Incremental Medians

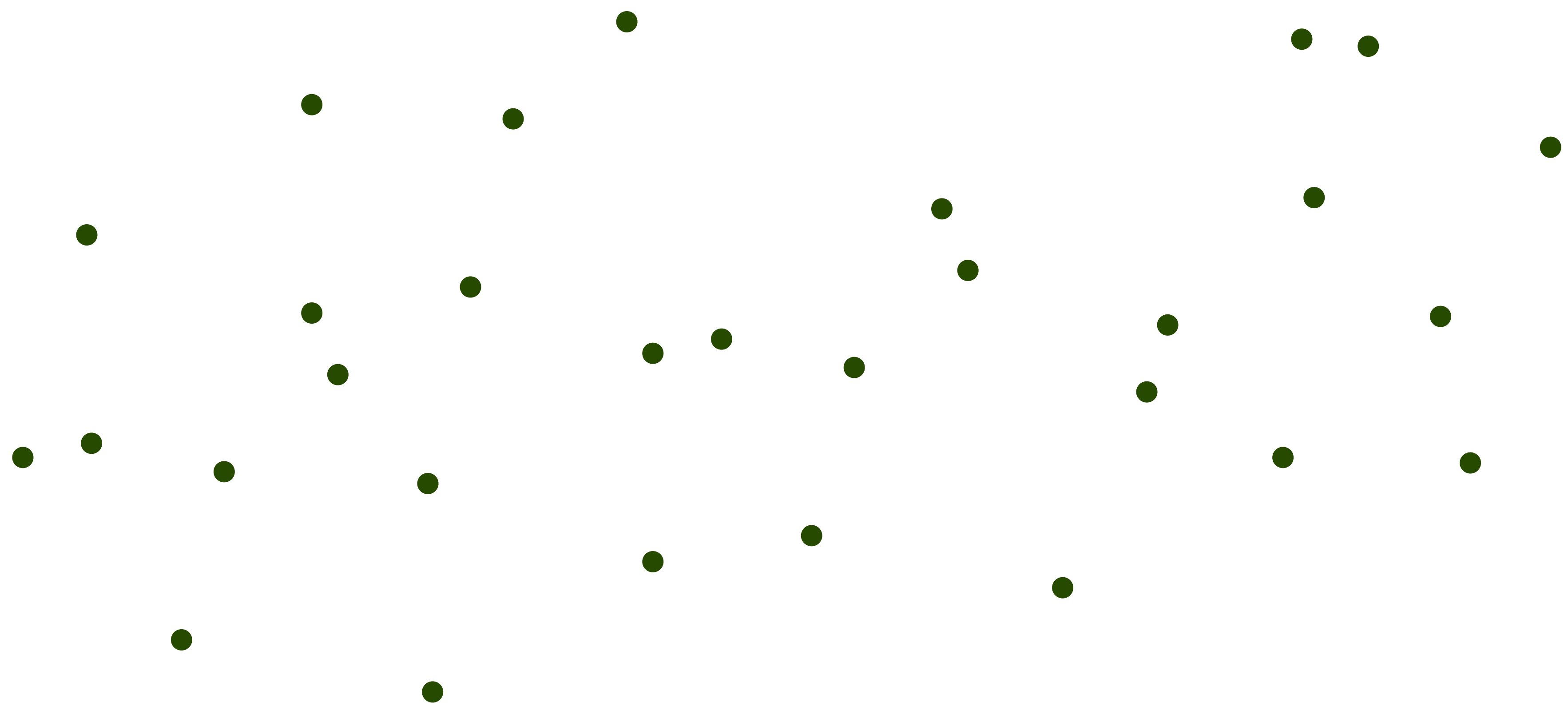
Marek Chrobak - University of California, Riverside

Claire Kenyon - Brown University

John Noga - California State University, Northridge

Neal Young - University of California, Riverside

definition of k-medians problem



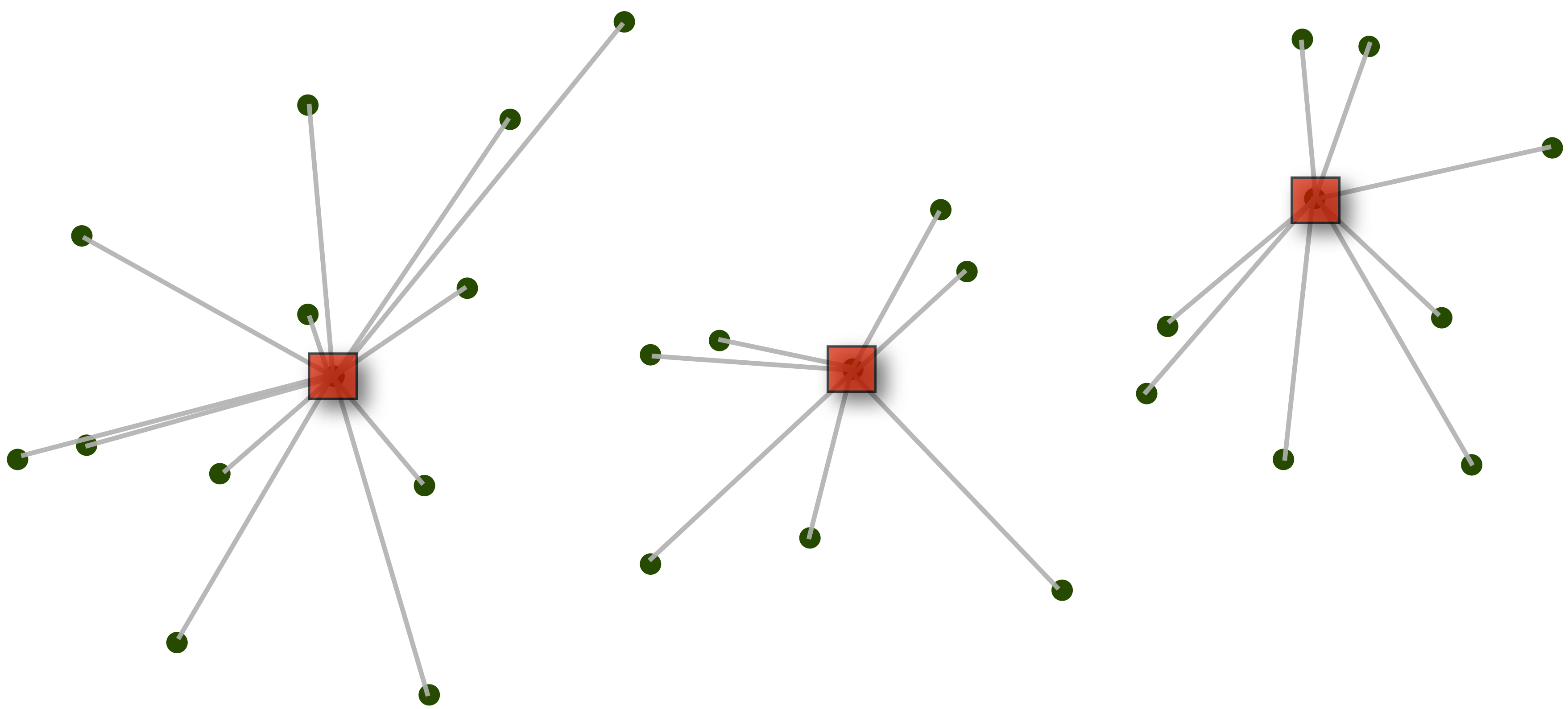
Given customers and facilities, choose k facilities to open.

Minimize sum, over all customers c,

of distance from c to nearest open facility.

(3+ε)-approximation algorithm for metric case. [Araya et al, 2001]

definition of k-medians problem



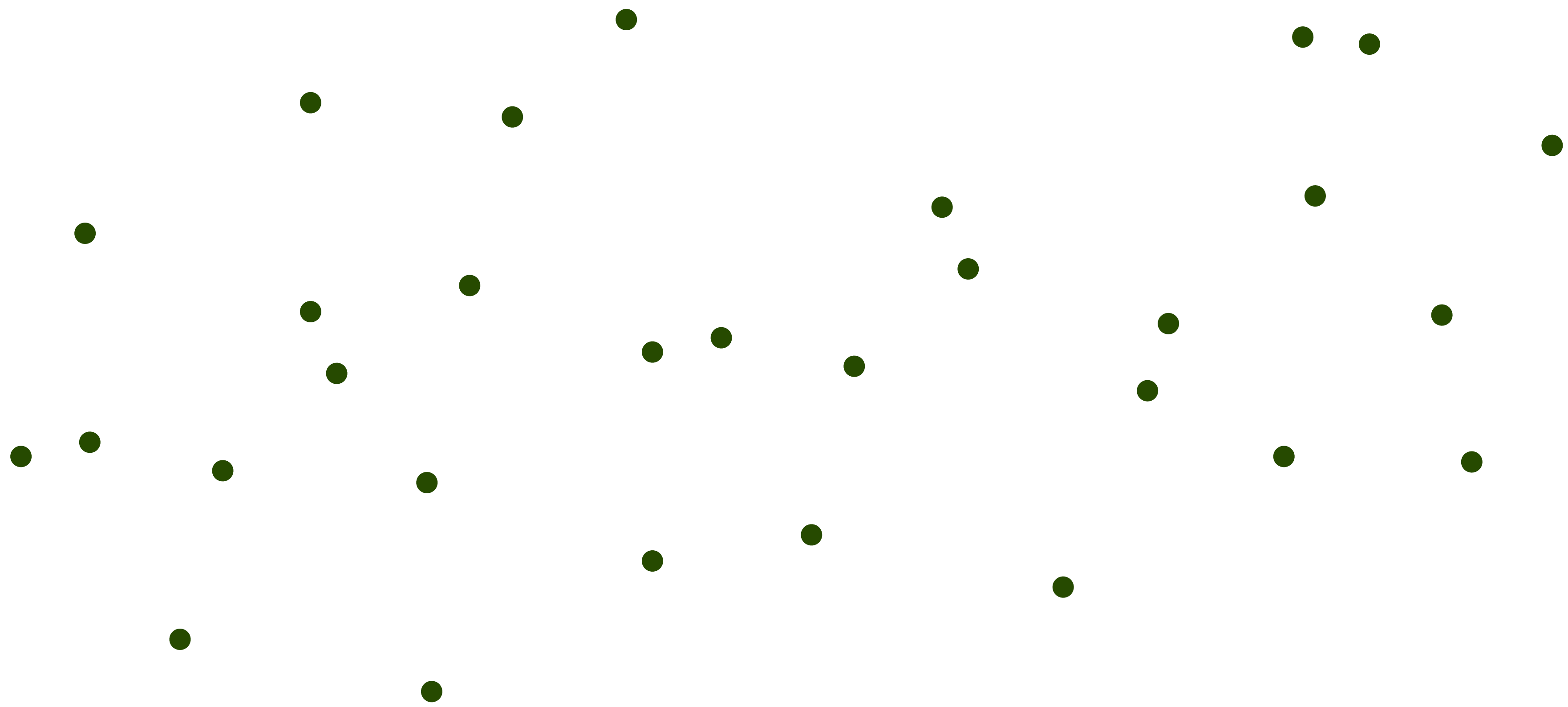
Given customers and facilities, choose k facilities to open.

Minimize sum, over all customers c,

of distance from c to nearest open facility.

(3+ε)-approximation algorithm for metric case. [Araya et al, 2001]

“online” variant

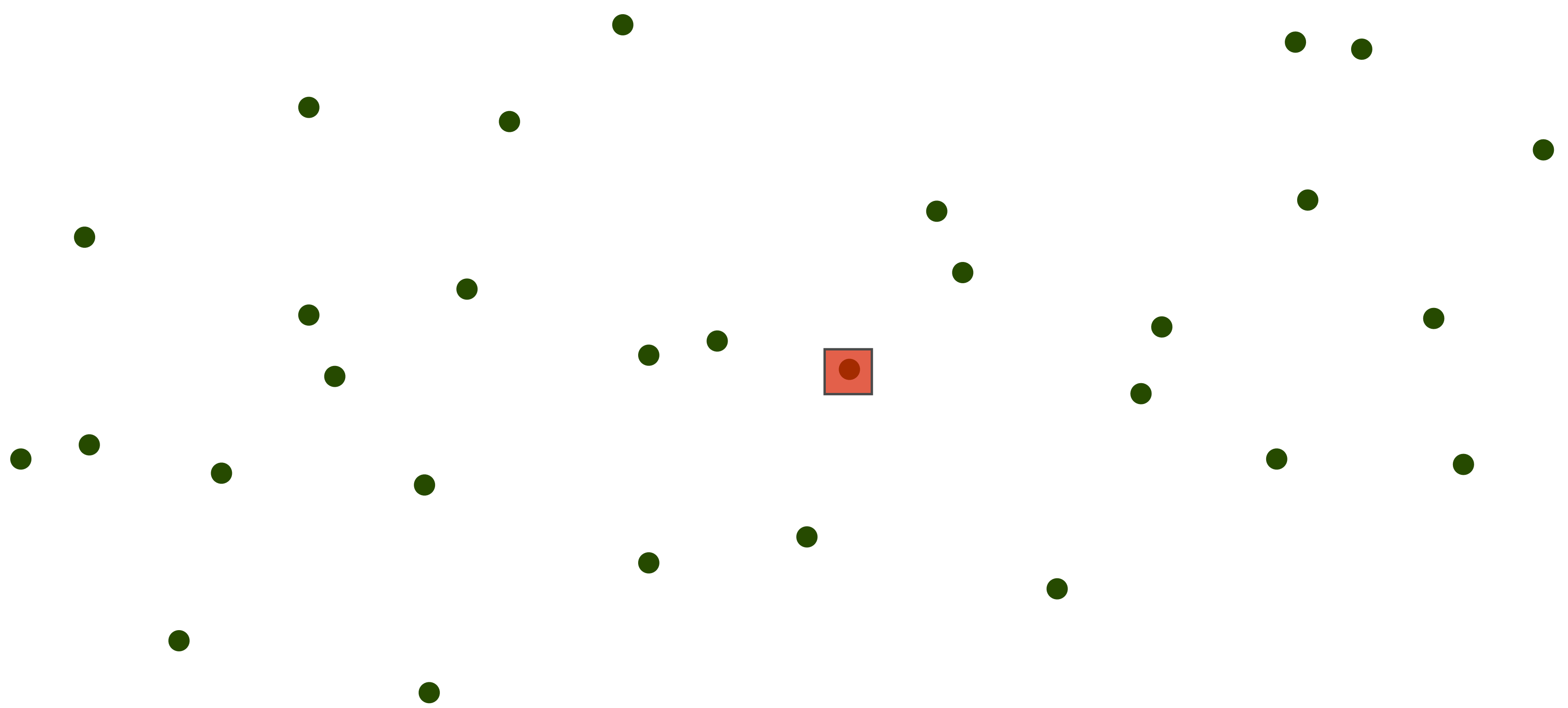


Open facilities one at a time:

Minimize competitive ratio = $\max_{k=1}^{n} \frac{\text{cost}(F_k)}{\text{opt}_k}$

