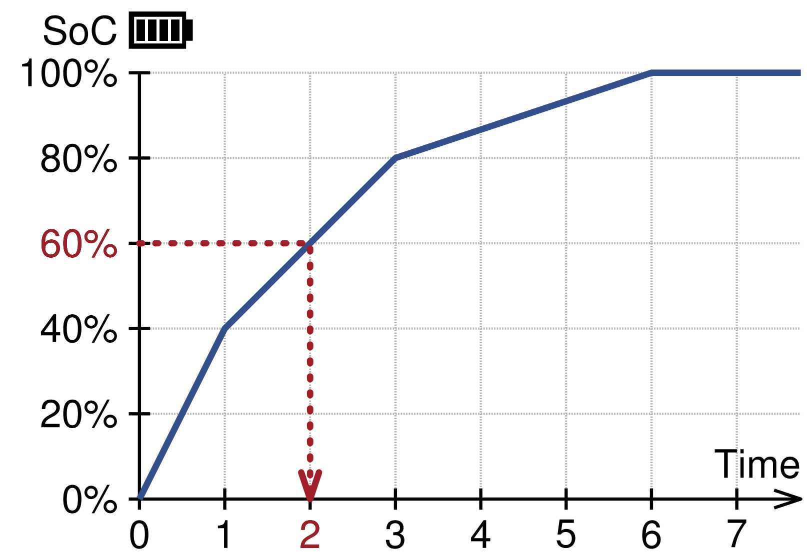
Examples:

- Charging an empty battery for 1 time unit $\Rightarrow$ 40% SoC

Charging Functions

Charging station model:

- A Function mapping charging time onto State of Charge (SoC)



Properties:

- Monotonically increasing
  Spending time will not decrease SoC
- Concave
  As SoC rises, the charging rate may only decline

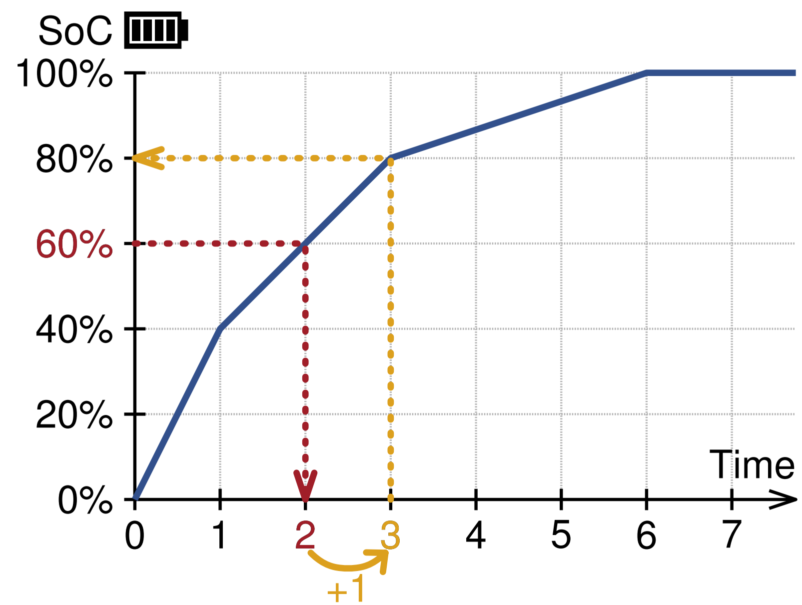
Examples:

- Charging an empty battery for 1 time unit ⇒ 40% SoC
- Charging an 60% full battery for 1 time unit

Charging Functions

Charging station model:

- A Function mapping charging time onto State of Charge (SoC)



Properties:

- Monotonically increasing
  Spending time will not decrease SoC
- Concave
  As SoC rises, the charging rate may only decline

Examples:

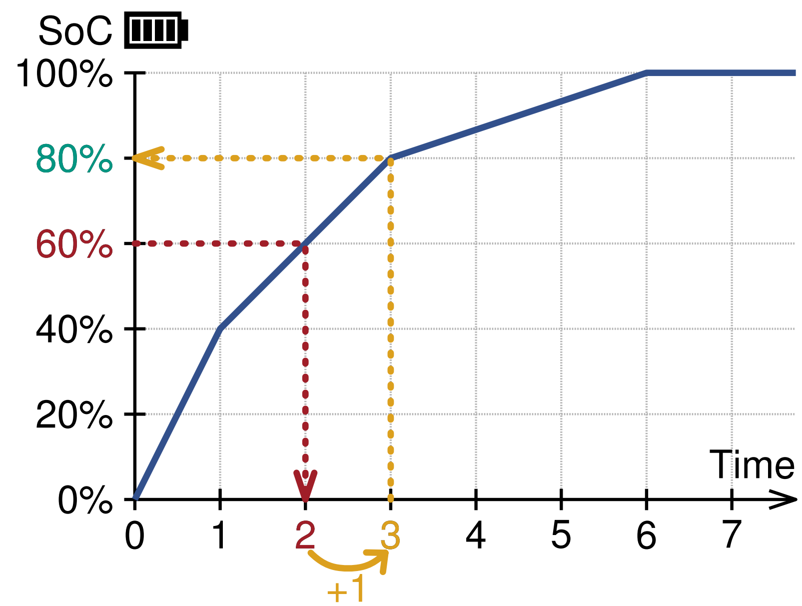
- Charging an empty battery for 1 time unit ⇒ 40% SoC
- Charging an 60% full battery for 1 time unit

Institute of Theoretical Informatics
Algorithmics Group

Charging Functions

Charging station model:

- A Function mapping charging time onto State of Charge (SoC)



Properties:

- Monotonically increasing
  Spending time will not decrease SoC
- Concave
  As SoC rises, the charging rate may only decline

Examples:

- Charging an empty battery for 1 time unit ⇒ 40% SoC
- Charging an 60% full battery for 1 time unit ⇒ 80% SoC

Charging Functions

Formally:

- A function $\mathrm{cf}\colon [0, M] \times \mathbb{R}_{\geq 0} \to [0, M]$, which maps
  - Initial SoC β_s and
  - Desired charging time τ_c onto
  - resulting SoC after charging
- Monotonically increasing (Charging longer $\Rightarrow$ increased SoC)
- Concave (Higher SoC $\Rightarrow$ lower charging rate)

