On Generalizations of the Parking Permit Problem and Network Leasing Problems

Murilo S. de Lima[1], Mário C. San Felice[2], Orlando Lee[1]

[1]Unicamp – Brazil
[2]USP – Brazil

LAGOS 2017
September 11-15, 2017

Supported by

Introduction

Parking Permit Problem

Johnny goes to work every day

Parking Permit Problem

Johnny goes to work every day

He lives close to his job

Parking Permit Problem

Johnny goes to work every day

He lives close to his job

He is ~~a hipster~~ an environmental-aware person

Parking Permit Problem

Johnny goes to work every day

He lives close to his job

He is ~~a hipster~~ an environmental-aware person

- sunny $\rightarrow$ walks

Parking Permit Problem

Johnny goes to work every day

He lives close to his job

He is ~~a hipster~~ an environmental-aware person

- sunny $\rightarrow$ walks
- rains $\rightarrow$ clean-fuel car

Parking Permit Problem

Johnny goes to work every day

He lives close to his job

He is ~~a hipster~~ an environmental-aware person

- sunny $\rightarrow$ walks
- rains $\rightarrow$ clean-fuel car

Parking costs money!

- K different permit lengths: e.g., daily, weekly, monthly

Parking Permit Problem

Johnny goes to work every day

He lives close to his job

He is ~~a hipster~~ an environmental-aware person

- sunny $\rightarrow$ walks
- rains $\rightarrow$ clean-fuel car

Parking costs money!

- K different permit lengths: e.g., daily, weekly, monthly
- each rainy day $\rightarrow$ valid permit

Parking Permit Problem

Johnny goes to work every day

He lives close to his job

He is ~~a hipster~~ an environmental-aware person

- sunny $\rightarrow$ walks
- rains $\rightarrow$ clean-fuel car

Parking costs money!

- K different permit lengths: e.g., daily, weekly, monthly
- each rainy day $\rightarrow$ valid permit
- permits expire (even if not used!)

Parking Permit Problem

Johnny goes to work every day

He lives close to his job

He is ~~a hipster~~ an environmental-aware person

- sunny $\rightarrow$ walks
- rains $\rightarrow$ clean-fuel car

Parking costs money!

- K different permit lengths: e.g., daily, weekly, monthly
- each rainy day $\rightarrow$ valid permit
- permits expire (even if not used!)
- sub-additive costs (economy of scale)

Parking Permit Problem

Johnny goes to work every day

He lives close to his job

He is ~~a hipster~~ an environmental-aware person

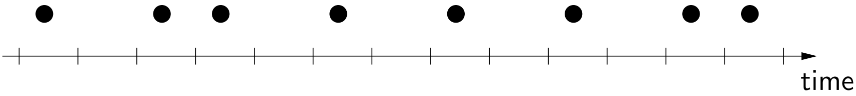
- sunny $\rightarrow$ walks
- rains $\rightarrow$ clean-fuel car

Parking costs money!

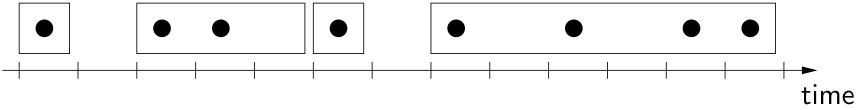
- K different permit lengths: e.g., daily, weekly, monthly
- each rainy day $\rightarrow$ valid permit
- permits expire (even if not used!)
- sub-additive costs (economy of scale)

How can Johnny save money for a trip to Marseille?

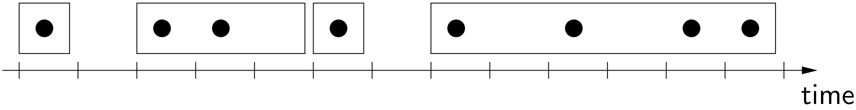
Parking Permit Problem (2)



Parking Permit Problem (2)

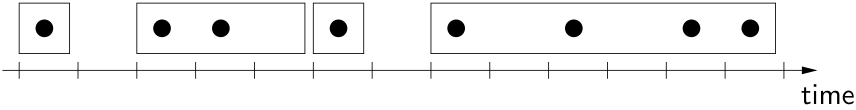


Parking Permit Problem (2)



Solvable via dynamic programming (Meyerson, 2005)

Parking Permit Problem (2)



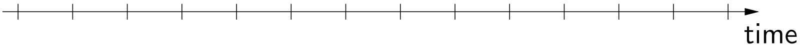
Solvable via dynamic programming (Meyerson, 2005)

Not very realistic

- Johnny does not know the future!
- weather forecast → unreliable for long-term

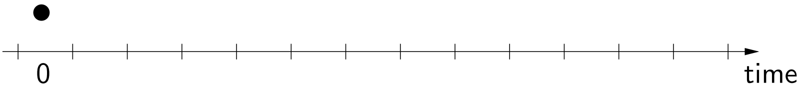
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



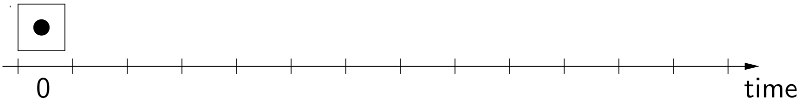
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



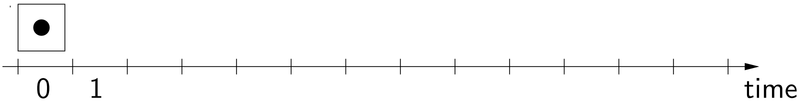
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



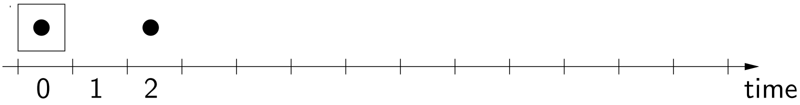
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



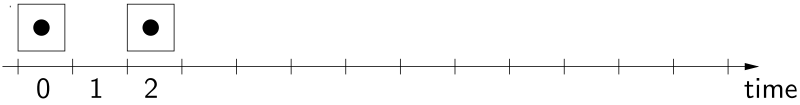
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



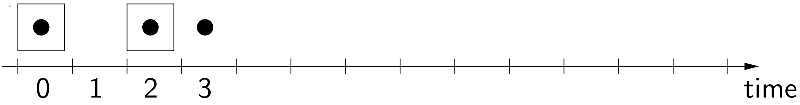
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