40-competitive algorithm for metric case. [Mettu & Plaxton, 2000]

"online" variant

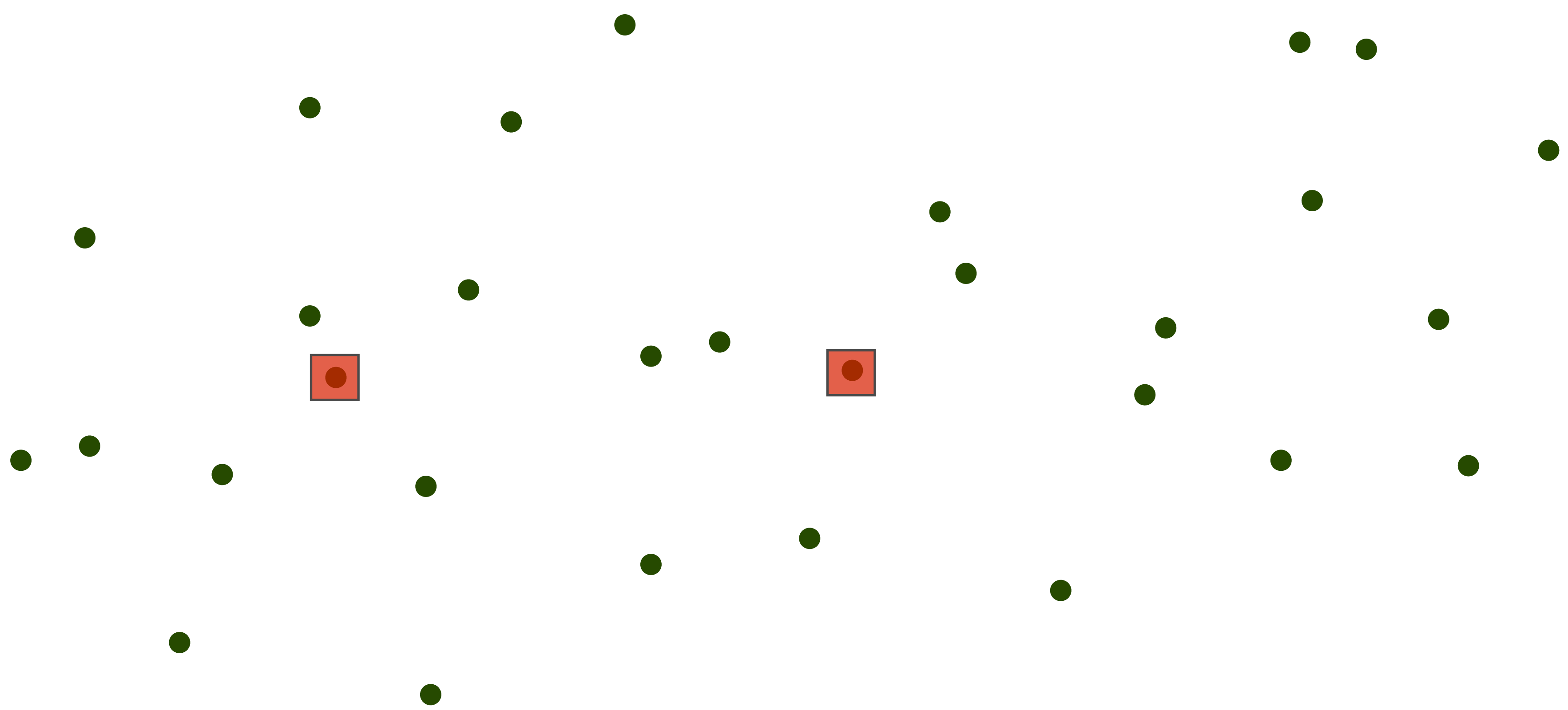


Open facilities one at a time: F_1

Minimize competitive ratio = $\max_{k=1}^{n} \frac{\mathrm{cost}(F_k)}{\mathrm{opt}_k}$

40-competitive algorithm for metric case. [Mettu & Plaxton, 2000]

“online” variant

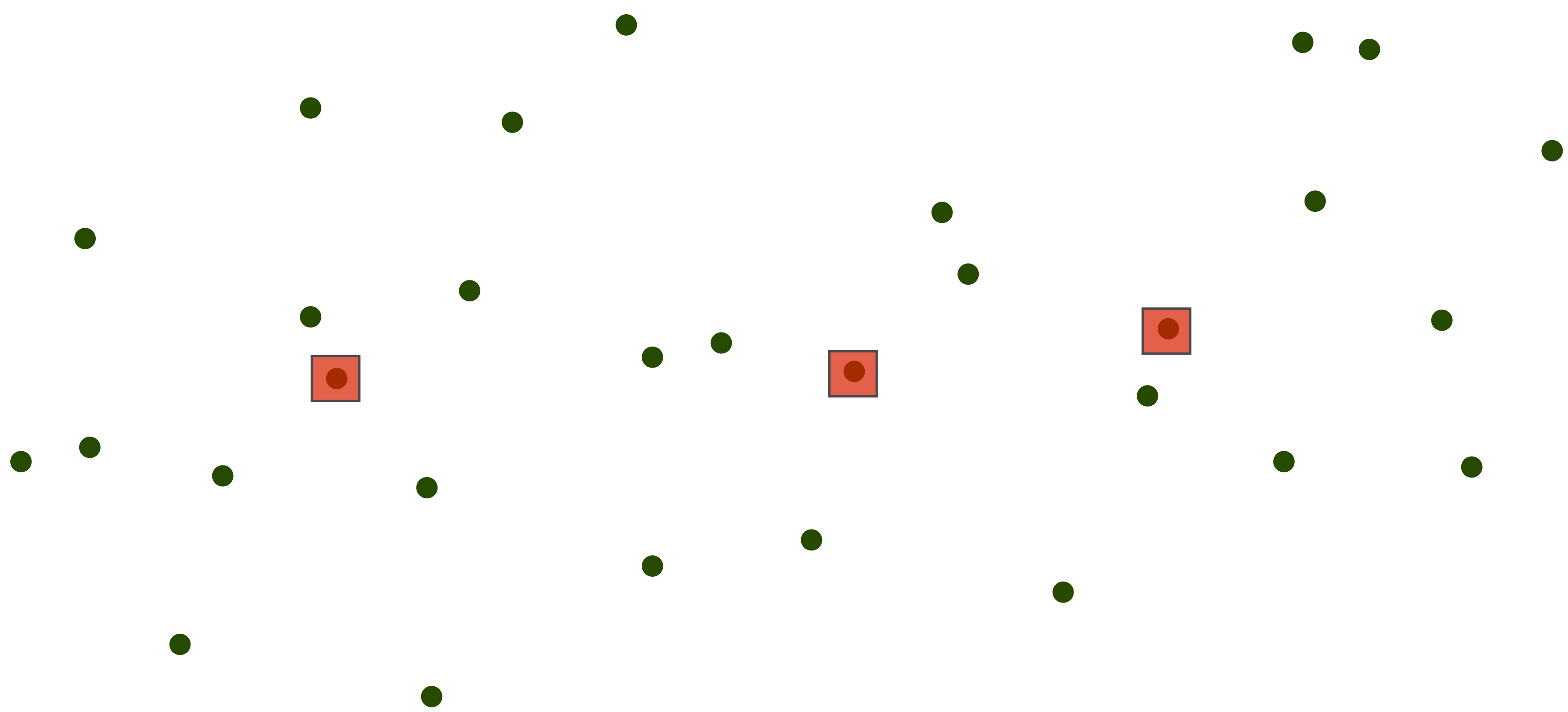


Open facilities one at a time: $F_1 \subset F_2$

Minimize competitive ratio = $\max_{k=1}^{n} \frac{\text{cost}(F_k)}{\text{opt}_k}$

40-competitive algorithm for metric case. [Mettu & Plaxton, 2000]

“online” variant

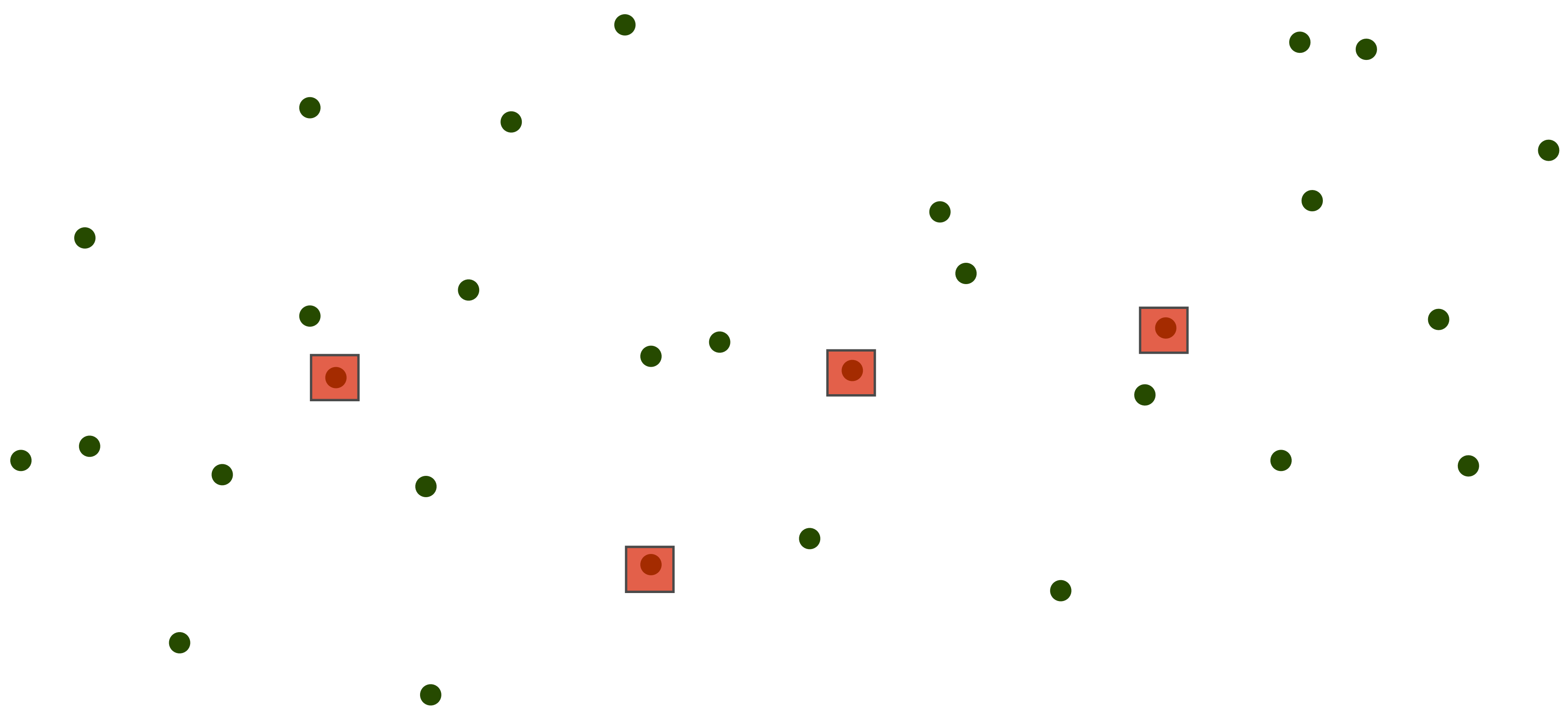


Open facilities one at a time: $F_1 \subset F_2 \subset F_3$

Minimize competitive ratio = $\max_{k=1}^{n} \frac{\text{cost}(F_k)}{\text{opt}_k}$

40-competitive algorithm for metric case. [Mettu & Plaxton, 2000]

"online" variant

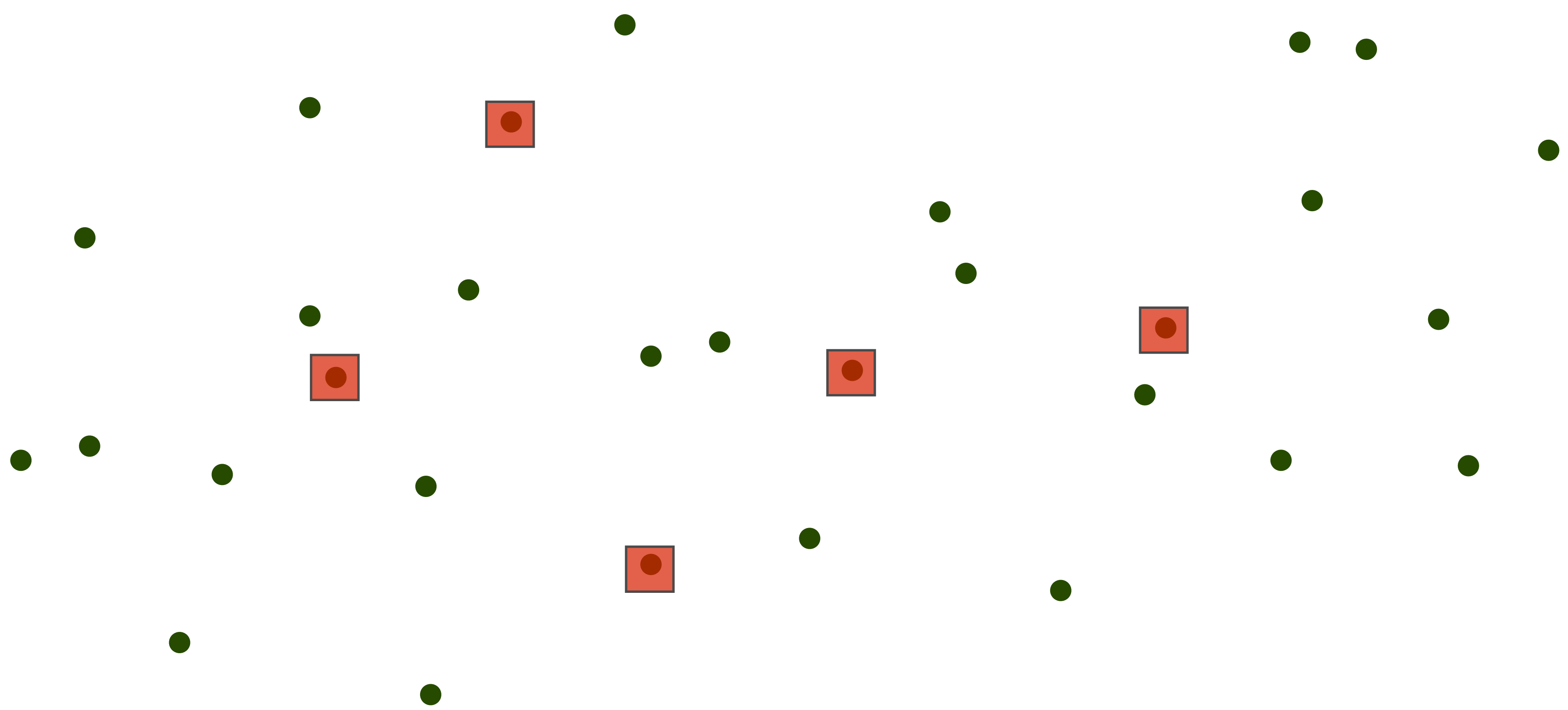


Open facilities one at a time: $F_1 \subset F_2 \subset F_3 \subset F_4$

Minimize competitive ratio = $\max_{k=1}^{n} \frac{\text{cost}(F_k)}{\text{opt}_k}$