Charging Functions

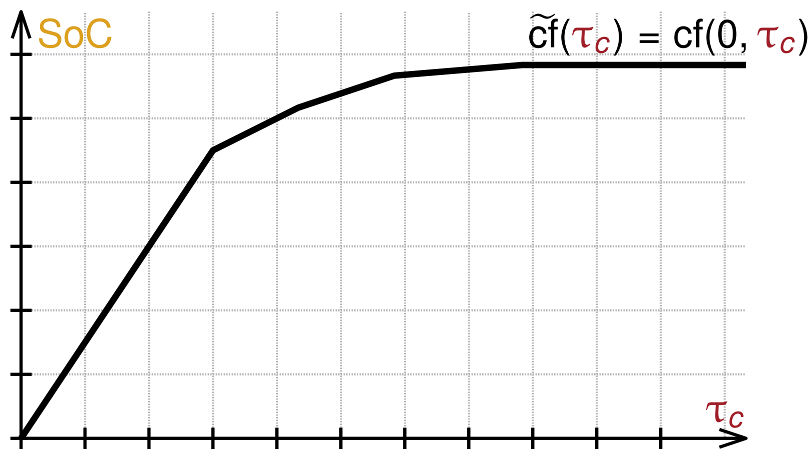
Formally:

- A function $\mathrm{cf}\colon [0, M] \times \mathbb{R}_{\geq 0} \to [0, M]$, which maps
  - Initial SoC β_s and
  - Desired charging time τ_c onto
  - resulting SoC after charging
- Monotonically increasing (Charging longer $\Rightarrow$ increased SoC)
- Concave (Higher SoC $\Rightarrow$ lower charging rate)

Observation: Realistic charging functions can be modeled using:

- An univariate function $\widetilde{\mathrm{cf}}\colon \mathbb{R}_{\geq 0} \to [0, M]$

$\mathrm{cf}(\beta, \tau_c) := \widetilde{\mathrm{cf}}(\tau_c + \widetilde{\mathrm{cf}}^{-1}(\beta))$



Charging Functions

Formally:

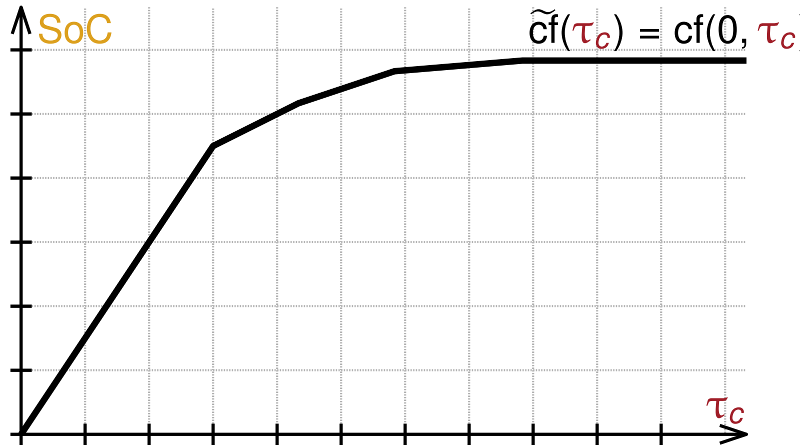
- A function $\mathrm{cf} \colon [0, M] \times \mathbb{R}_{\geq 0} \to [0, M]$, which maps
  - Initial SoC β_s and
  - Desired charging time τ_c onto
  - resulting SoC after charging
- Monotonically increasing (Charging longer $\Rightarrow$ increased SoC)
- Concave (Higher SoC $\Rightarrow$ lower charging rate)

Observation: Realistic charging functions can be modeled using:

- An univariate function $\widetilde{\mathrm{cf}} \colon \mathbb{R}_{\geq 0} \to [0, M]$

  $\mathrm{cf}(\beta, \tau_c) := \widetilde{\mathrm{cf}}(\tau_c + \widetilde{\mathrm{cf}}^{-1}(\beta))$

  $\mathrm{cf}(3, 2)$



Charging Functions

Formally:

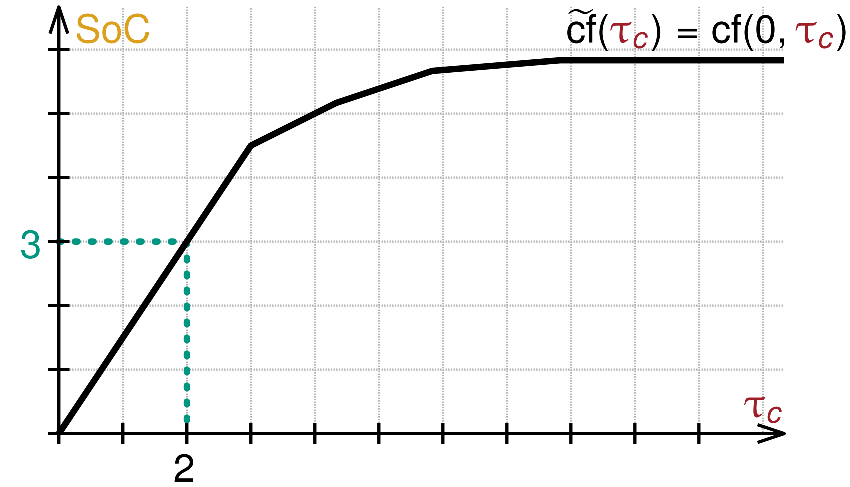
- A function $\mathrm{cf}\colon [0, M] \times \mathbb{R}_{\geq 0} \to [0, M]$, which maps
  - Initial SoC β_s and
  - Desired charging time τ_c onto
  - resulting SoC after charging
- Monotonically increasing (Charging longer $\Rightarrow$ increased SoC)
- Concave (Higher SoC $\Rightarrow$ lower charging rate)

Observation: Realistic charging functions can be modeled using:

- An univariate function $\widetilde{\mathrm{cf}}\colon \mathbb{R}_{\geq 0} \to [0, M]$

  $\mathrm{cf}(\beta, \tau_c) := \widetilde{\mathrm{cf}}(\tau_c + \widetilde{\mathrm{cf}}^{-1}(\beta))$

  $\mathrm{cf}(3, 2) = \widetilde{\mathrm{cf}}(2 + \widetilde{\mathrm{cf}}^{-1}(3))$



Charging Functions

Formally:

- A function $\mathrm{cf}\colon [0, M] \times \mathbb{R}_{\geq 0} \to [0, M]$, which maps
  - Initial SoC β_s and
  - Desired charging time τ_c onto
  - resulting SoC after charging
- Monotonically increasing (Charging longer $\Rightarrow$ increased SoC)
- Concave (Higher SoC $\Rightarrow$ lower charging rate)