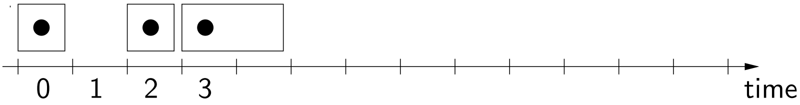
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



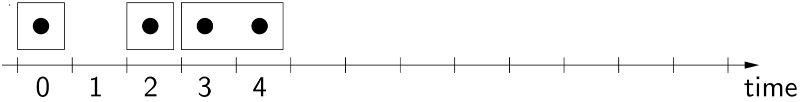
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



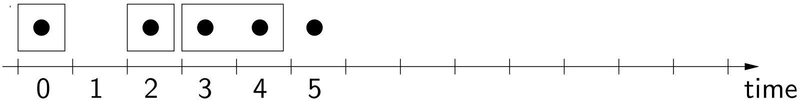
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



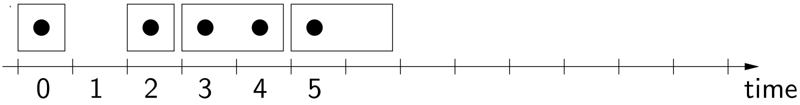
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



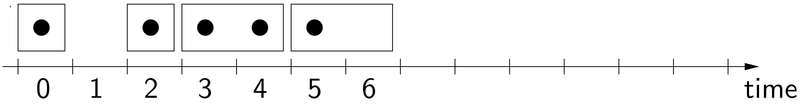
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



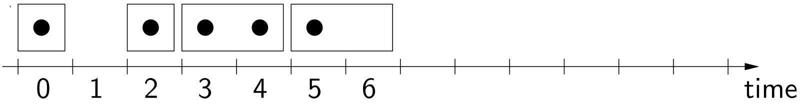
Online Parking Permit Problem

length	cost
1	2
2	3
4	4



Online Parking Permit Problem

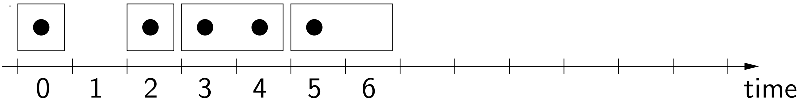
length	cost
1	2
2	3
4	4



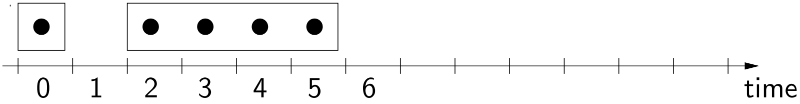
cost: 10

Online Parking Permit Problem

length	cost
1	2
2	3
4	4



cost: 10



optimum costs 6

Online Parking Permit Problem (2)

(Meyerson, 2005)

- deterministic $\mathrm{O}(K)$-competitive algorithm
- deterministic $\Omega(K)$ lower bound

Online Parking Permit Problem (2)

(Meyerson, 2005)

- deterministic $\mathrm{O}(K)$-competitive algorithm
- deterministic $\Omega(K)$ lower bound
- randomized $\mathrm{O}(\lg K)$-competitive algorithm
- randomized $\Omega(\lg K)$ lower bound

Online Parking Permit Problem (2)

(Meyerson, 2005)

- deterministic $\mathrm{O}(K)$-competitive algorithm
- deterministic $\Omega(K)$ lower bound
- randomized $\mathrm{O}(\lg K)$-competitive algorithm
- randomized $\Omega(\lg K)$ lower bound

Seminal problem for **leasing optimization**

Online Parking Permit Problem (2)

(Meyerson, 2005)

- deterministic $\mathrm{O}(K)$-competitive algorithm
- deterministic $\Omega(K)$ lower bound
- randomized $\mathrm{O}(\lg K)$-competitive algorithm
- randomized $\Omega(\lg K)$ lower bound

Seminal problem for **leasing optimization**

- cloud service allocation

Online Parking Permit Problem (2)

(Meyerson, 2005)

- deterministic $\mathrm{O}(K)$-competitive algorithm
- deterministic $\Omega(K)$ lower bound
- randomized $\mathrm{O}(\lg K)$-competitive algorithm
- randomized $\Omega(\lg K)$ lower bound

Seminal problem for **leasing optimization**

- cloud service allocation

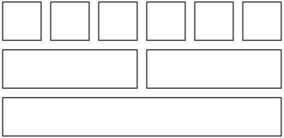
Imply results for **Steiner Leasing Problem** (Meyerson, 2005)

- approximation by tree metrics
  (Bartal, 1996; Fakcharoenphol *et al.*, 2004)