40-competitive algorithm for metric case. [Mettu & Plaxton, 2000]

"online" variant

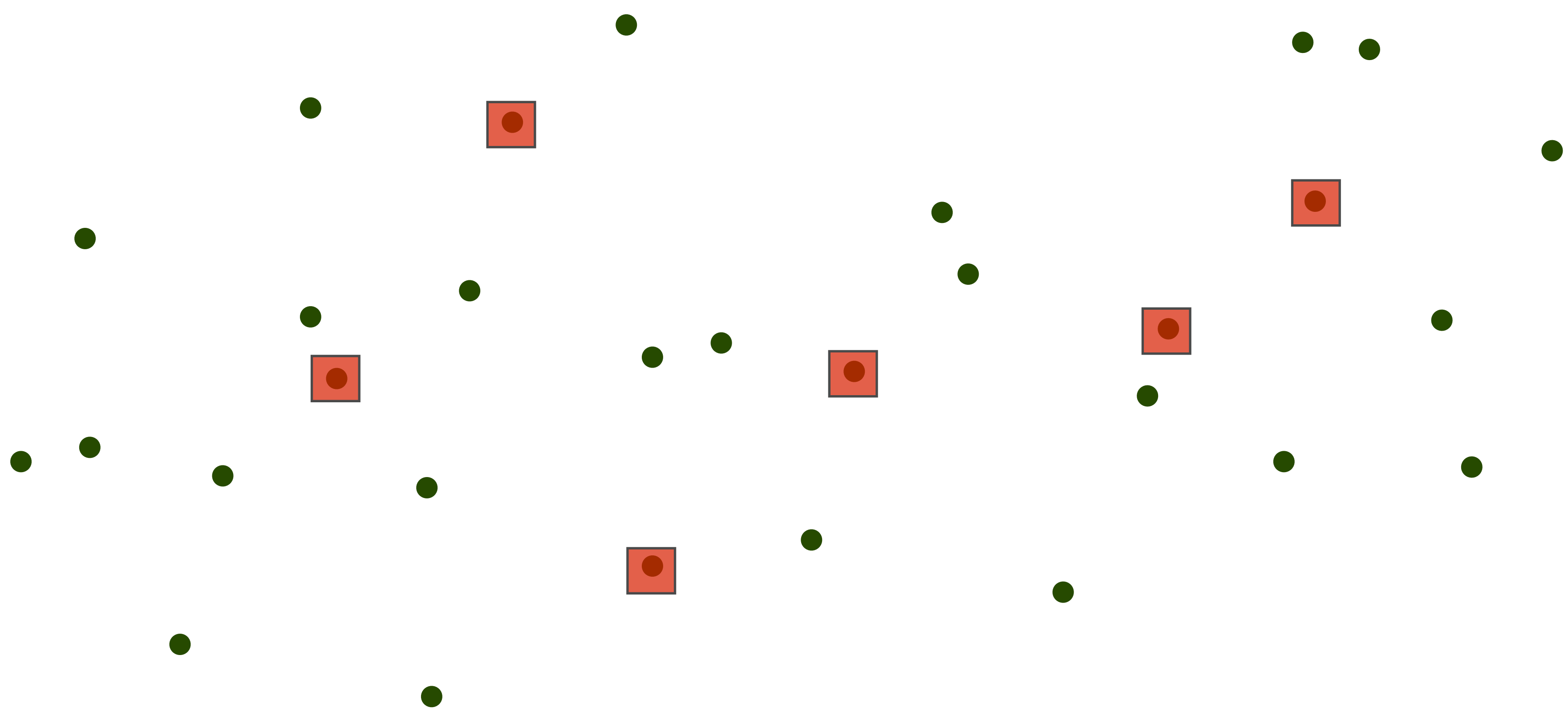


Open facilities one at a time: $F_1 \subset F_2 \subset F_3 \subset F_4 \subset F_5$

Minimize competitive ratio = $\max_{k=1}^{n} \frac{\text{cost}(F_k)}{\text{opt}_k}$

40-competitive algorithm for metric case. [Mettu & Plaxton, 2000]

"online" variant

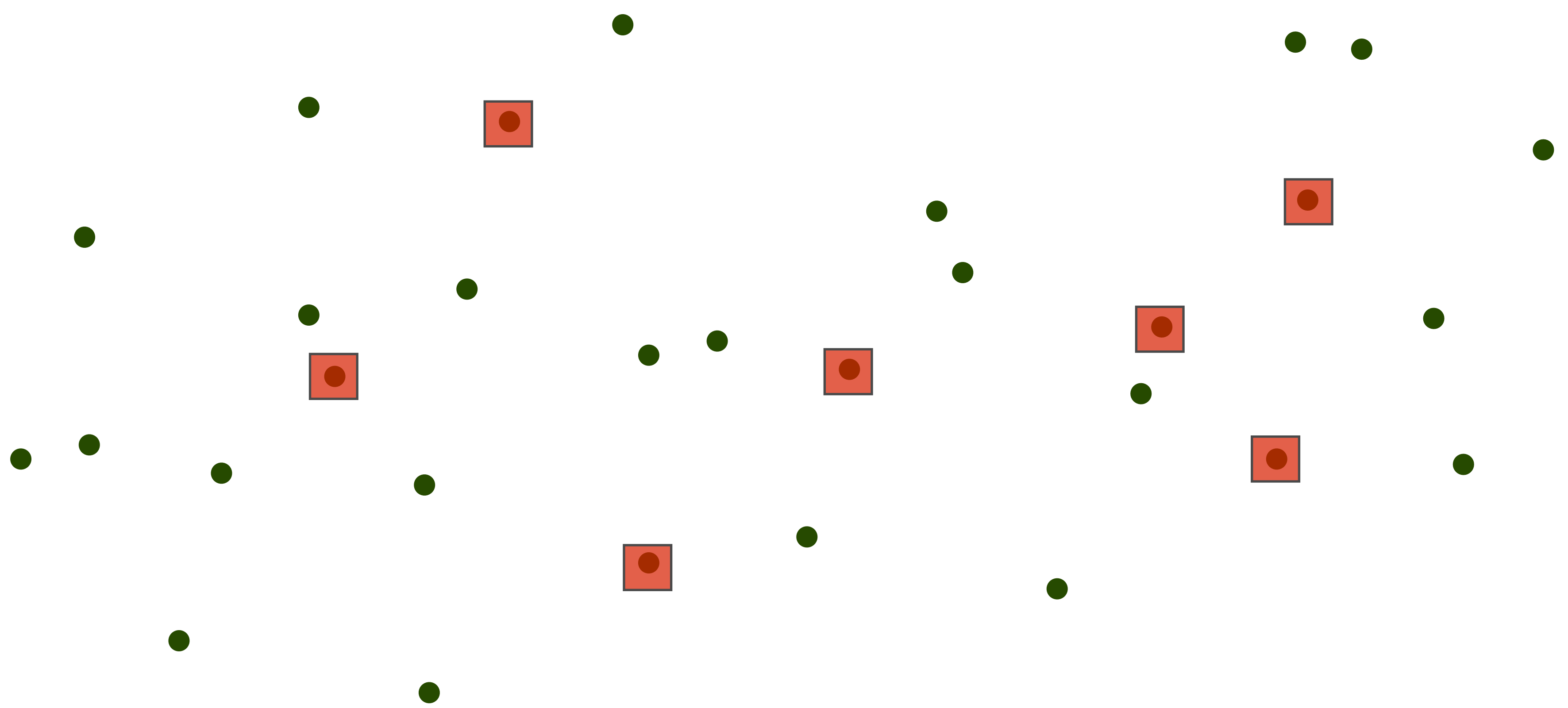


Open facilities one at a time: $F_1 \subset F_2 \subset F_3 \subset F_4 \subset F_5 \subset F_6$

Minimize competitive ratio = $\max_{k=1}^{n} \frac{\text{cost}(F_k)}{\text{opt}_k}$

40-competitive algorithm for metric case. [Mettu & Plaxton, 2000]

“online” variant

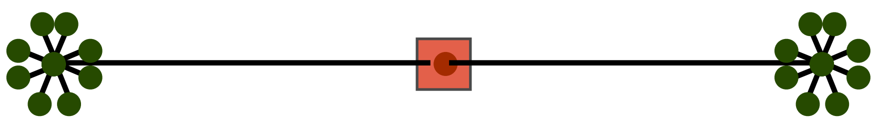


Open facilities one at a time: $F_1 \subset F_2 \subset F_3 \subset F_4 \subset F_5 \subset F_6 \cdots$

Minimize competitive ratio = $\max_{k=1}^{n} \frac{\text{cost}(F_k)}{\text{opt}_k}$

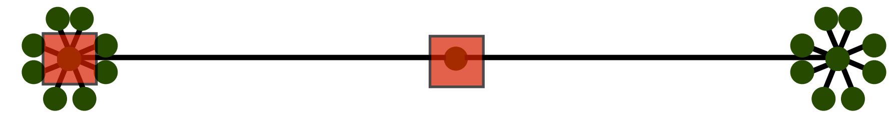
40-competitive algorithm for metric case. [Mettu & Plaxton, 2000]

greedy algorithm not competitive



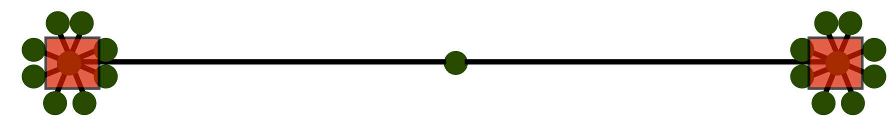
F_1

(greedy)



F_2

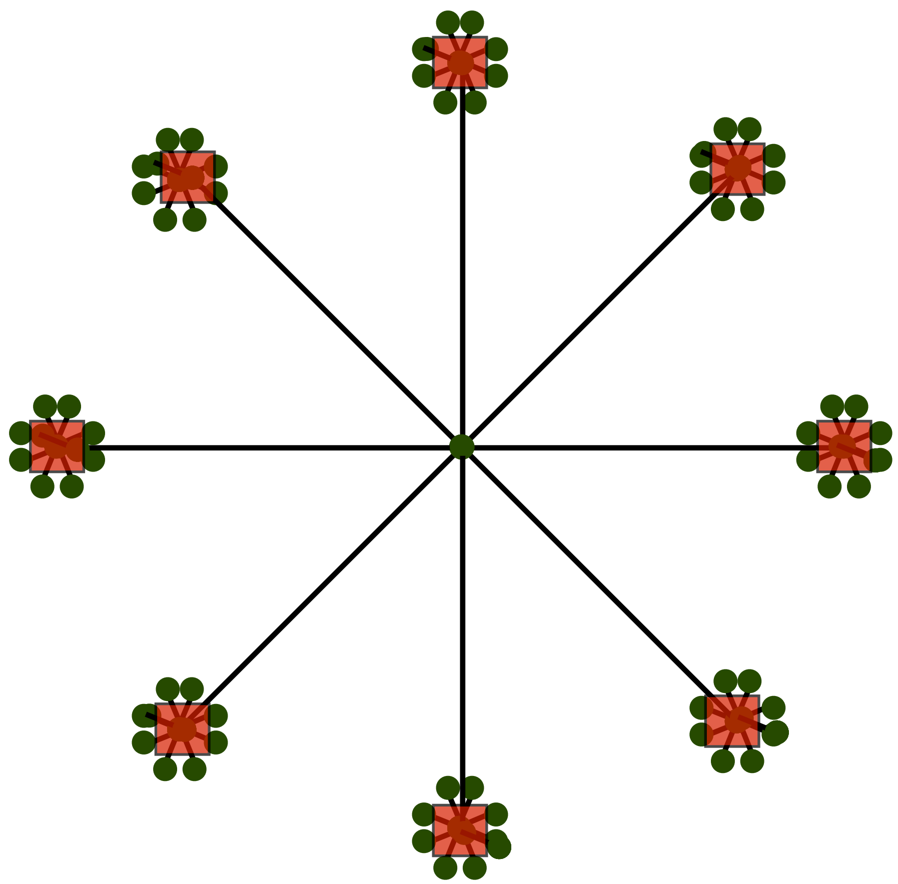
(greedy, cost (n-1)/2)



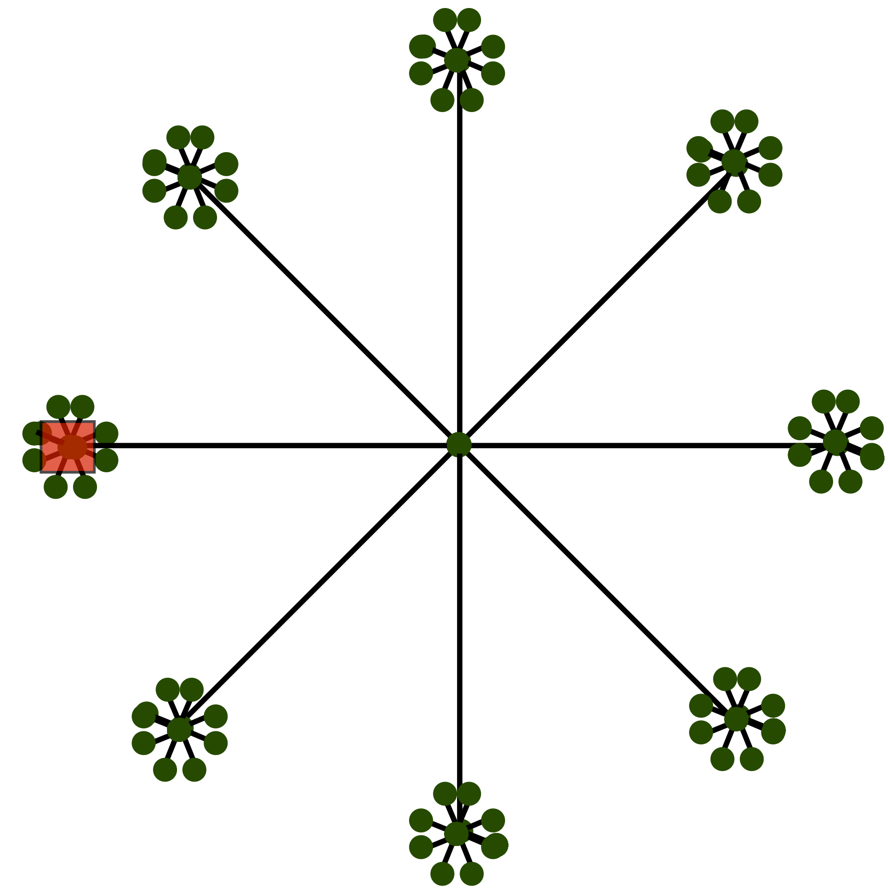
OPT_2

(cost 1)

lower bound of 2 for any deterministic algorithm



F_8 ?