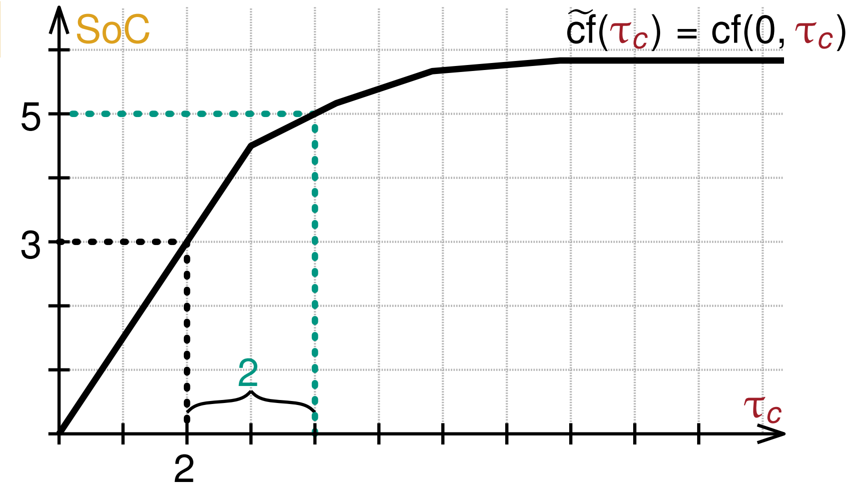
Observation: Realistic charging functions can be modeled using:

- An univariate function $\widetilde{\mathrm{cf}}\colon \mathbb{R}_{\geq 0} \to [0, M]$

$$\mathrm{cf}(\beta, \tau_c) := \widetilde{\mathrm{cf}}(\tau_c + \widetilde{\mathrm{cf}}^{-1}(\beta))$$

$$\mathrm{cf}(3, 2) = \widetilde{\mathrm{cf}}(2 + \widetilde{\mathrm{cf}}^{-1}(3))$$

$$= \widetilde{\mathrm{cf}}(2 + 2)$$



Charging Functions

Formally:

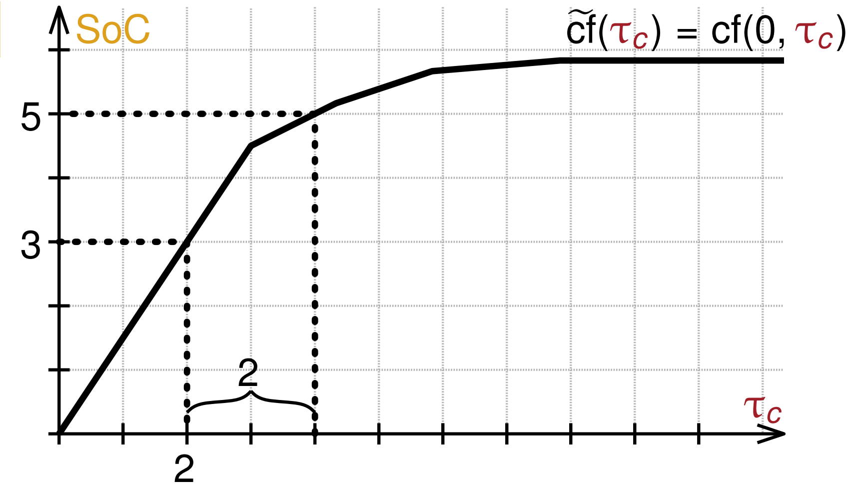
- A function $\mathrm{cf}\colon [0, M] \times \mathbb{R}_{\geq 0} \to [0, M]$, which maps
  - Initial SoC β_s and
  - Desired charging time τ_c onto
  - resulting SoC after charging
- Monotonically increasing (Charging longer $\Rightarrow$ increased SoC)
- Concave (Higher SoC $\Rightarrow$ lower charging rate)

Observation: Realistic charging functions can be modeled using:

- An univariate function $\widetilde{\mathrm{cf}}\colon \mathbb{R}_{\geq 0} \to [0, M]$

$$\mathrm{cf}(\beta, \tau_c) := \widetilde{\mathrm{cf}}(\tau_c + \widetilde{\mathrm{cf}}^{-1}(\beta))$$

$$\begin{aligned}\mathrm{cf}(3, 2) &= \widetilde{\mathrm{cf}}(2 + \widetilde{\mathrm{cf}}^{-1}(3)) \\ &= \widetilde{\mathrm{cf}}(2 + 2) \\ &= 5\end{aligned}$$



Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Algorithm:

- Based on multi-criteria Dijkstra
- If no charging station has been used: label = tuple (trip time, SoC)
- Per vertex: Maintain a set of Pareto-optimal labels

Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Algorithm:

- Based on multi-criteria Dijkstra
- If no charging station has been used: label = tuple (trip time, SoC)
- Per vertex: Maintain a set of Pareto-optimal labels

Problem: When reaching a charging station: How long to stay?

Charging Function Propagation (CFP)

Algorithm:

- Based on multi-criteria Dijkstra
- If no charging station has been used: label = tuple (trip time, SoC)
- Per vertex: Maintain a set of Pareto-optimal labels

Problem: When reaching a charging station: How long to stay?

- Depends on the remaining path to t
- Optimal SoC for departure is yet unknown

Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Algorithm:

- Based on multi-criteria Dijkstra
- If no charging station has been used: label = tuple (trip time, SoC)
- Per vertex: Maintain a set of Pareto-optimal labels

Problem: When reaching a charging station: How long to stay?

- Depends on the remaining path to t
- Optimal SoC for departure is yet unknown

Solution:

- Delay this decision!
- Keep track of the last passed charging station

Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\dots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\dots,v]}$ of the path from u to v (initially $\perp$)

Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\dots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\dots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\dots,v]})$

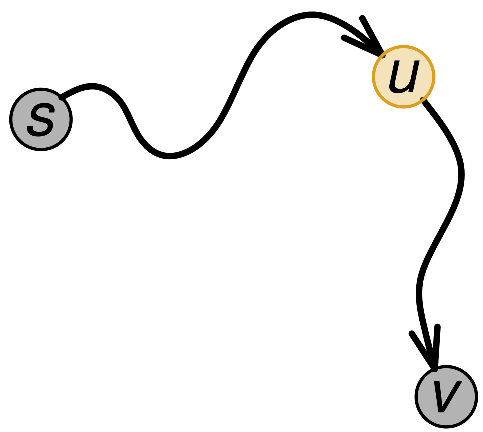
Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\ldots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\ldots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\ldots,v]})$