Interval Model

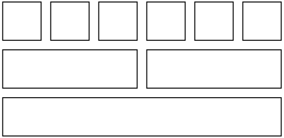
Interval Model

Permit lengths divide each other

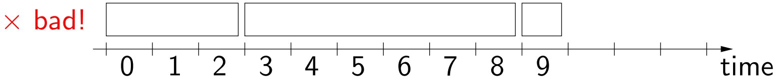


Interval Model

Permit lengths divide each other

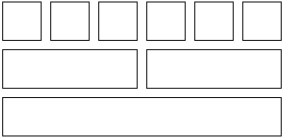


Permits start on multiples of their lenghts

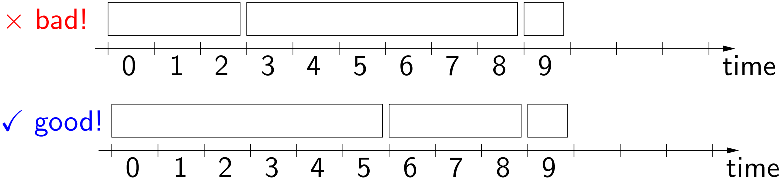


Interval Model

Permit lengths divide each other

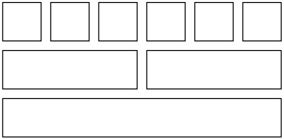


Permits start on multiples of their lenghts

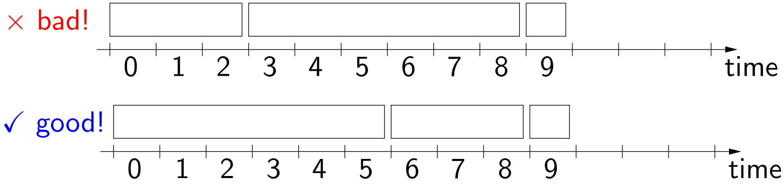


Interval Model

Permit lengths divide each other



Permits start on multiples of their lenghts



α-competitive under IM $\rightarrow$ 4α-competitive in general

Multi Parking Permit Problem

Multi Parking Permit Problem

~~Communists~~ Labor union obliges Johnny's company to pay employees' permits

Multi Parking Permit Problem

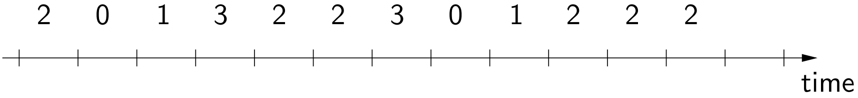
~~Communists~~ Labor union obliges Johnny's company to pay employees' permits

The same permit can be used by different employees on different days

Multi Parking Permit Problem

~~Communists~~ Labor union obliges Johnny's company to pay employees' permits

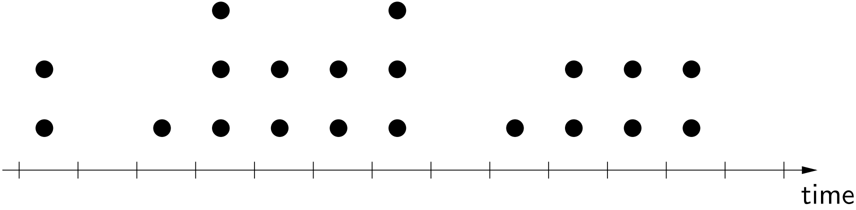
The same permit can be used by different employees on different days



Multi Parking Permit Problem

~~Communists~~ Labor union obliges Johnny's company to pay employees' permits

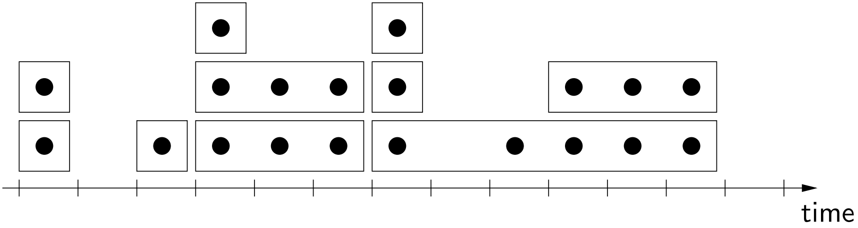
The same permit can be used by different employees on different days



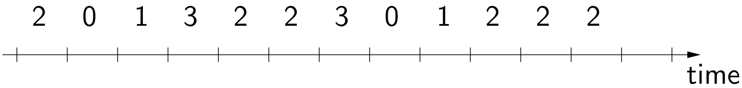
Multi Parking Permit Problem

~~Communists~~ Labor union obliges Johnny's company to pay employees' permits

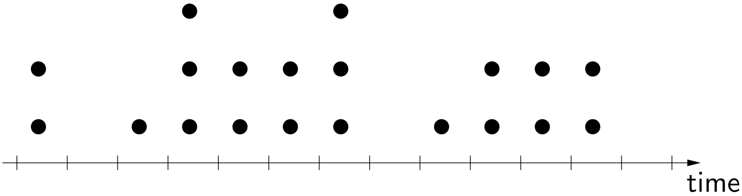
The same permit can be used by different employees on different days