F_1 ?

reverse greedy

1. Let F_n = all facilities
2. For $k = n, n\text{-}1, n\text{-}2, \ldots, 2$ do
3. Choose facility f in F_k to minimize cost(F_k - { f }).
4. $F_{k-1} = F_k$ - { f }.

upper bound: 2log(n)-competitive

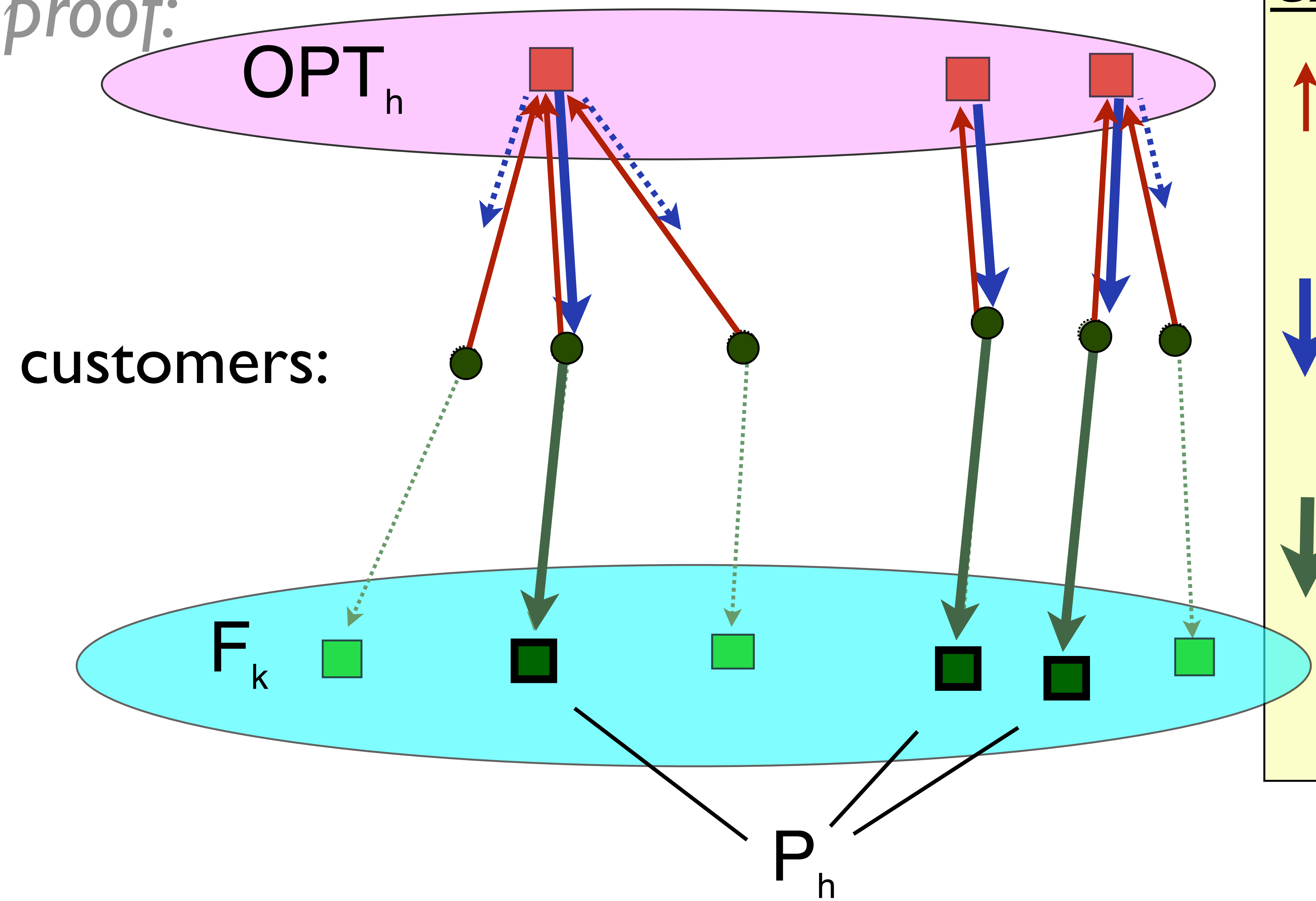
lower bound: Ω(log(n)/log log n) (won't show today)

projection lemma

For any F_k and any h there exists $P_h \subseteq F_k$ of size h

such that $cost(P_h) \leq cost(F_k) + 2\ OPT_h$.

proof:



expected cost

1. *gather on OPT_h*

   cost = OPT_h

2. *return in h groups*

   cost = OPT_h

3. *move groups to F_k*

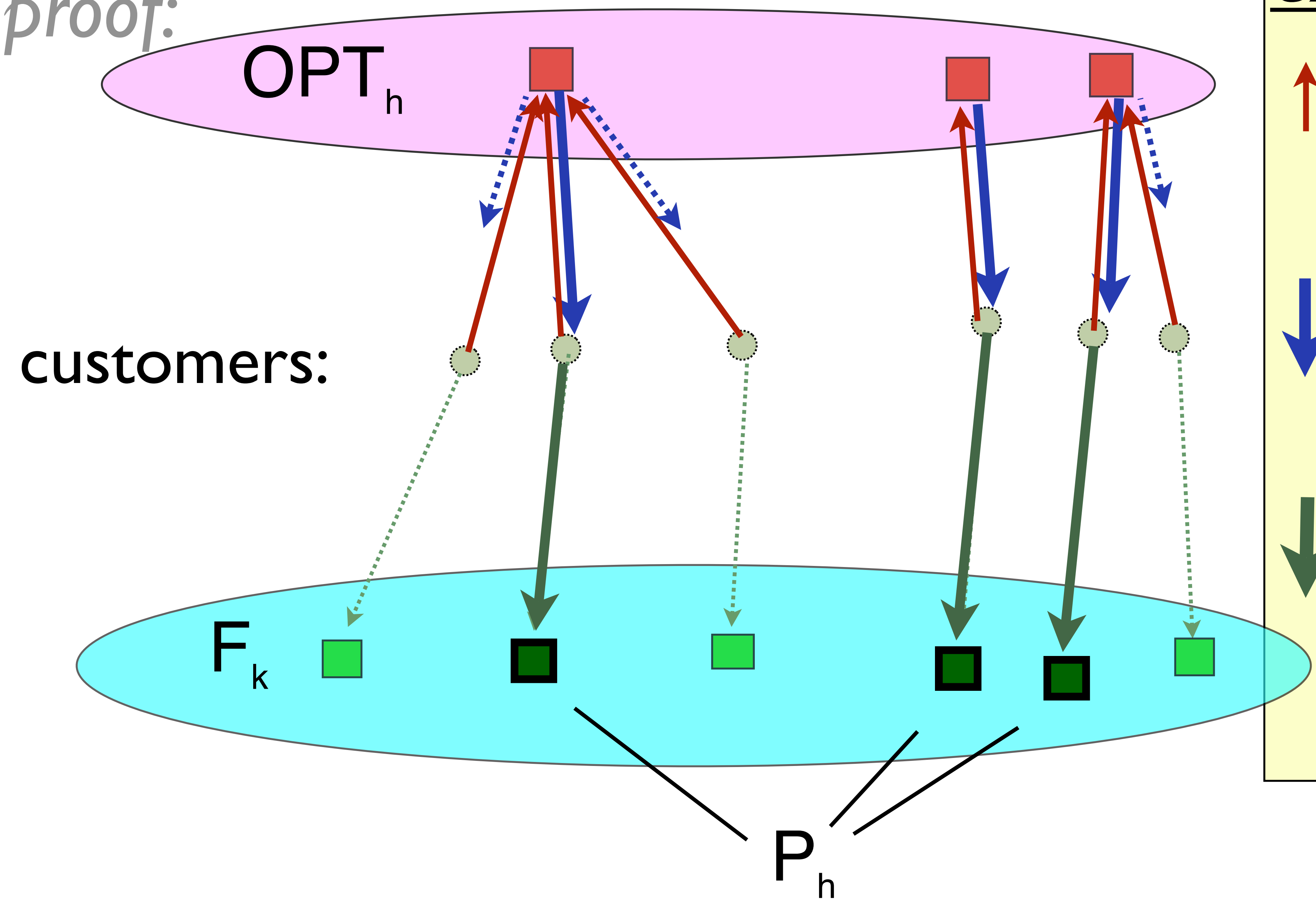
   cost = $cost(F_k)$

projection lemma

For any F_k and any h there exists $P_h \subseteq F_k$ of size h

such that $cost(P_h) \leq cost(F_k) + 2\ OPT_h$.

proof:



expected cost

1. *gather on OPT_h*

   cost = OPT_h

2. *return in h groups*

   cost = OPT_h

3. *move groups to F_k*

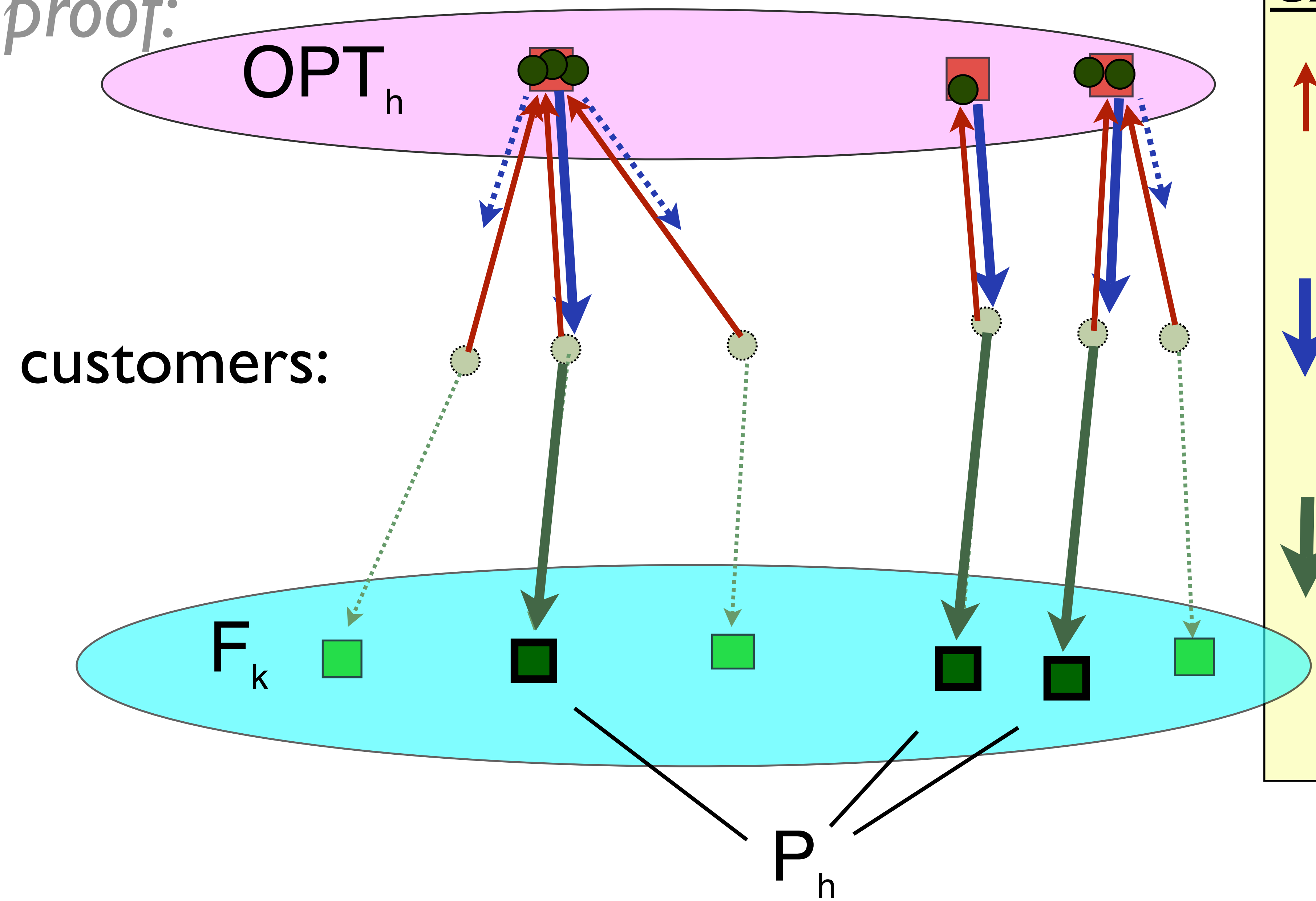
   cost = $cost(F_k)$

projection lemma

For any F_k and any h there exists $P_h \subseteq F_k$ of size h

such that $cost(P_h) \leq cost(F_k) + 2\ OPT_h$.

proof:



expected cost

1. *gather on OPT_h*

   cost = OPT_h

2. *return in h groups*

   cost = OPT_h

3. *move groups to F_k*

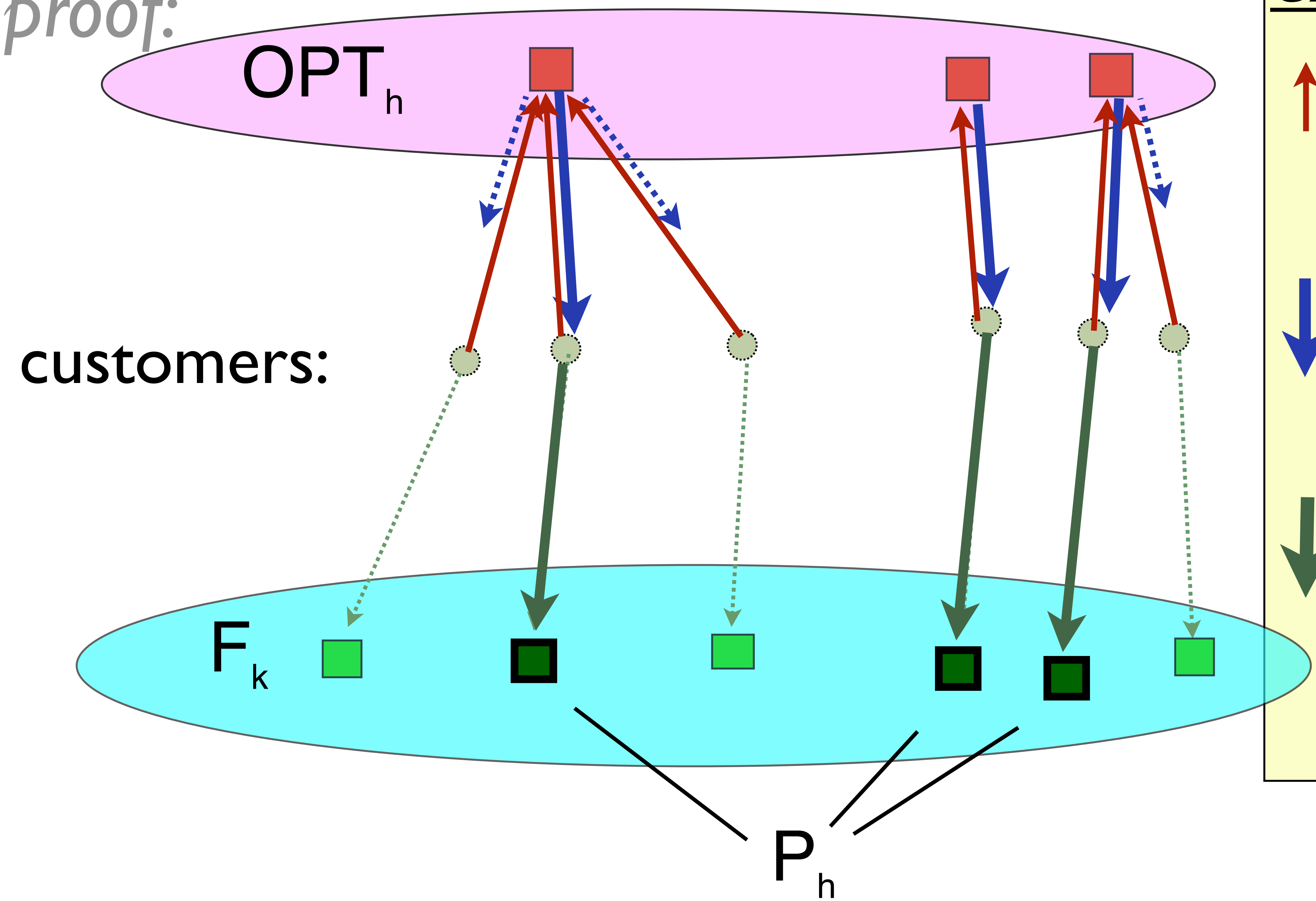
   cost = $cost(F_k)$

projection lemma

For any F_k and any h there exists $P_h \subseteq F_k$ of size h

such that $cost(P_h) \leq cost(F_k) + 2\ OPT_h$.