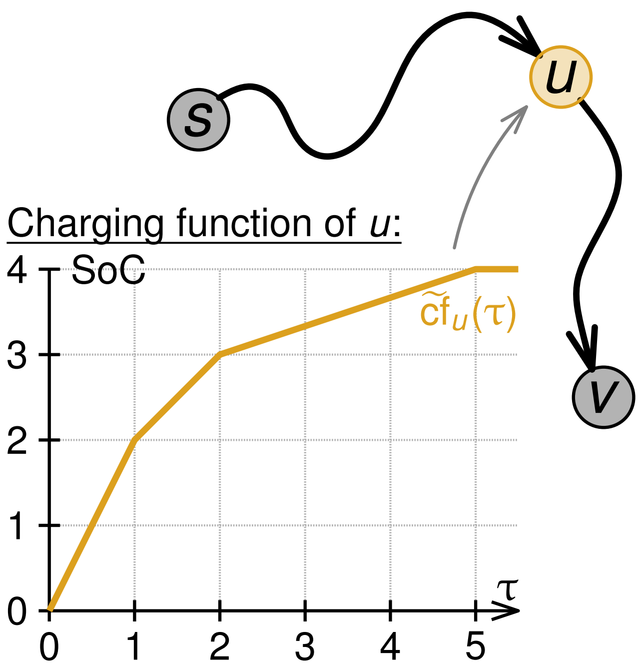


Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\ldots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\ldots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\ldots,v]})$

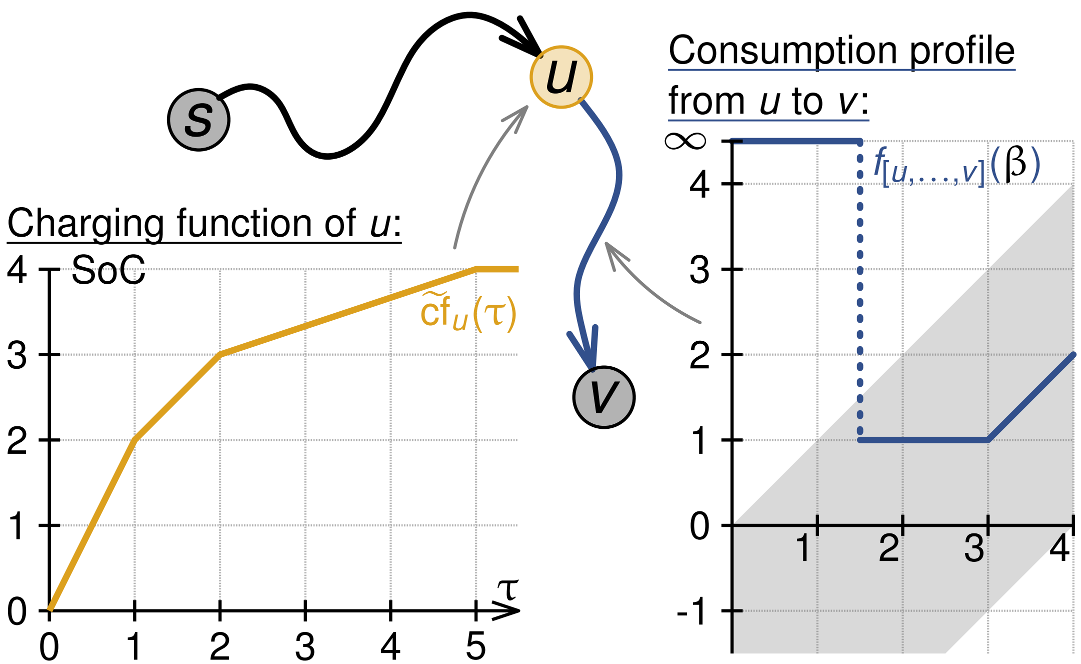


Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\ldots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\ldots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\ldots,v]})$



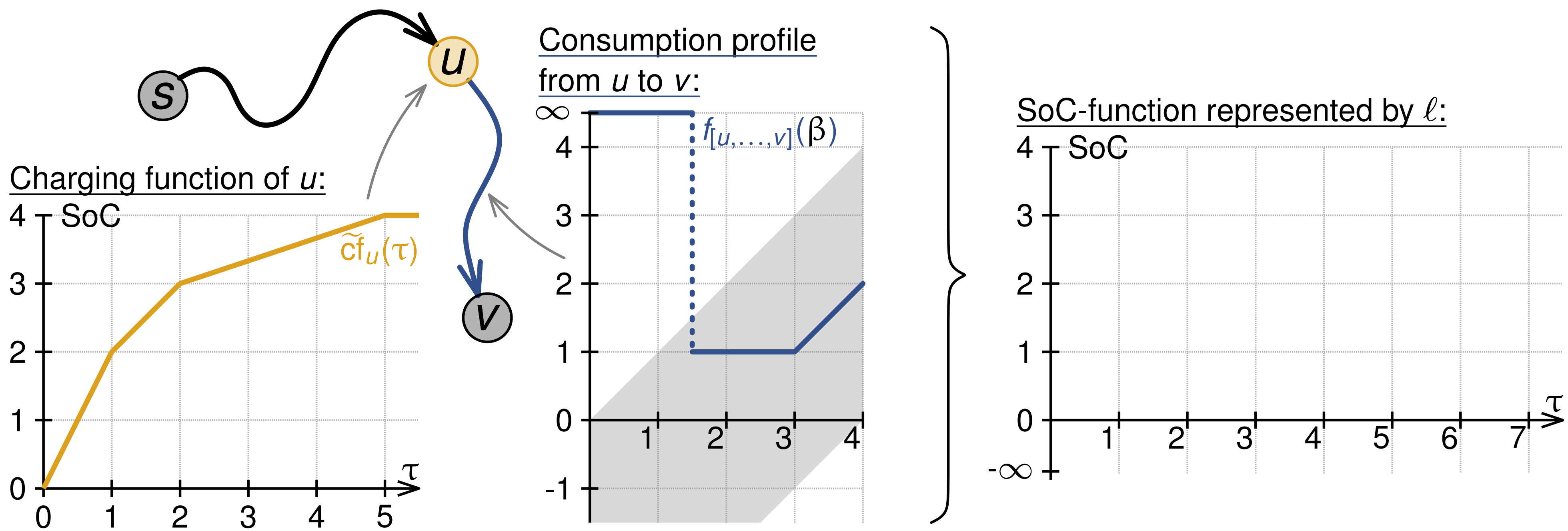
Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\dots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\dots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\dots,v]})$