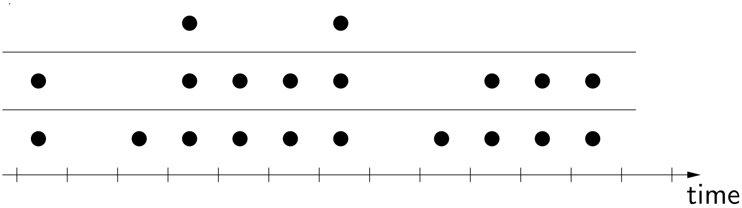
Hanoi Tower Property



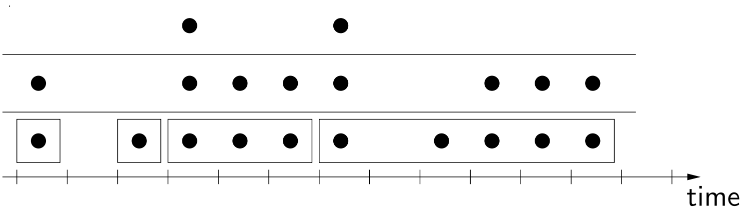
Hanoi Tower Property



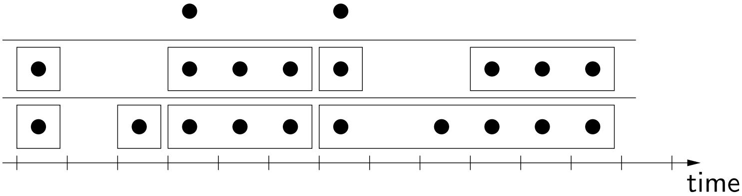
Hanoi Tower Property



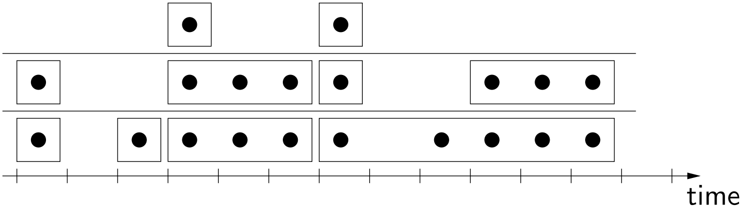
Hanoi Tower Property



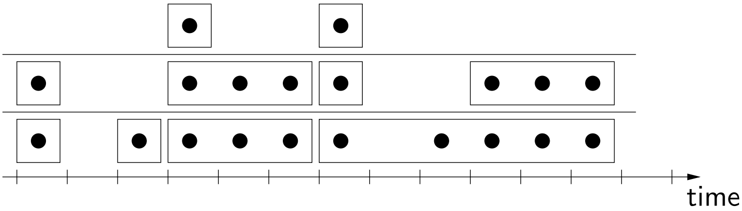
Hanoi Tower Property



Hanoi Tower Property

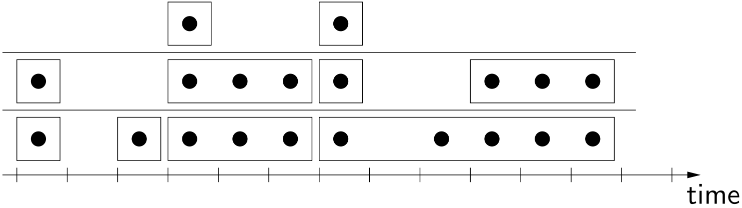


Hanoi Tower Property



opt on each level $\rightarrow$ opt for the whole input

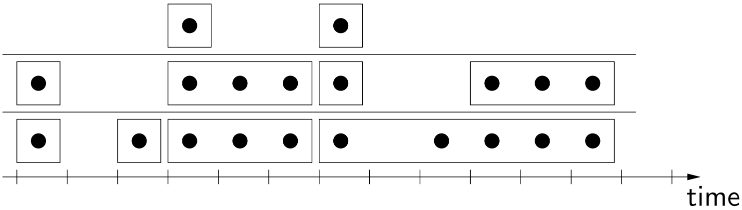
Hanoi Tower Property



opt on each level → opt for the whole input

Permits are stacked as in Hanoi tower

Hanoi Tower Property

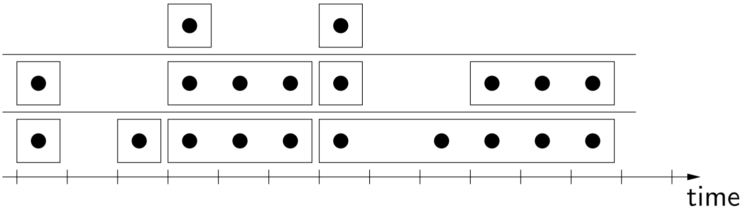


opt on each level $\rightarrow$ opt for the whole input

Permits are stacked as in Hanoi tower

Induces a simple reduction algorithm (pseudo-polynomial!)

Hanoi Tower Property



opt on each level $\rightarrow$ opt for the whole input

Permits are stacked as in Hanoi tower

Induces a simple reduction algorithm (pseudo-polynomial!)

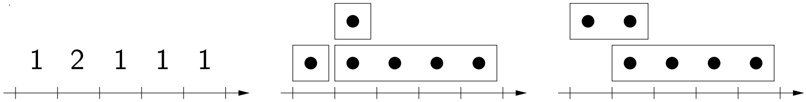
Relies on Interval Model

Hanoi Tower Property (2)

No Interval Model $\rightarrow$ no Hanoi Tower Property!

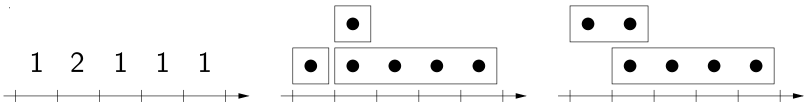
Hanoi Tower Property (2)

No Interval Model $\rightarrow$ no Hanoi Tower Property!



Hanoi Tower Property (2)

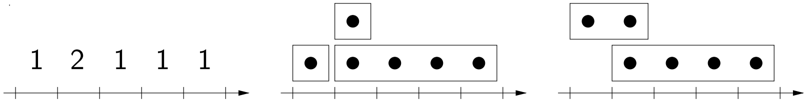
No Interval Model → no Hanoi Tower Property!



We conjecture that the problem is NP-hard if we do not assume the Interval Model.

Hanoi Tower Property (2)

No Interval Model → no Hanoi Tower Property!



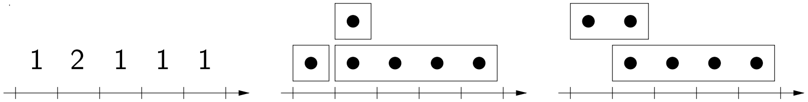
We conjecture that the problem is NP-hard if we do not assume the Interval Model.

Under Interval Model we have:

- poly-time exact algorithm (4-approx. in general)

Hanoi Tower Property (2)

No Interval Model → no Hanoi Tower Property!



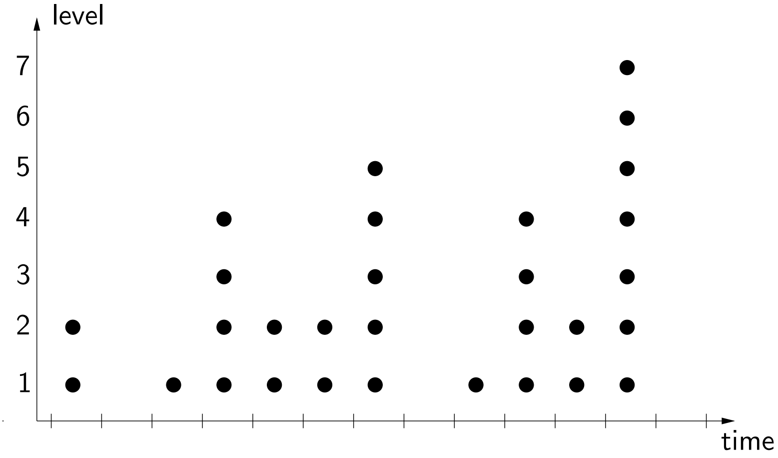
We conjecture that the problem is NP-hard if we do not assume the Interval Model.