proof:



expected cost

↑ 1. *gather on OPT_h*

cost = OPT_h

↓ 2. *return in h groups*

cost = OPT_h

↓ 3. *move groups to F_k*

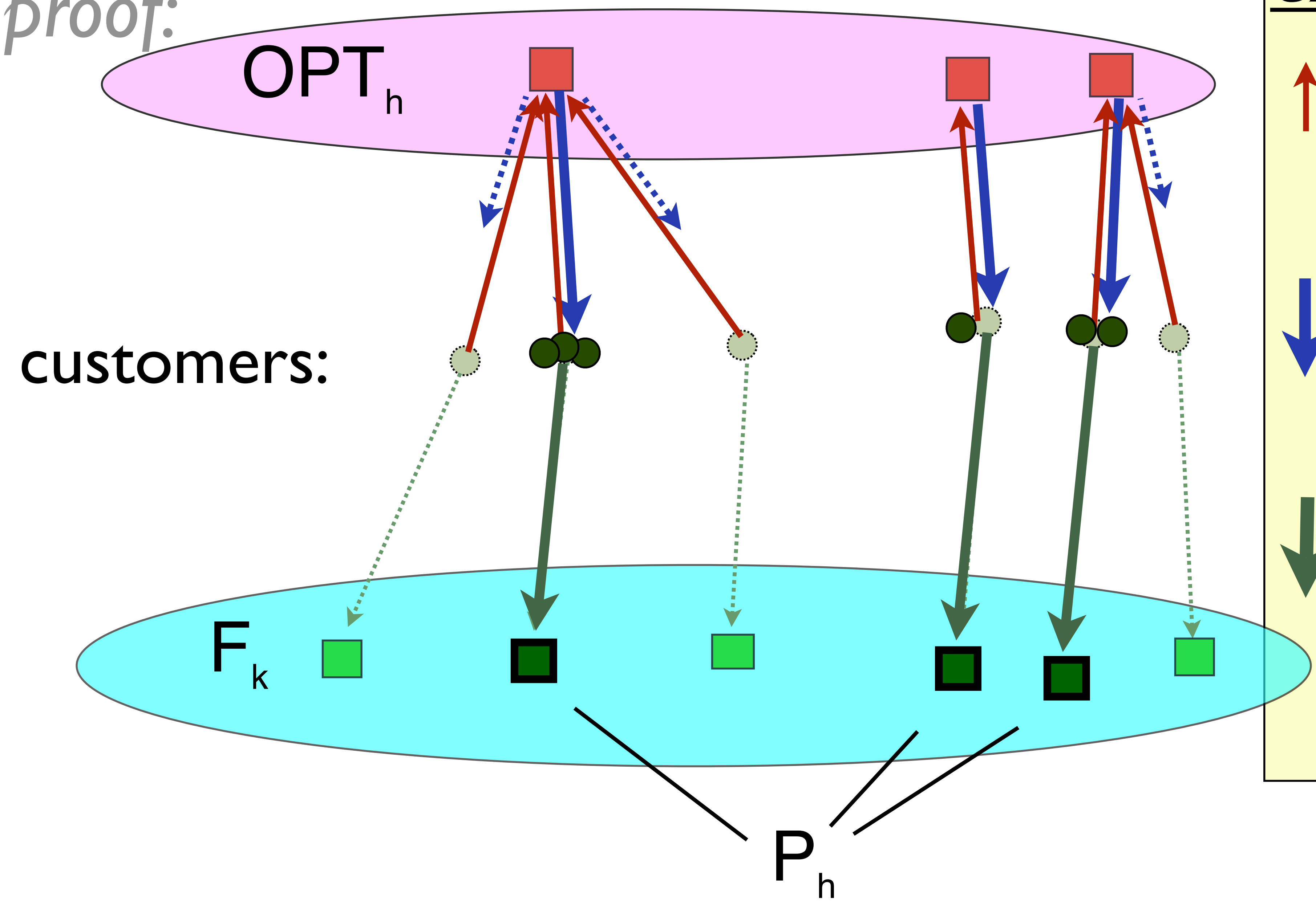
cost = $cost(F_k)$

projection lemma

For any F_k and any h there exists $P_h \subseteq F_k$ of size h

such that $cost(P_h) \leq cost(F_k) + 2\ OPT_h$.

proof:



expected cost

1. *gather on OPT_h*

   cost = OPT_h

2. *return in h groups*

   cost = OPT_h

3. *move groups to F_k*

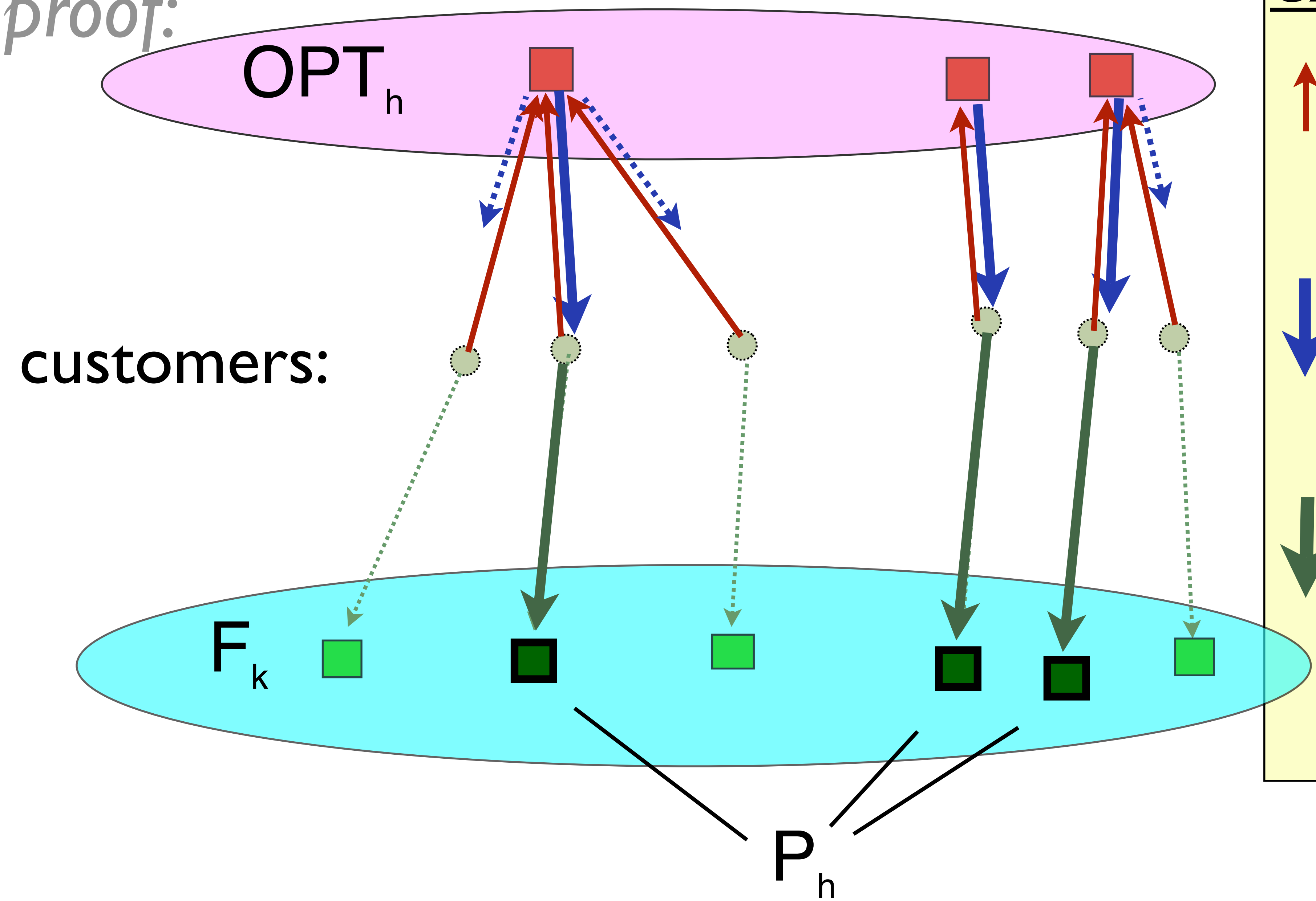
   cost = $cost(F_k)$

projection lemma

For any F_k and any h there exists $P_h \subseteq F_k$ of size h

such that $cost(P_h) \leq cost(F_k) + 2\,OPT_h$.

proof:



expected cost

1. *gather on OPT_h*

   cost = OPT_h

2. *return in h groups*

   cost = OPT_h

3. *move groups to F_k*

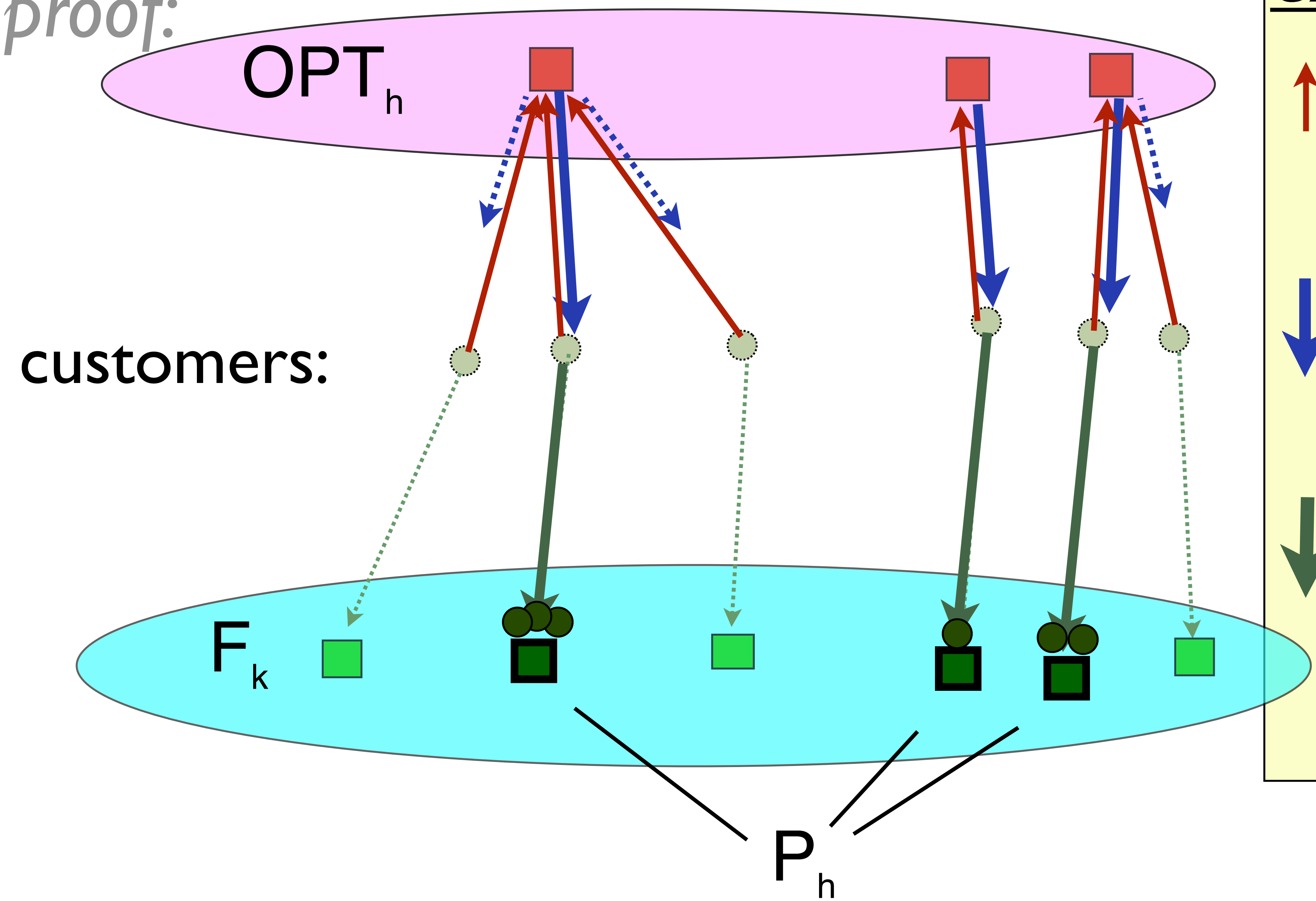
   cost = $cost(F_k)$

projection lemma

For any F_k *and any* h *there exists* $P_h \subseteq F_k$ *of size* h

such that $cost(P_h) \;\leq\; cost(F_k) + 2\; OPT_h$.