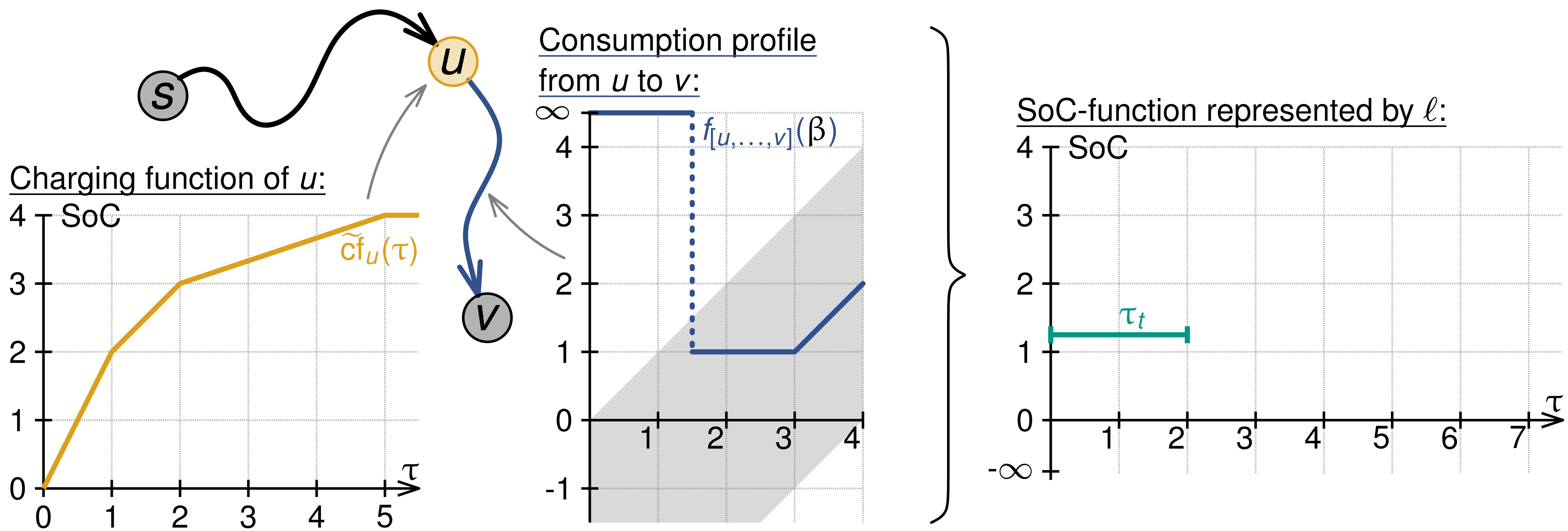
Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\dots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\dots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\dots,v]})$

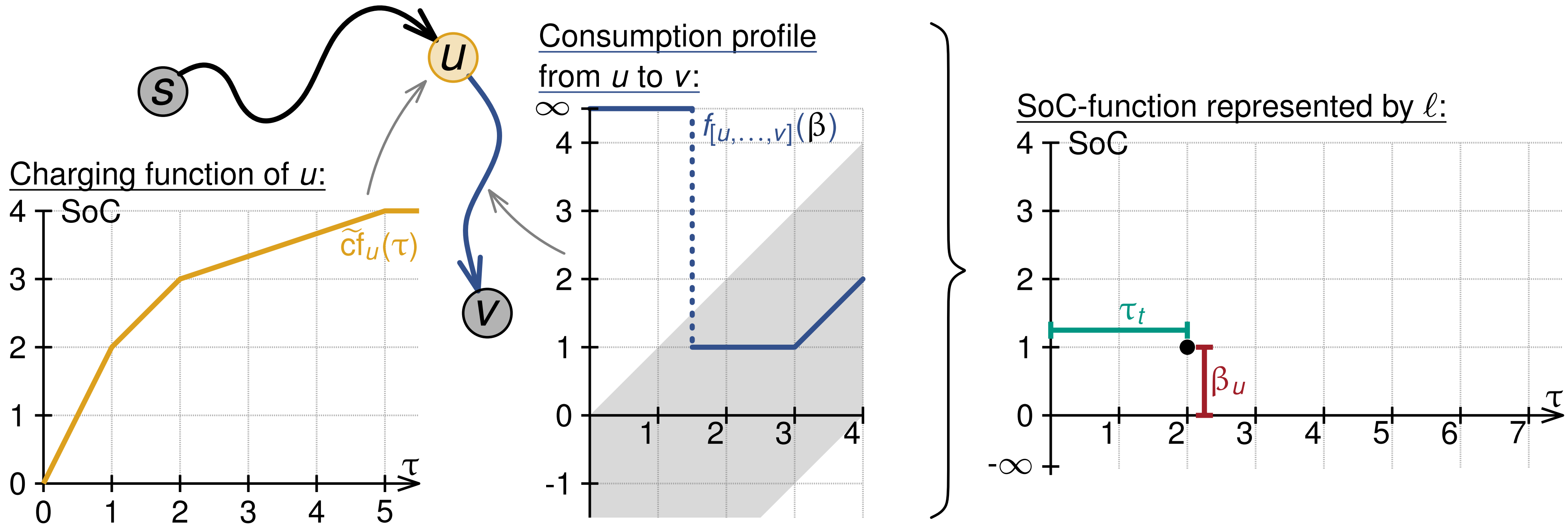


Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\dots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\dots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\dots,v]})$