Under Interval Model we have:

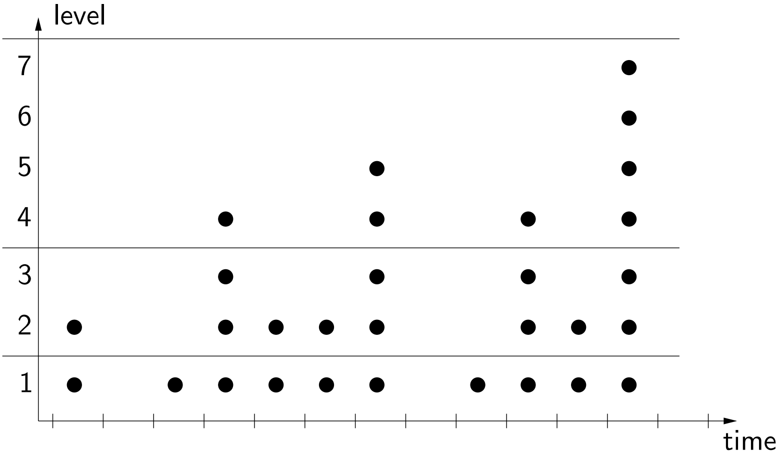
- poly-time exact algorithm (4-approx. in general)
- poly-time approximation-preserving reduction

Poly-time approximation-preserving reduction

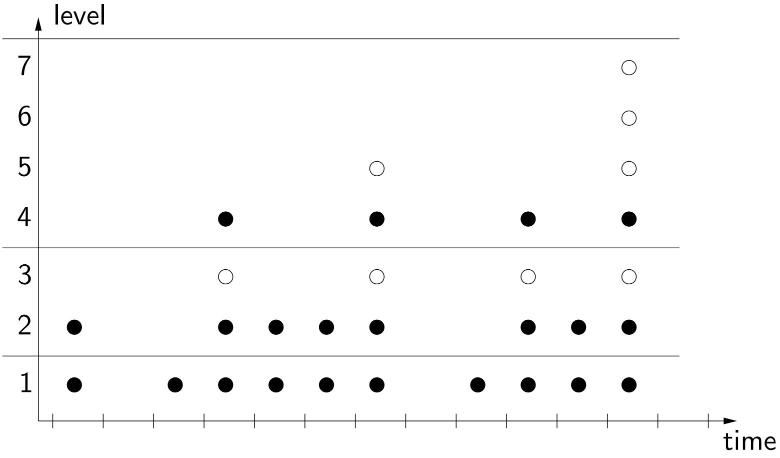
Poly-time approximation-preserving reduction



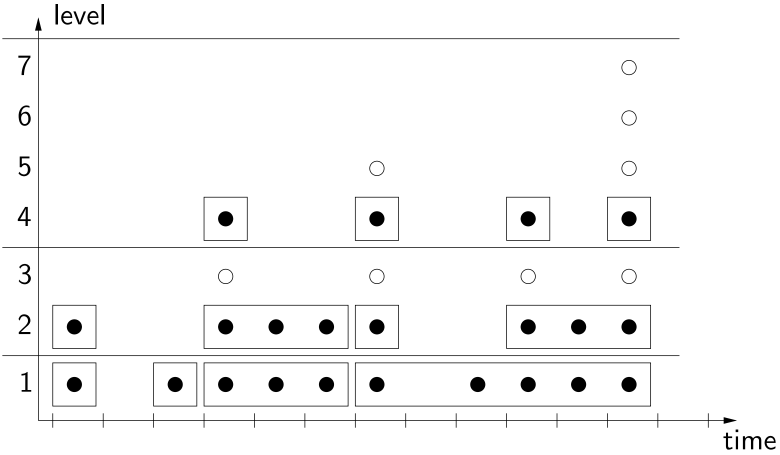
Poly-time approximation-preserving reduction



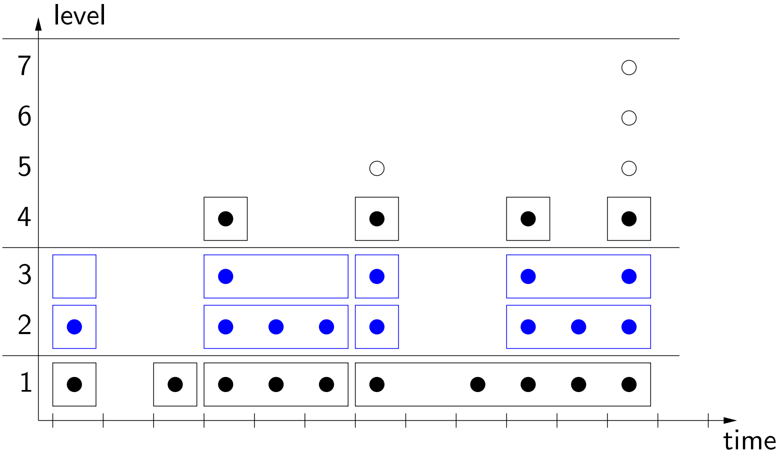
Poly-time approximation-preserving reduction



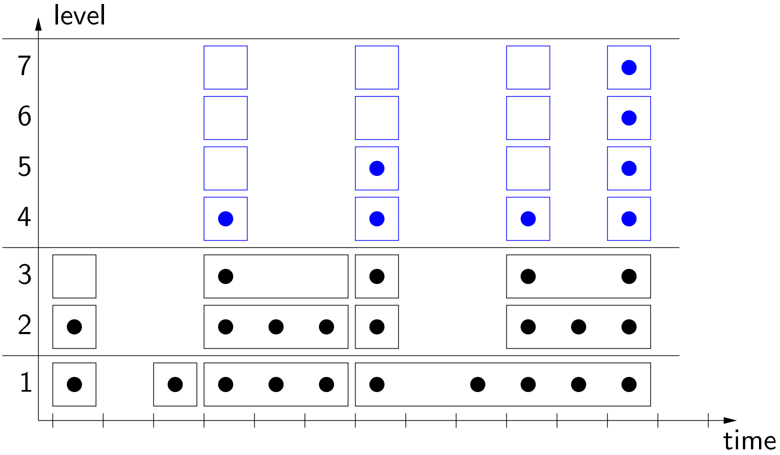
Poly-time approximation-preserving reduction



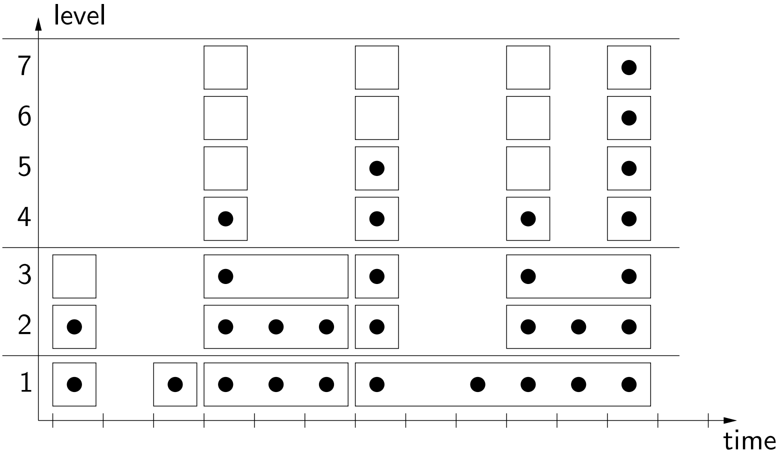
Poly-time approximation-preserving reduction