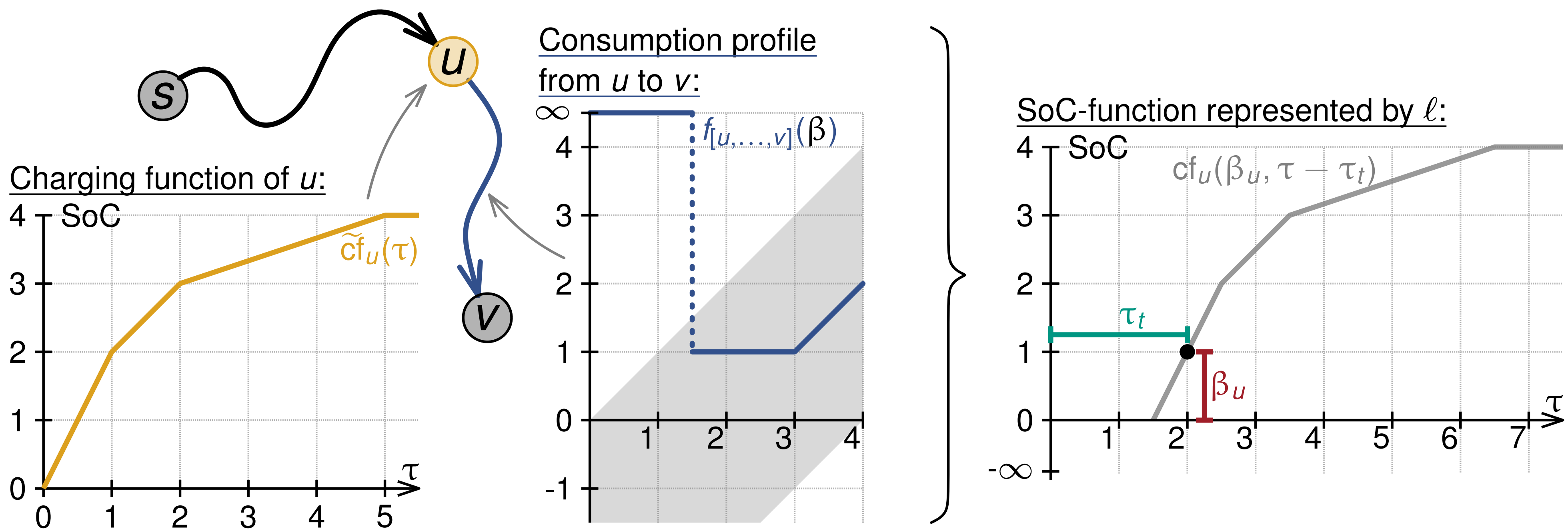


Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\ldots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\ldots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\ldots,v]})$



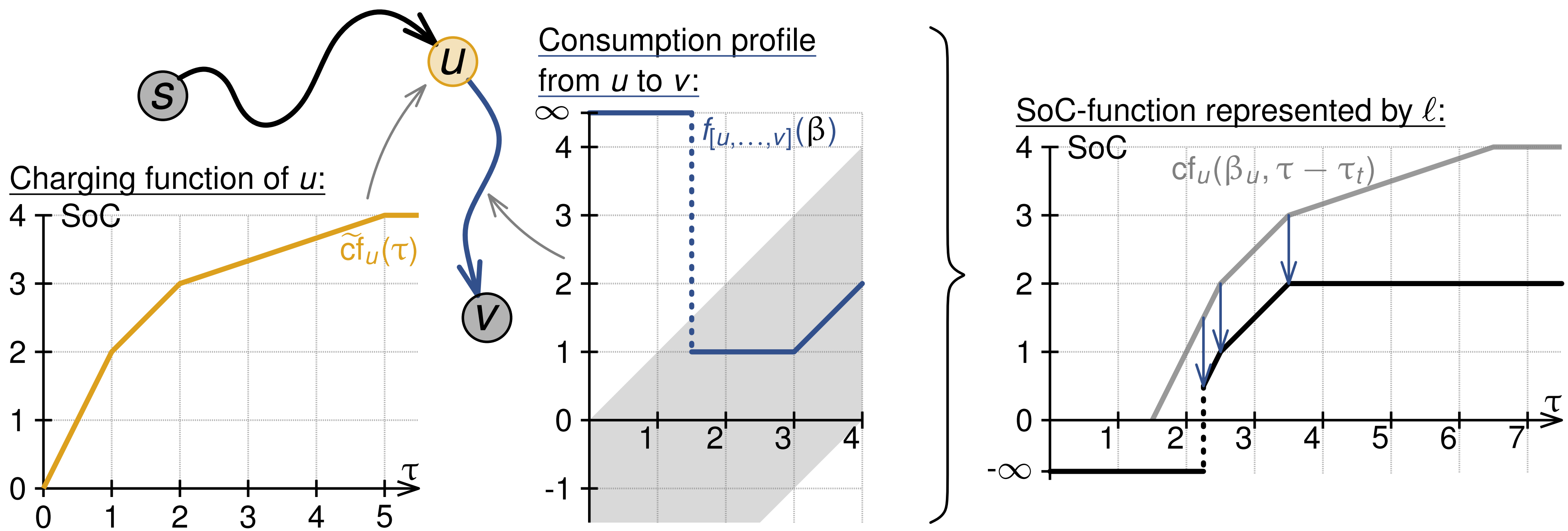
Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\dots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\dots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\dots,v]})$



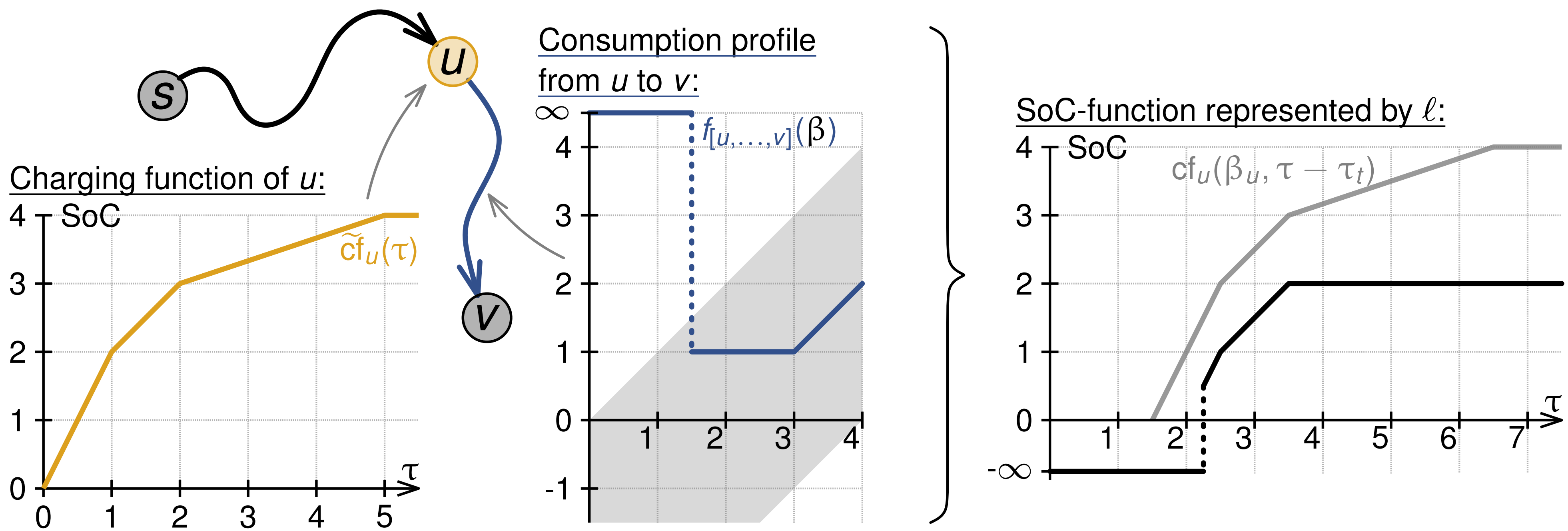
Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\ldots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\ldots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\ldots,v]})$



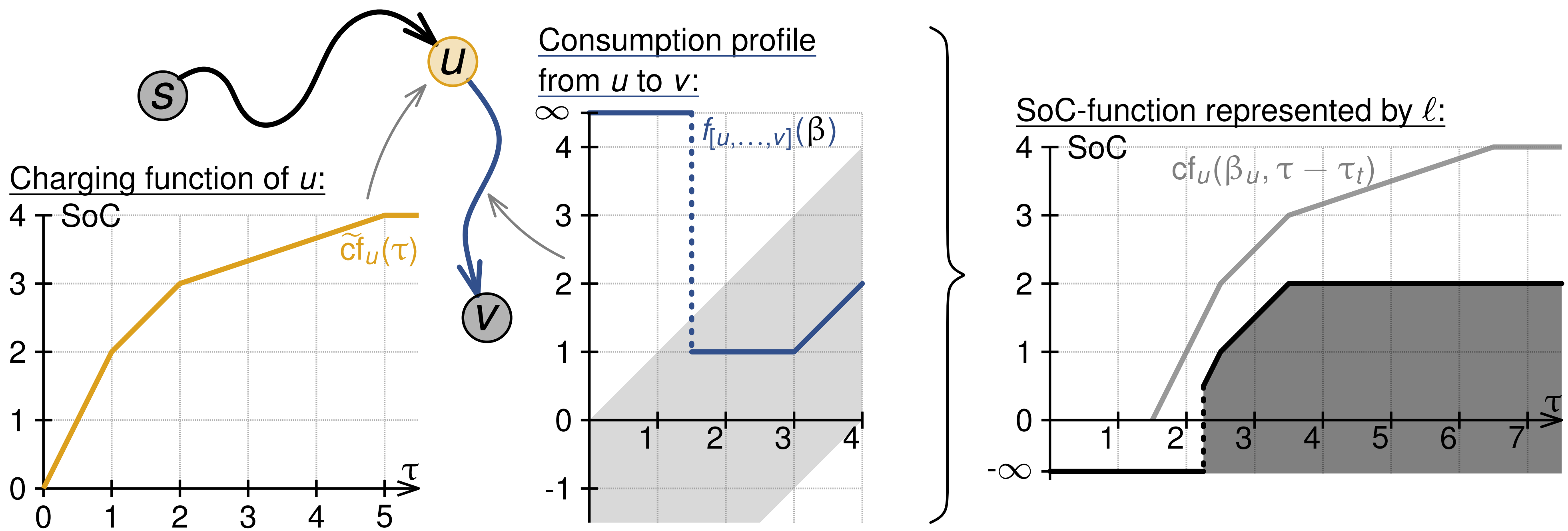
Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Label: A label ℓ at vertex v is a quadruple $(\tau_t, \beta_u, u, f_{[u,\ldots,v]})$ with:

- Trip time τ_t from s to v (including charging times except at u)
- SoC β_u by which the last charging station (u) was reached
- The last seen charging station u (initially $\perp$)
- Consumption profile $f_{[u,\ldots,v]}$ of the path from u to v (initially $\perp$)

Interpretation: Label ℓ at $v = (\tau_t = 2, \beta_u = 1, u, f_{[u,\ldots,v]})$



Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Edge relaxation:

- Label propagation along an edge: Constant time operation
- Given a label $\ell_v = (\tau_t, \beta_u, u, f_{[u,\ldots,v]})$ at v and an edge $e = (v, w)$:
  $\ell_w := (\tau_t + \tau_d(e), \beta_u, u, f_{[u,\ldots,v]} \circ f_e)$

Institute of Theoretical Informatics
Algorithmics Group

Charging Function Propagation (CFP)

Edge relaxation:

- Label propagation along an edge: Constant time operation
- Given a label $\ell_v = (\tau_t, \beta_u, u, f_{[u,\dots,v]})$ at v and an edge $e = (v, w)$:
  $\ell_w := (\tau_t + \tau_d(e), \beta_u, u, f_{[u,\dots,v]} \circ f_e)$

Label Pareto-domination:

- Linear in complexity of charging function
- But: Pareto-domination is checked rarely

  (Only for the next label to be settled)

Charging Function Propagation (CFP)

Edge relaxation:

- Label propagation along an edge: Constant time operation
- Given a label $\ell_v = (\tau_t, \beta_u, u, f_{[u,\ldots,v]})$ at v and an edge $e = (v, w)$:
  $\ell_w := (\tau_t + \tau_d(e), \beta_u, u, f_{[u,\ldots,v]} \circ f_e)$

Label Pareto-domination:

- Linear in complexity of charging function
- But: Pareto-domination is checked rarely

  (Only for the next label to be settled)

Reaching another charging station:

- Our labels can store at most one charging station
- Have to specify the charging time for the second last station
- Theorem 1 in the paper proves that this is easy

Institute of Theoretical Informatics
Algorithmics Group

Speed Up Techniques

CFP & Contraction Hierarchies:

- Shortcut-based technique
- Shortcuts have to maintain Pareto-sets
  (w.r.t. travel time & energy consumption)

  Problem: Shortcut size grows exponentially $\Rightarrow$ uncontracted Core

Institute of Theoretical Informatics
Algorithmics Group

Speed Up Techniques

CFP & Contraction Hierarchies:

- Shortcut-based technique
- Shortcuts have to maintain Pareto-sets
  (w.r.t. travel time & energy consumption)

  Problem: Shortcut size grows exponentially $\Rightarrow$ uncontracted Core

CFP & A*:

- Goal directed technique
- Potential function $\pi(v)$ estimates time from v to target t

  Idea: Use bivariate potential $\pi(v, \beta)$ to account for SoC β

Speed Up Techniques

CFP & Contraction Hierarchies:

- Shortcut-based technique
- Shortcuts have to maintain Pareto-sets
  (w.r.t. travel time & energy consumption)