Poly-time approximation-preserving reduction



Poly-time approximation-preserving reduction



α-competitive Parking Permit $\rightarrow$ 2α-comp. Multi Parking Permit

Poly-time approximation-preserving reduction (2)

α-competitive Parking Permit $\rightarrow$ 2α-comp. Multi Parking Permit

Poly-time approximation-preserving reduction (2)

α-competitive Parking Permit $\rightarrow$ 2α-comp. Multi Parking Permit

- 2-approximation under Interval Model (8-approx. in general)

Poly-time approximation-preserving reduction (2)

α-competitive Parking Permit $\rightarrow$ 2α-comp. Multi Parking Permit

- 2-approximation under Interval Model (8-approx. in general)
- $\mathrm{O}(K)$-competitive deterministic online algorithm
- $\mathrm{O}(\lg K)$-competitive randomized online algorithm

Poly-time exact algorithm

Poly-time exact algorithm

We represent multiset of permits $\rightarrow$ set of tuples: (t, ℓ, k, q)

Poly-time exact algorithm

We represent multiset of permits $\rightarrow$ set of tuples: (t, ℓ, k, q)

- $t \rightarrow$ starting time

Poly-time exact algorithm

We represent multiset of permits $\rightarrow$ set of tuples: (t, ℓ, k, q)

- $t \rightarrow$ starting time
- $\ell \rightarrow$ first level covered

Poly-time exact algorithm

We represent multiset of permits $\rightarrow$ set of tuples: (t, ℓ, k, q)

- $t \rightarrow$ starting time
- $\ell \rightarrow$ first level covered
- $k \rightarrow$ permit type

Poly-time exact algorithm

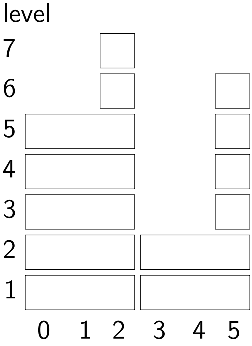
We represent multiset of permits $\rightarrow$ set of tuples: (t, ℓ, k, q)

- $t \rightarrow$ starting time
- $\ell \rightarrow$ first level covered
- $k \rightarrow$ permit type
- $q \rightarrow$ multiplicity

Poly-time exact algorithm

We represent multiset of permits $\rightarrow$ set of tuples: (t, ℓ, k, q)

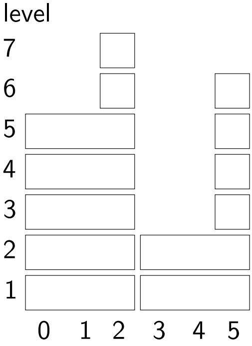
- $t \rightarrow$ starting time
- $\ell \rightarrow$ first level covered
- $k \rightarrow$ permit type
- $q \rightarrow$ multiplicity

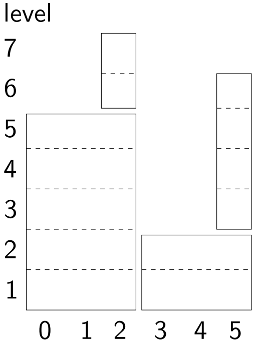


Poly-time exact algorithm

We represent multiset of permits → set of tuples: (t, ℓ, k, q)

- $t \rightarrow$ starting time
- $\ell \rightarrow$ first level covered
- $k \rightarrow$ permit type
- $q \rightarrow$ multiplicity

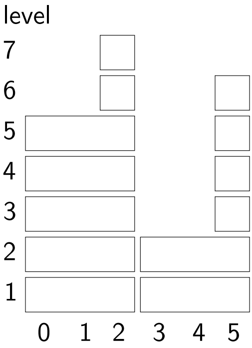


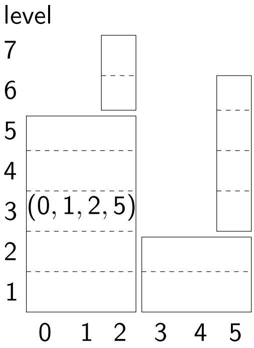


Poly-time exact algorithm

We represent multiset of permits $\rightarrow$ set of tuples: (t, ℓ, k, q)

- $t \rightarrow$ starting time
- $\ell \rightarrow$ first level covered
- $k \rightarrow$ permit type
- $q \rightarrow$ multiplicity

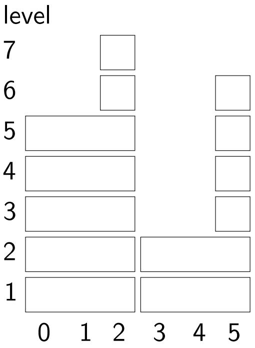


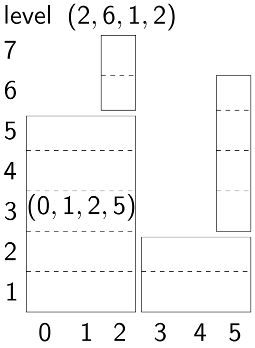


Poly-time exact algorithm

We represent multiset of permits → set of tuples: (t, ℓ, k, q)

- t → starting time
- ℓ → first level covered
- k → permit type
- q → multiplicity

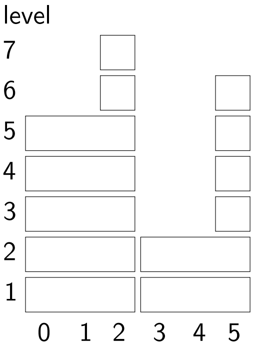


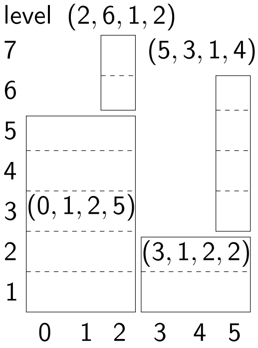


Poly-time exact algorithm

We represent multiset of permits $\rightarrow$ set of tuples: (t, ℓ, k, q)