  Problem: Shortcut size grows exponentially $\Rightarrow$ uncontracted Core

CFP & A*:

- Goal directed technique
- Potential function $\pi(v)$ estimates time from v to target t

  Idea: Use bivariate potential $\pi(v, \beta)$ to account for SoC β

CHArge = CH & A* & CFP:

- Stop contraction as mean core degree gets to big
- Combine CH-query with A*-search on core

Institute of Theoretical Informatics
Algorithmics Group

Experiments

- **Road Networks:**

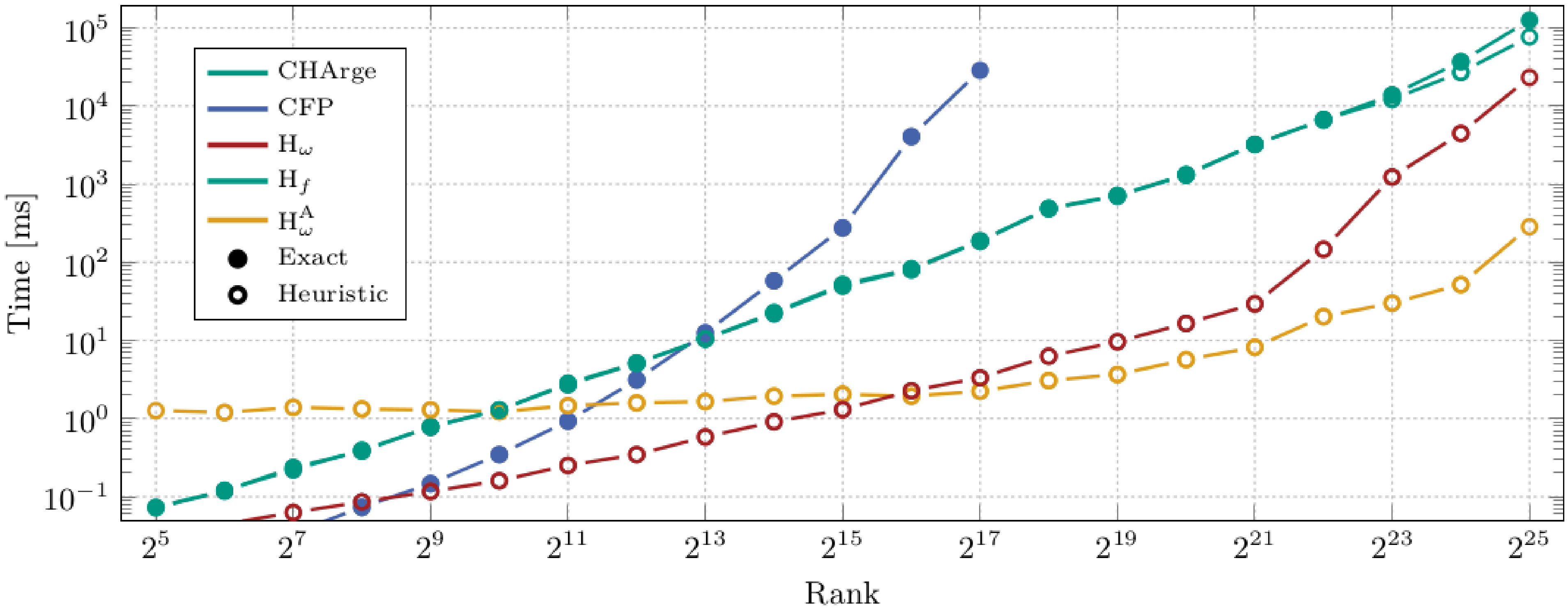
Instance	# Vertices	# Arcs	# Arcs with $\gamma < 0$	# CS
Ger (PTV)	4 692 091	10 805 429	1 119 710 (10.36%)	1 966
Eur (PTV)	22 198 628	51 088 095	6 060 648 (11.86%)	13 810
Osg (OSM)	5 588 146	11 711 088	1 142 391 (9.75%)	643

- **Elevation data:** SRTM, v4.1 (`srtm.csi.cgiar.org`)
- **Energy consumption:** [Hausberger et al. 09]
  Micro-scale emission model (PHEM), calibrated to Peugeot iOn
- **Charging stations:** ChargeMap (`chargemap.com`)
  random distributions
- **Station Types:**
  - Battery swapping stations (BSS)
  - Superchargers (50 % in 20 min, 80 % in 40 min)
  - Regular stations (44 kW; 22 kW; 11 kW)

Experiments

	Instance	# CS	M	Prepro. [m:s]	Query Feas.	Query [ms]
Only BSS	Osg	1 000	100 km	11:37	100 %	122
	Osg	100	150 km	11:10	99 %	206
	Osg	643	100 km	11:21	98 %	326
	Osg	643	150 km	11:28	99 %	308
Only BSS	Ger	1 966	16 kWh	5:03	100 %	1 398
	Ger	1 966	85 kWh	4:59	100 %	1 013
	Eur	13 810	16 kWh	30:32	63 %	10 786
	Eur	13 810	85 kWh	30:16	100 %	47 921
Mixed CS	Ger	1 966	16 kWh	5:03	100 %	8 629
	Ger	1 966	85 kWh	4:59	100 %	2 614
	Eur	13 810	16 kWh	30:32	63 %	24 148
	Eur	13 810	85 kWh	30:16	100 %	86 193

Experiments



- Good scalability
- Even better with good heuristics:
  Europe 0.1 – 1 sec; Germany 20 – 100 ms
  often optimal solutions, mean error $\sim$1%

Institute of Theoretical Informatics
Algorithmics Group

Conclusion

- Route planning for EVs raises new challenges
  - Considering energy consumption is essential
  - Charging stops should be planned in advance
  - Results in a (weakly) NP-hard problem

- Our approach **CHArge**:
  - Can handle arbitrary charging station types
  - Moderate preprocessing times
  - Fast queries on continental sized networks:
    Europe $\sim$1 min; Germany $\sim$1 sec
    $\Rightarrow$ More general *and* faster than state-of-the-art (e. g., [Storandt '12])
  - Even better results possible, using heuristics

Institute of Theoretical Informatics
Algorithmics Group

Conclusion

- More details:

  M. Baum, J. Dibbelt, A. Gemsa, D. Wagner, T. Zündorf, *Shortest Feasible Paths with Charging Stops for Battery Electric Vehicles*, SIGSPATIAL'15, November 3–6, 2015, Bellevue, WA, USA

Institute of Theoretical Informatics
Algorithmics Group