- $t \rightarrow$ starting time
- $\ell \rightarrow$ first level covered
- $k \rightarrow$ permit type
- $q \rightarrow$ multiplicity





Poly-time exact algorithm (2)

Optimal substructure

Poly-time exact algorithm (2)

Optimal substructure

For $k = 1, \ldots, K \to$ find optimum using only types $1, \ldots, k$

Poly-time exact algorithm (2)

Optimal substructure

For $k = 1, \ldots, K \rightarrow$ find optimum using only types $1, \ldots, k$

- $k = 1 \rightarrow$ trivial

Poly-time exact algorithm (2)

Optimal substructure

For $k = 1, \ldots, K \rightarrow$ find optimum using only types $1, \ldots, k$

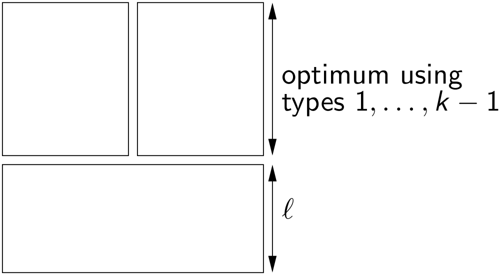
- $k = 1 \rightarrow$ trivial
- suppose we have optimum using types $1, \ldots, k-1$

Poly-time exact algorithm (2)

Optimal substructure

For $k = 1, \ldots, K$ $\rightarrow$ find optimum using only types $1, \ldots, k$

- $k = 1 \rightarrow$ trivial
- suppose we have optimum using types $1, \ldots, k-1$

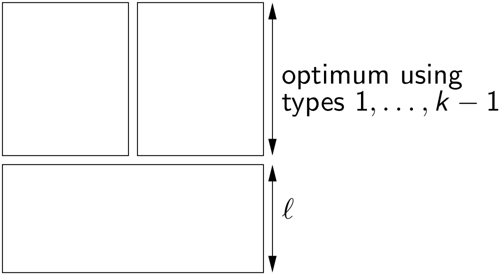


Poly-time exact algorithm (2)

Optimal substructure

For $k = 1, \ldots, K \rightarrow$ find optimum using only types $1, \ldots, k$

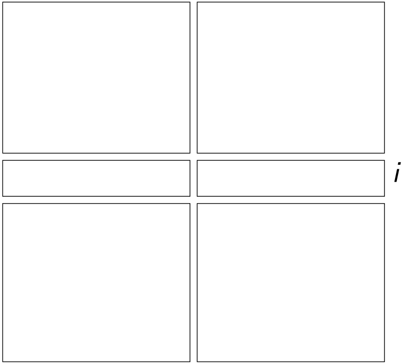
- $k = 1 \rightarrow$ trivial
- suppose we have optimum using types $1, \ldots, k-1$



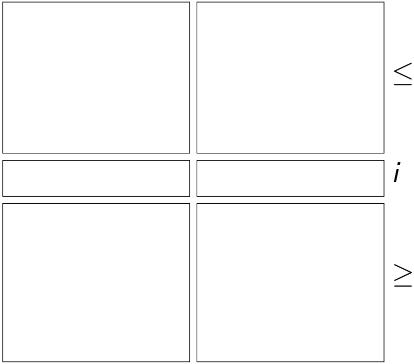
How do we find ℓ?

Poly-time exact algorithm (3)

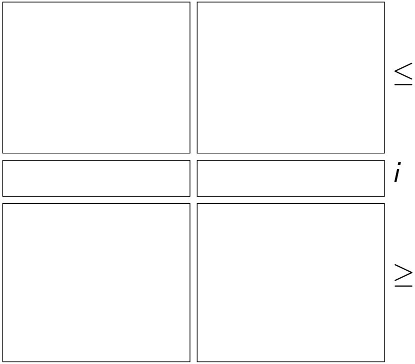
Poly-time exact algorithm (3)



Poly-time exact algorithm (3)

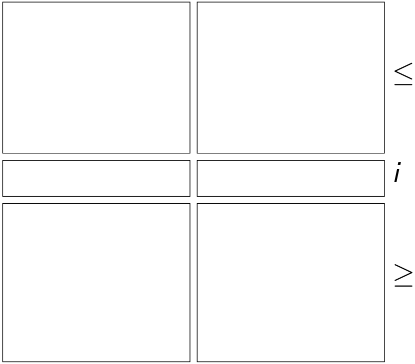


Poly-time exact algorithm (3)



We can use binary search!

Poly-time exact algorithm (3)



We can use binary search!

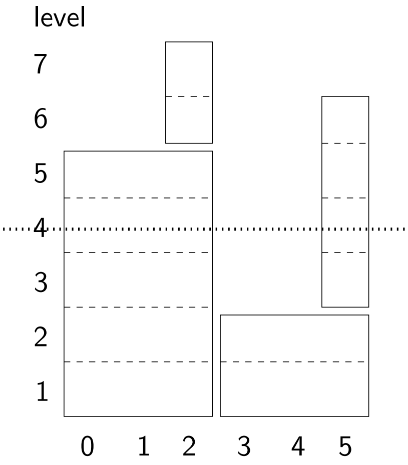
ℓ: highest level s.t. optimum using types $1, \ldots, k-1$ is worse than permit of type k

Poly-time exact algorithm (4)

Our representation helps to guarantee polynomial time

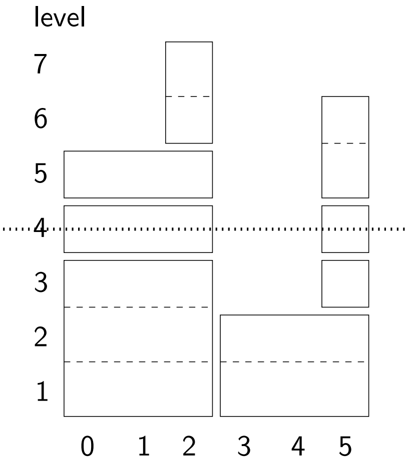
Poly-time exact algorithm (4)

Our representation helps to guarantee polynomial time



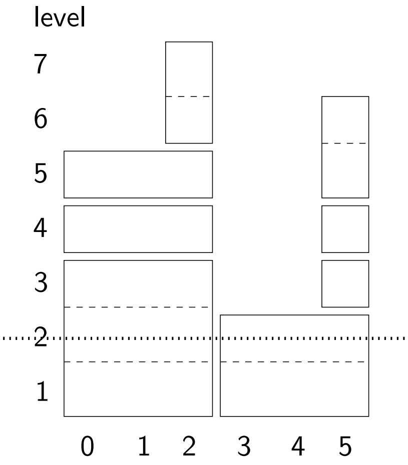
Poly-time exact algorithm (4)

Our representation helps to guarantee polynomial time



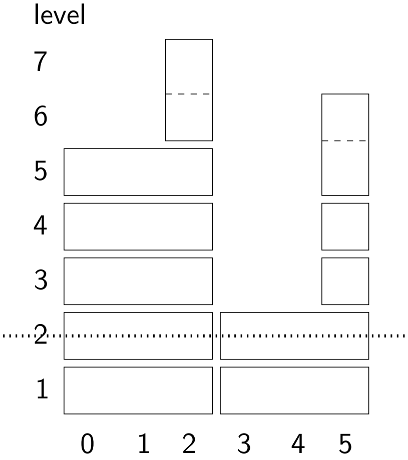
Poly-time exact algorithm (4)

Our representation helps to guarantee polynomial time



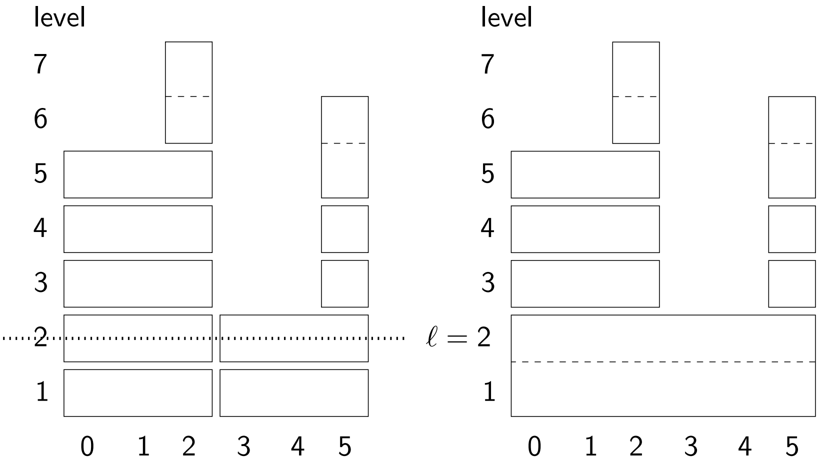
Poly-time exact algorithm (4)

Our representation helps to guarantee polynomial time



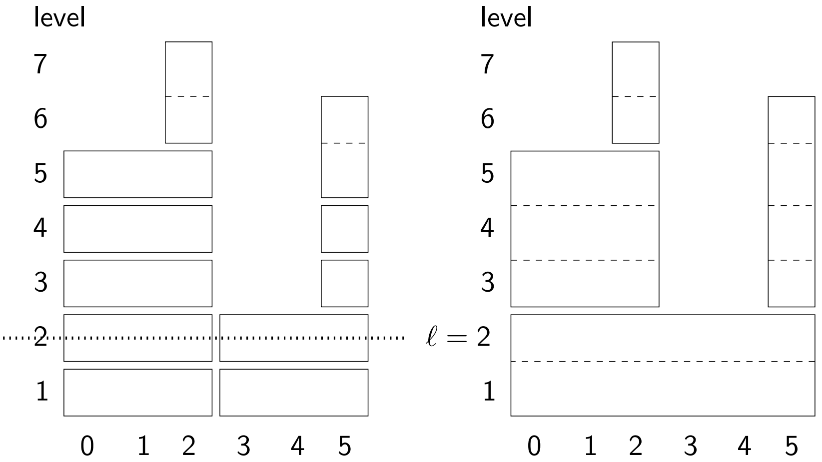
Poly-time exact algorithm (4)

Our representation helps to guarantee polynomial time



Poly-time exact algorithm (4)

Our representation helps to guarantee polynomial time



Consequences on Network Leasing Problems

Consequences on Network Leasing Problems

Multi Parking Permit $\rightarrow$ Steiner Network Leasing Problem

Consequences on Network Leasing Problems

Multi Parking Permit $\rightarrow$ Steiner Network Leasing Problem

- $\mathrm{O}(\lg n)$-approximation
- $\mathrm{O}(\lg K \lg |V|)$-competitive online algorithm

Consequences on Network Leasing Problems

Multi Parking Permit $\rightarrow$ Steiner Network Leasing Problem

- $\mathrm{O}(\lg n)$-approximation
- $\mathrm{O}(\lg K \lg |V|)$-competitive online algorithm

2D Parking Permit Problem

- Hu *et al.*, 2015: pseudo-polynomial algorithms
  - constant approximation
  - $\mathrm{O}(K)$-competitive online algorithm

Consequences on Network Leasing Problems

Multi Parking Permit $\rightarrow$ Steiner Network Leasing Problem

- $\mathrm{O}(\lg n)$-approximation
- $\mathrm{O}(\lg K \lg |V|)$-competitive online algorithm

2D Parking Permit Problem

- Hu *et al.*, 2015: pseudo-polynomial algorithms
  - constant approximation
  - $\mathrm{O}(K)$-competitive online algorithm
- **we got poly-time!** (dynamic programming + binary search)

Consequences on Network Leasing Problems

Multi Parking Permit $\rightarrow$ Steiner Network Leasing Problem

- $\mathrm{O}(\lg n)$-approximation
- $\mathrm{O}(\lg K \lg |V|)$-competitive online algorithm

2D Parking Permit Problem

- Hu *et al.*, 2015: pseudo-polynomial algorithms
  - constant approximation
  - $\mathrm{O}(K)$-competitive online algorithm
- **we got poly-time!** (dynamic programming + binary search)

2D Parking Permit $\rightarrow$ Leasing Buy-at-Bulk Network Design Problem

- $\mathrm{O}(\lg n)$-approximation
- $\mathrm{O}(\lg K \lg |V|)$-competitive online algorithm

Thank you!