proof:



expected cost

1. *gather on* OPT_h

   cost = OPT_h

2. *return in* h *groups*

   cost = OPT_h

3. *move groups to* F_k

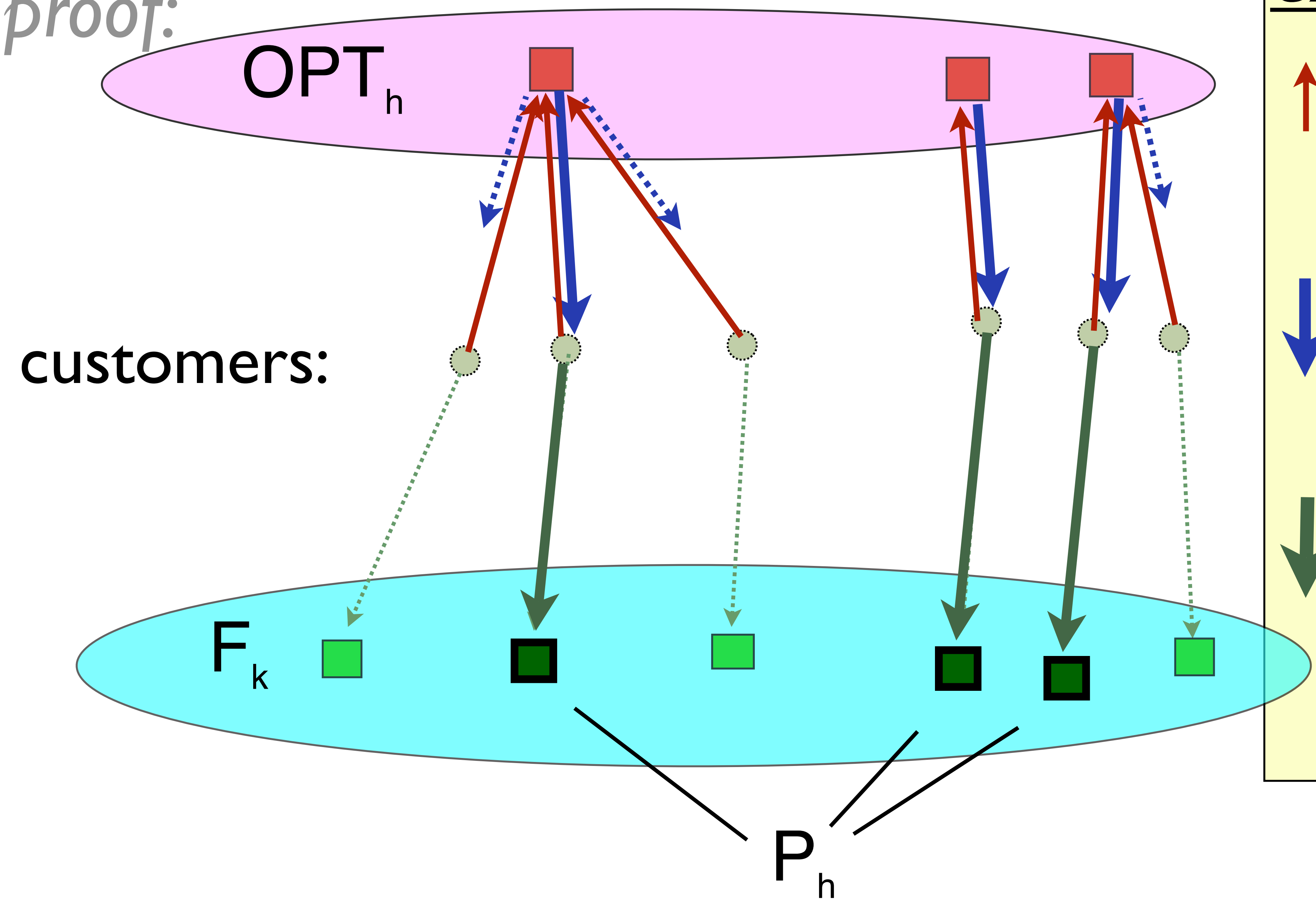
   cost = $cost(F_k)$

projection lemma

For any F_k and any h there exists $P_h \subseteq F_k$ of size h

such that $cost(P_h) \leq cost(F_k) + 2\,OPT_h$.

proof:



expected cost

↑ 1. *gather on OPT_h*

cost = OPT_h

↓ 2. *return in h groups*

cost = OPT_h

↓ 3. *move groups to F_k*

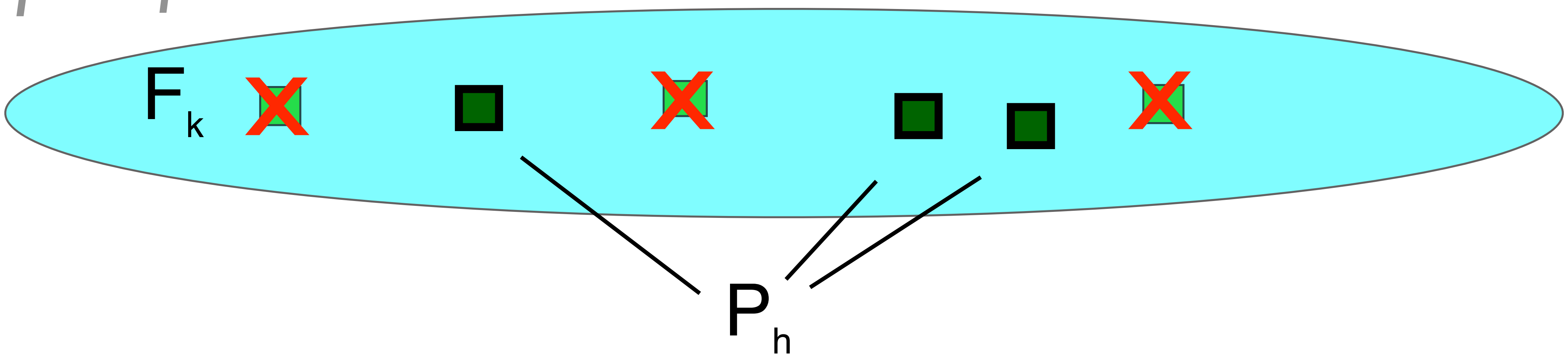
cost = $cost(F_k)$

corollary

For any F_k and any h there exists f in F_k with

$$cost(F_k - \{ f \}) \leq cost(F_k) + 2\ OPT_h / (k-h).$$

proof:



Projection lemma ⇒ removing *all* k-h facilities in $F_k - P_h$ would increase cost by at most $2\ OPT_h$...

So there must be *one* to remove that increases cost by at most $2\ OPT_h$ over k-h.

corollary: reverse greedy is 2 H_n-competitive

1. Showed there exists f in F_k with

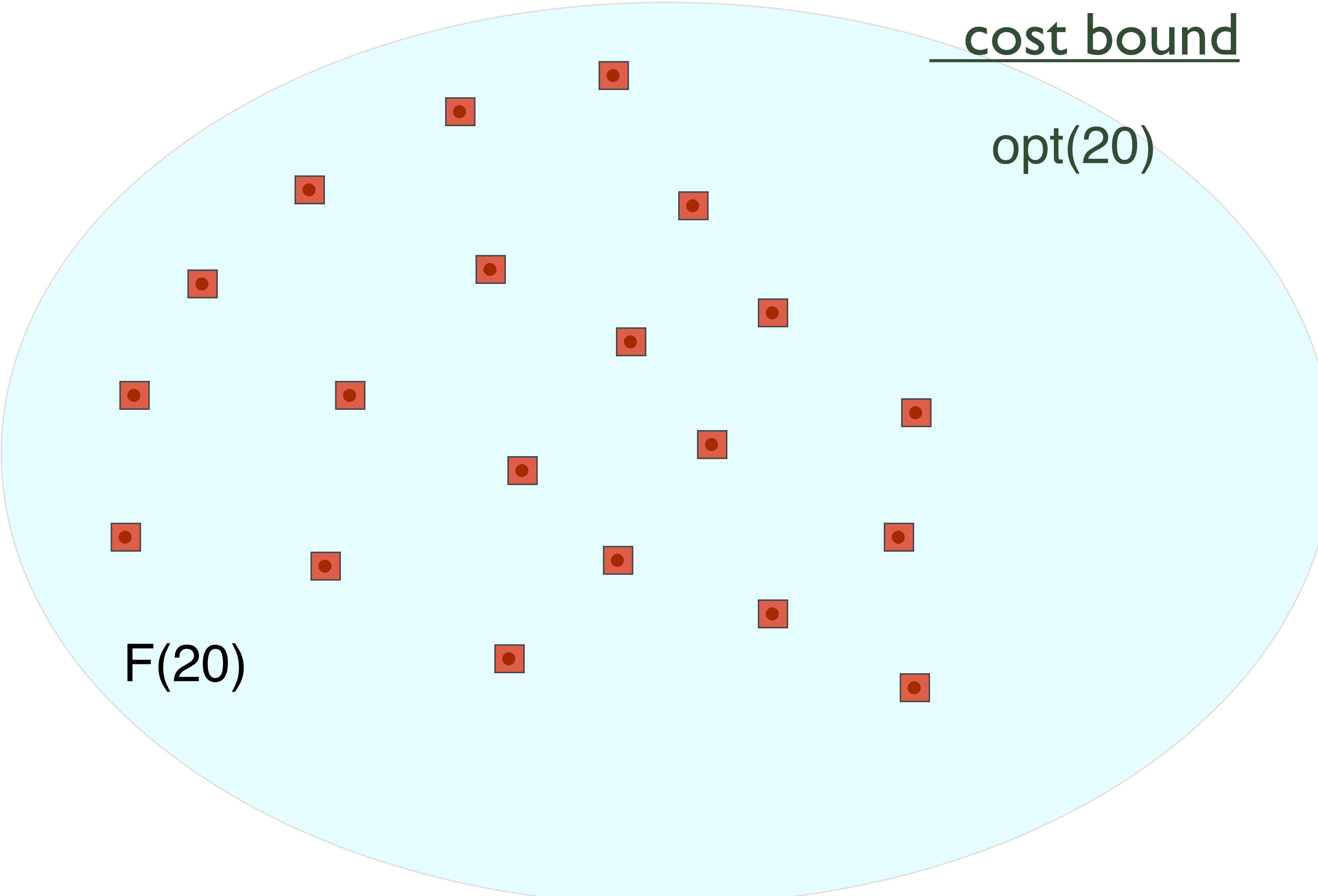
$$\mathrm{cost}(F_k - \{f\}) \le \mathrm{cost}(F_k) + \frac{2\,\mathrm{OPT}_h}{k-h}.$$

2. Thus (taking $k = n, n-1, n-2, ..., h+1$),

$$\mathrm{cost}(F_h) \le 2\,\mathrm{OPT}_h \left[\frac{1}{n-h} + \frac{1}{n-1-h} + \cdots + \frac{1}{1}\right].$$

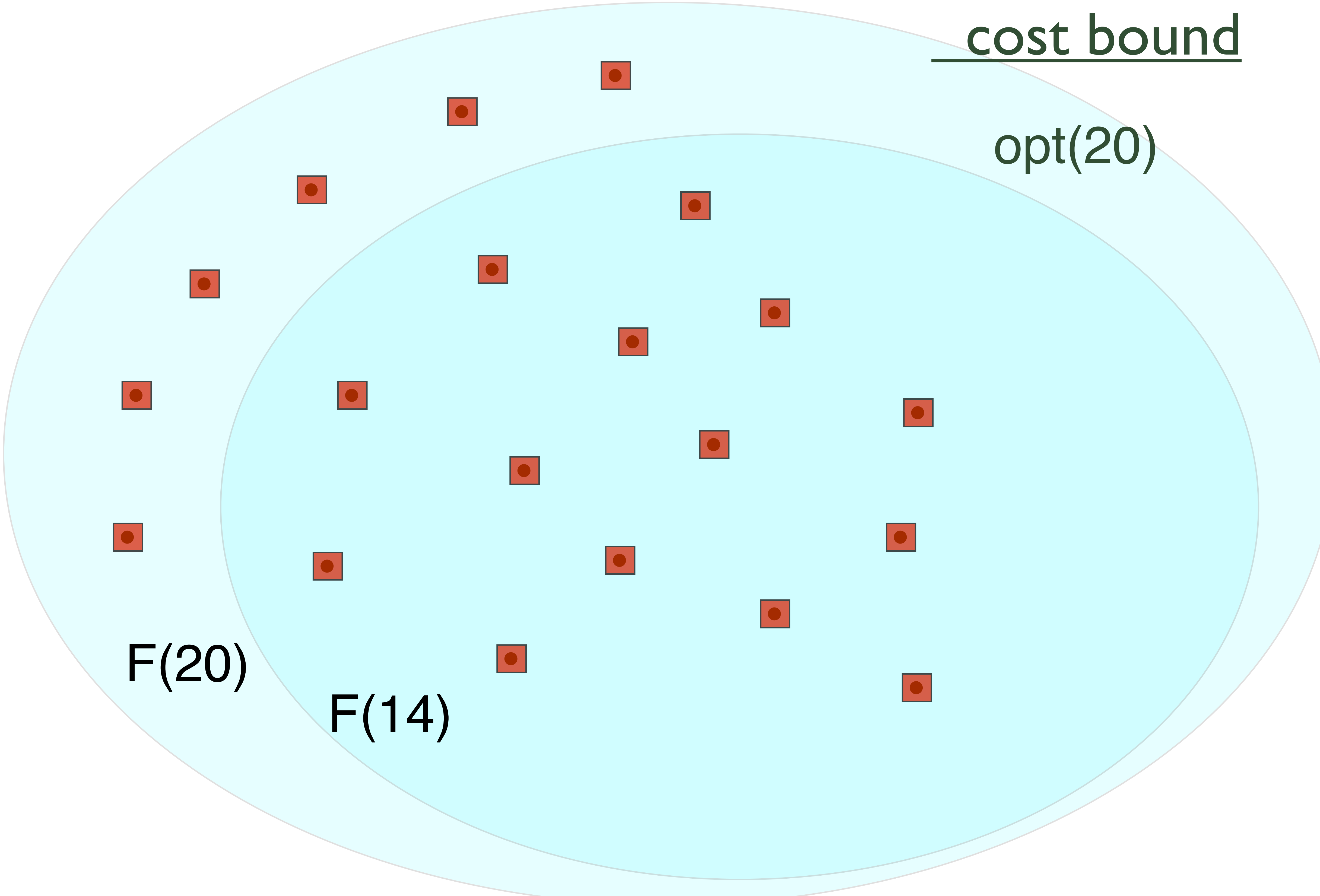
better algorithm -- "batch" reverse greedy

Project down at a few well-chosen k's:



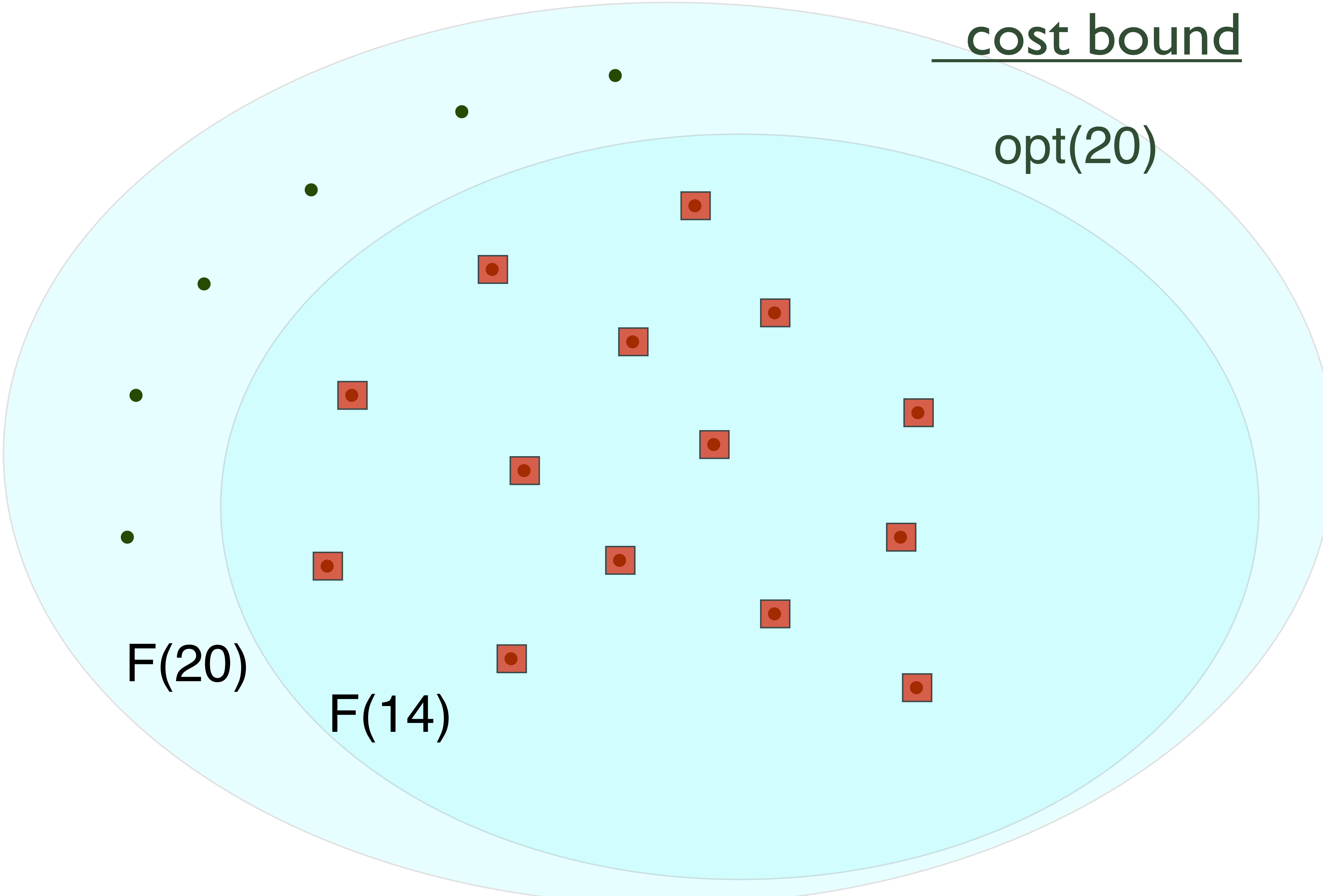
better algorithm -- "batch" reverse greedy

Project down at a few well-chosen k's:



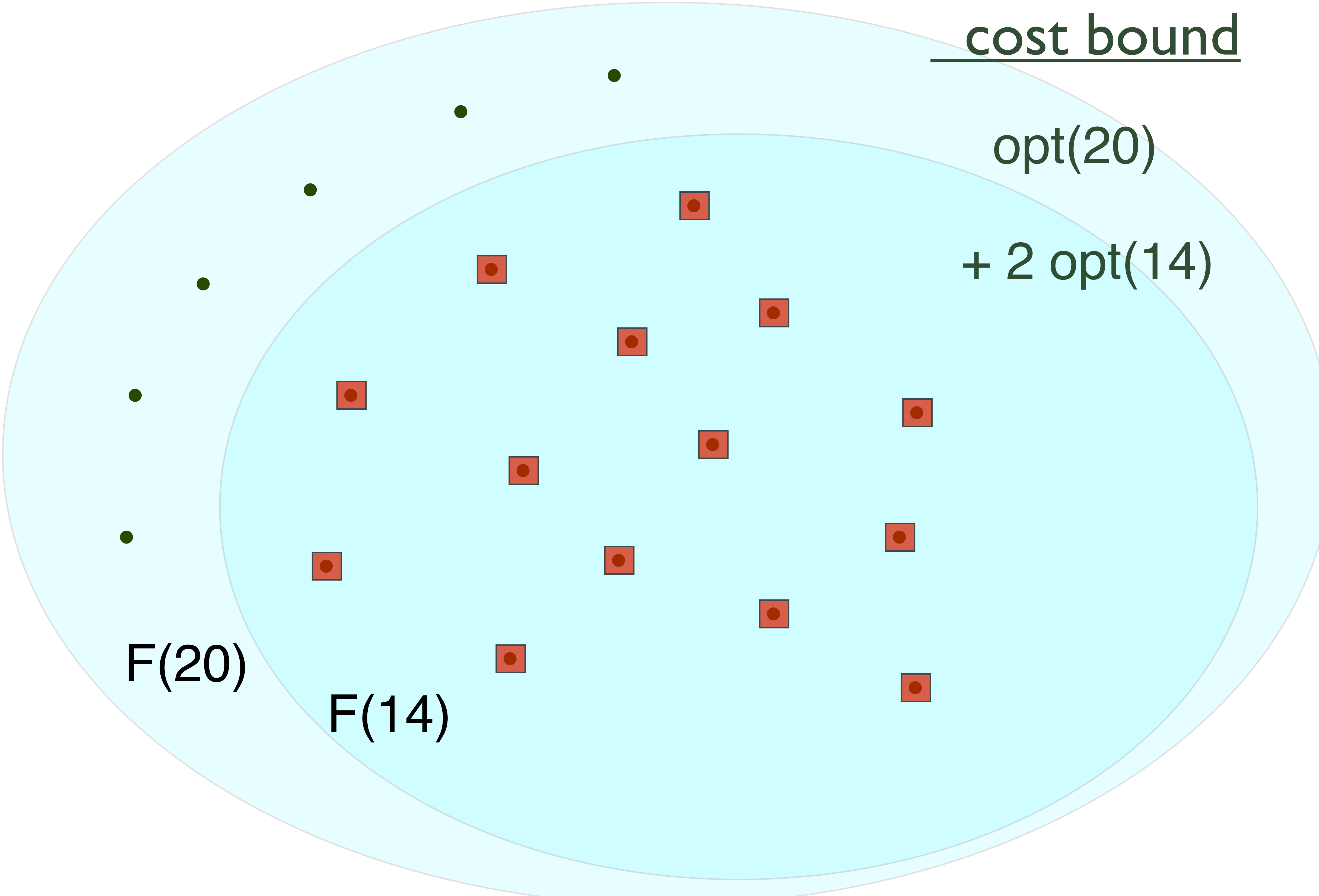
better algorithm -- "batch" reverse greedy

Project down at a few well-chosen k's:



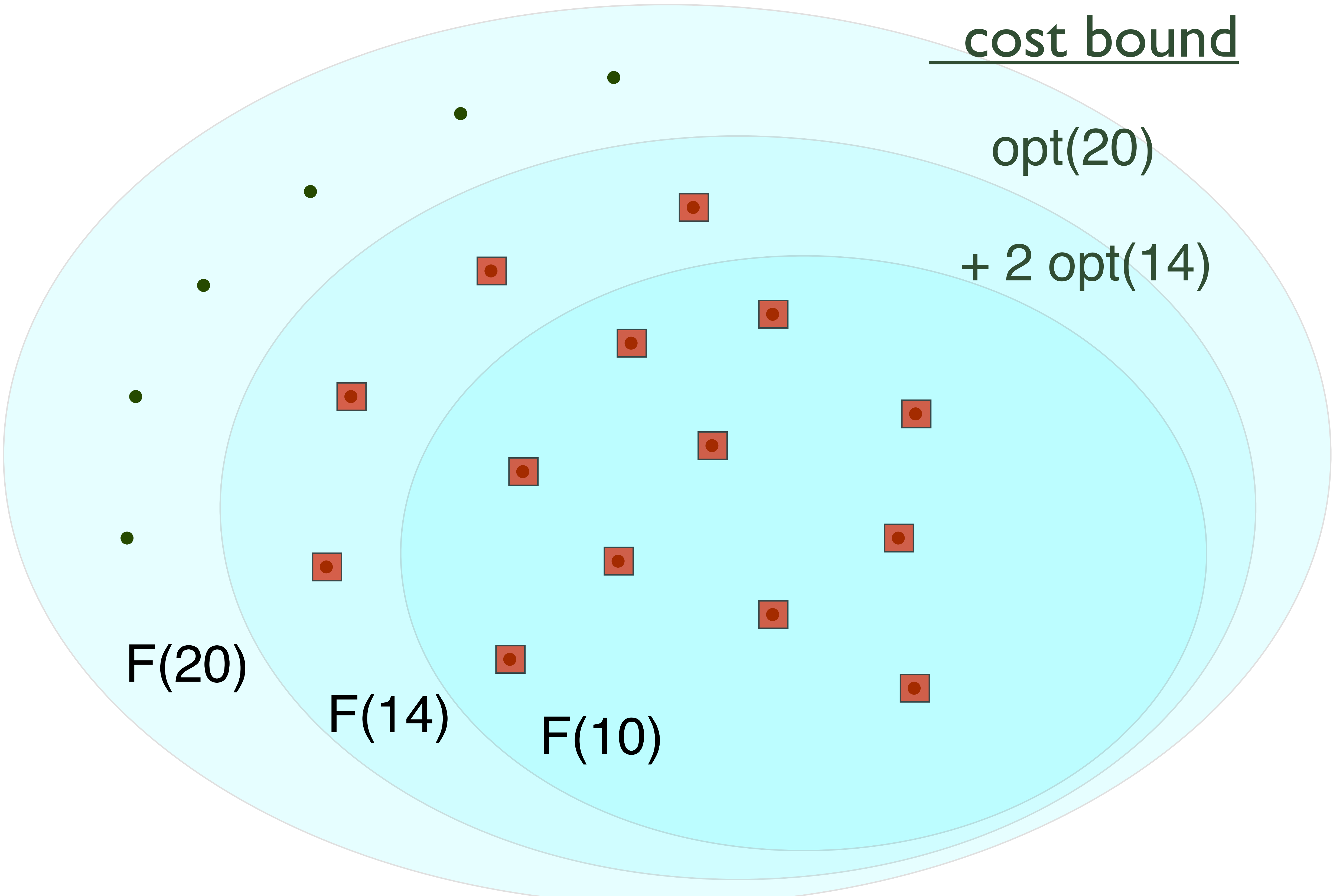
better algorithm -- "batch" reverse greedy

Project down at a few well-chosen k's:



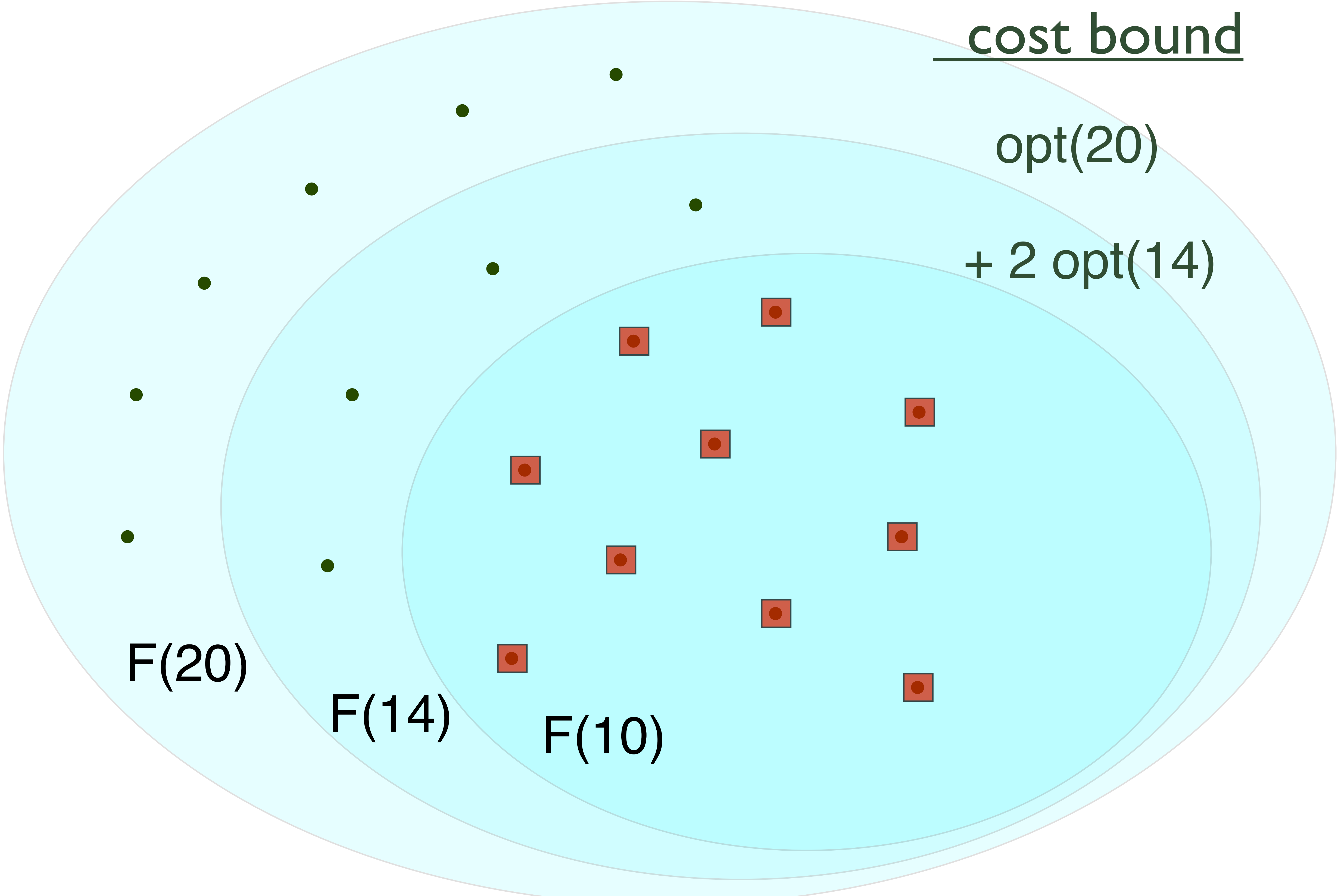
better algorithm -- "batch" reverse greedy

Project down at a few well-chosen k's:



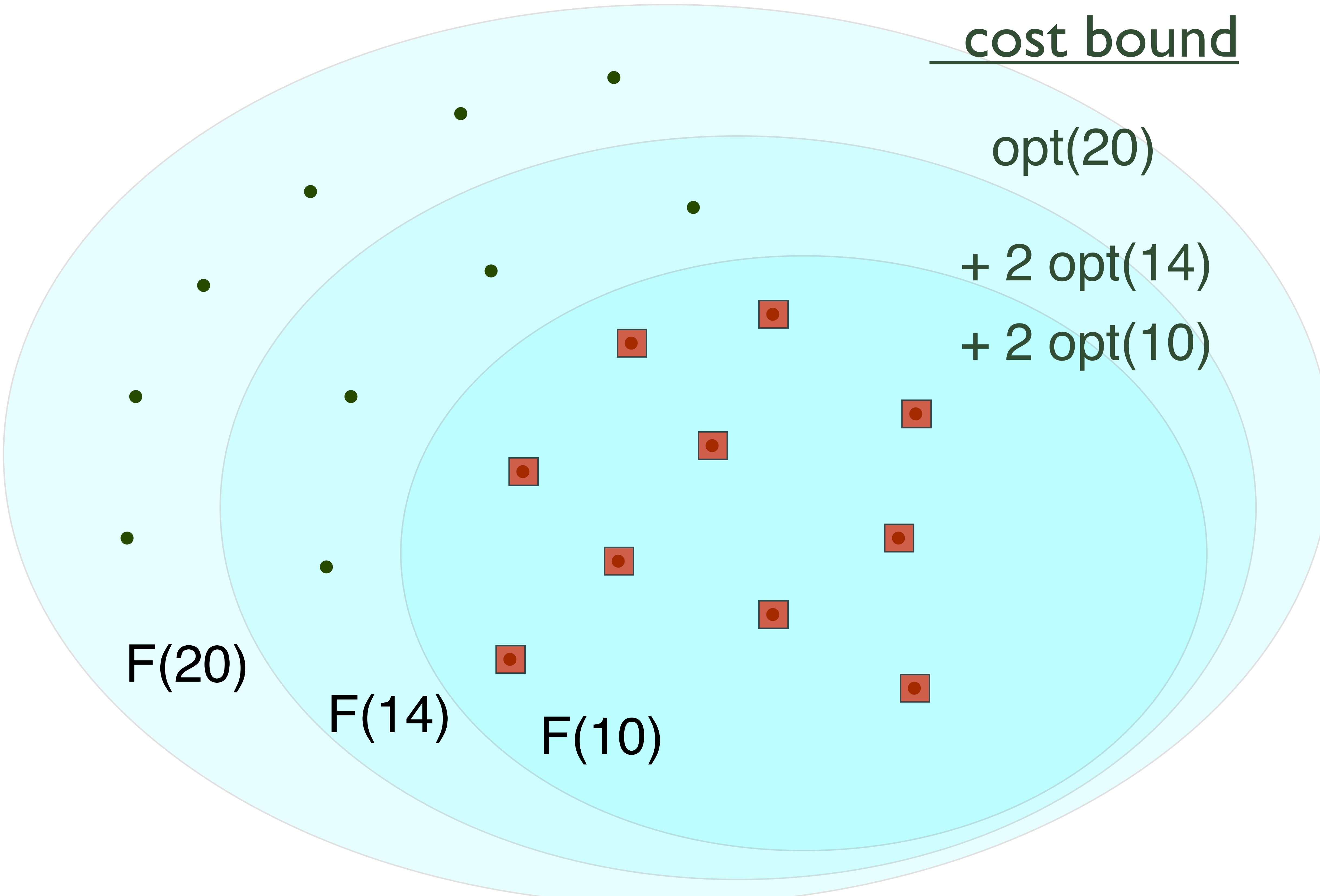
better algorithm -- "batch" reverse greedy

Project down at a few well-chosen k's:



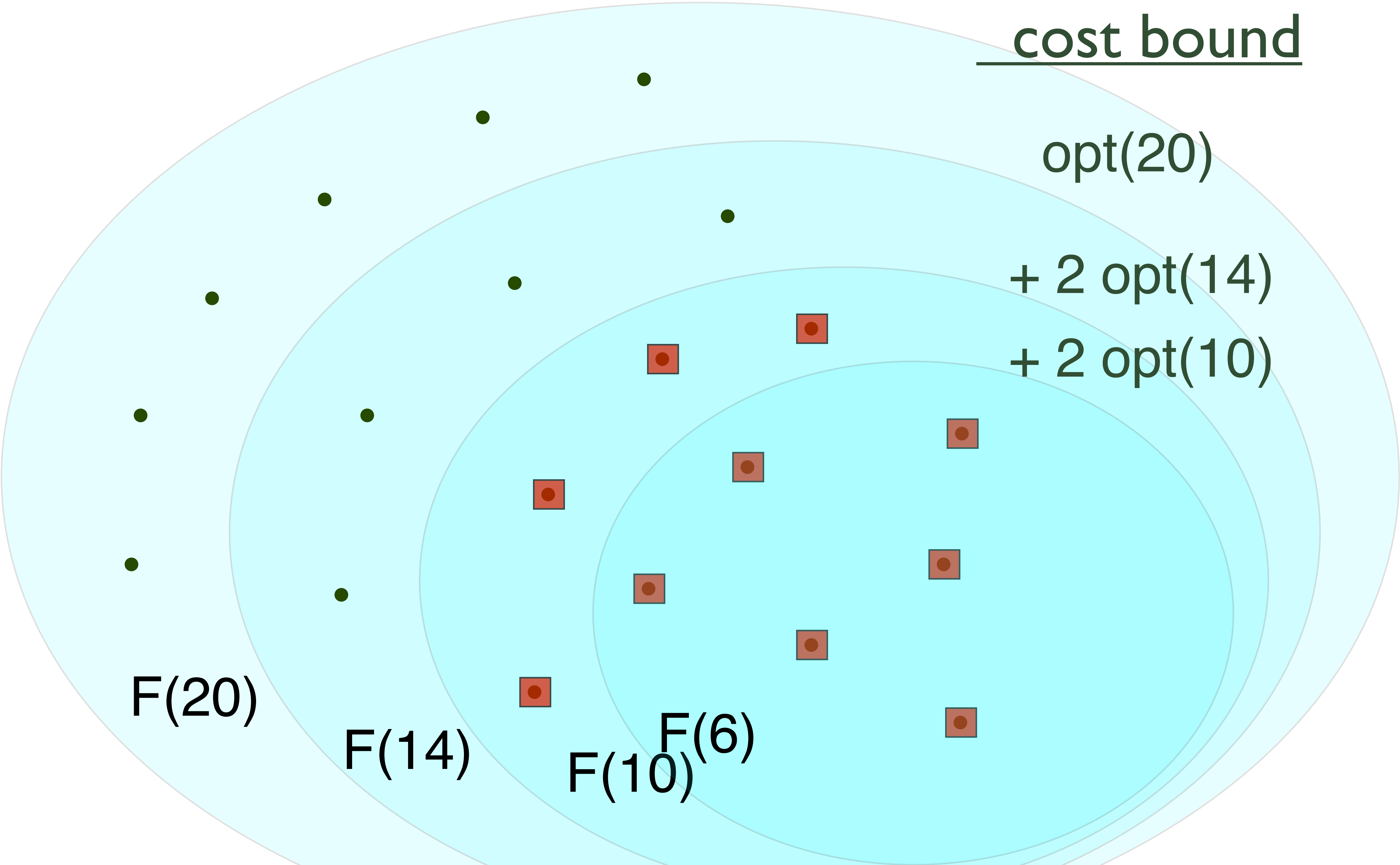
better algorithm -- "batch" reverse greedy

Project down at a few well-chosen k's:



better algorithm -- "batch" reverse greedy

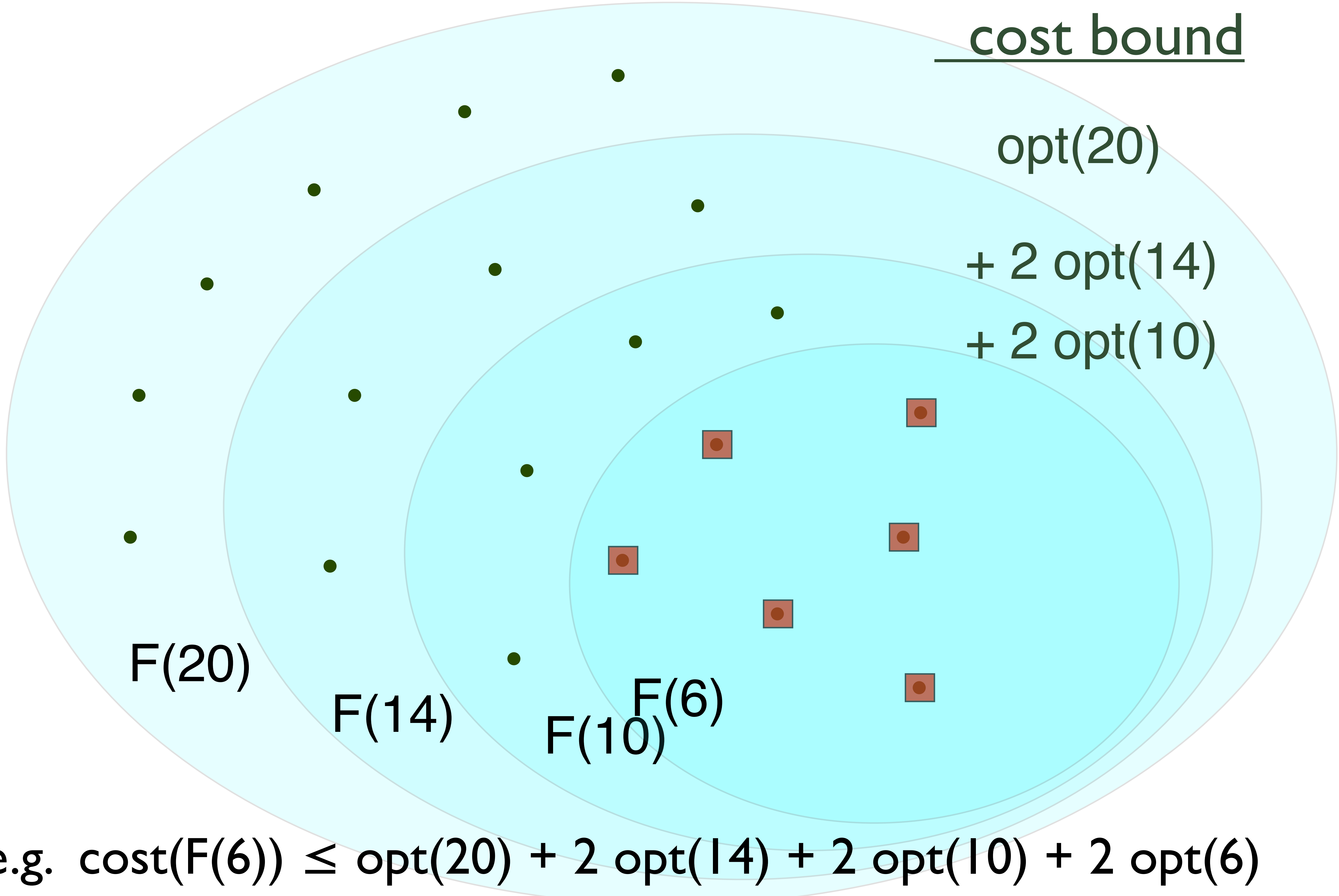
Project down at a few well-chosen k's:



e.g. $\text{cost}(F(6)) \le \text{opt}(20) + 2\,\text{opt}(14) + 2\,\text{opt}(10) + 2\,\text{opt}(6)$

better algorithm -- "batch" reverse greedy

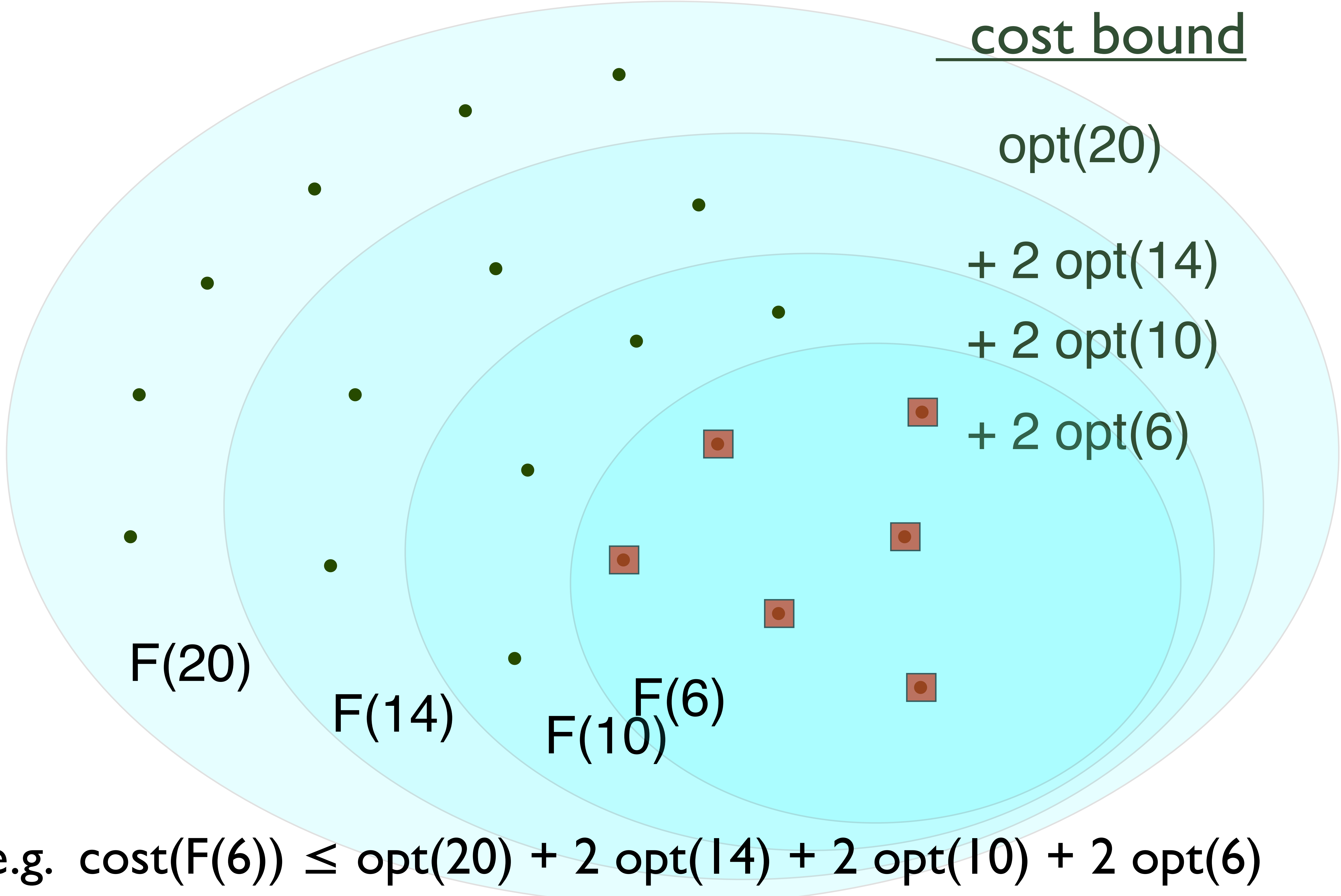
Project down at a few well-chosen k's:



e.g. cost(F(6)) $\leq$ opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6)

better algorithm -- "batch" reverse greedy

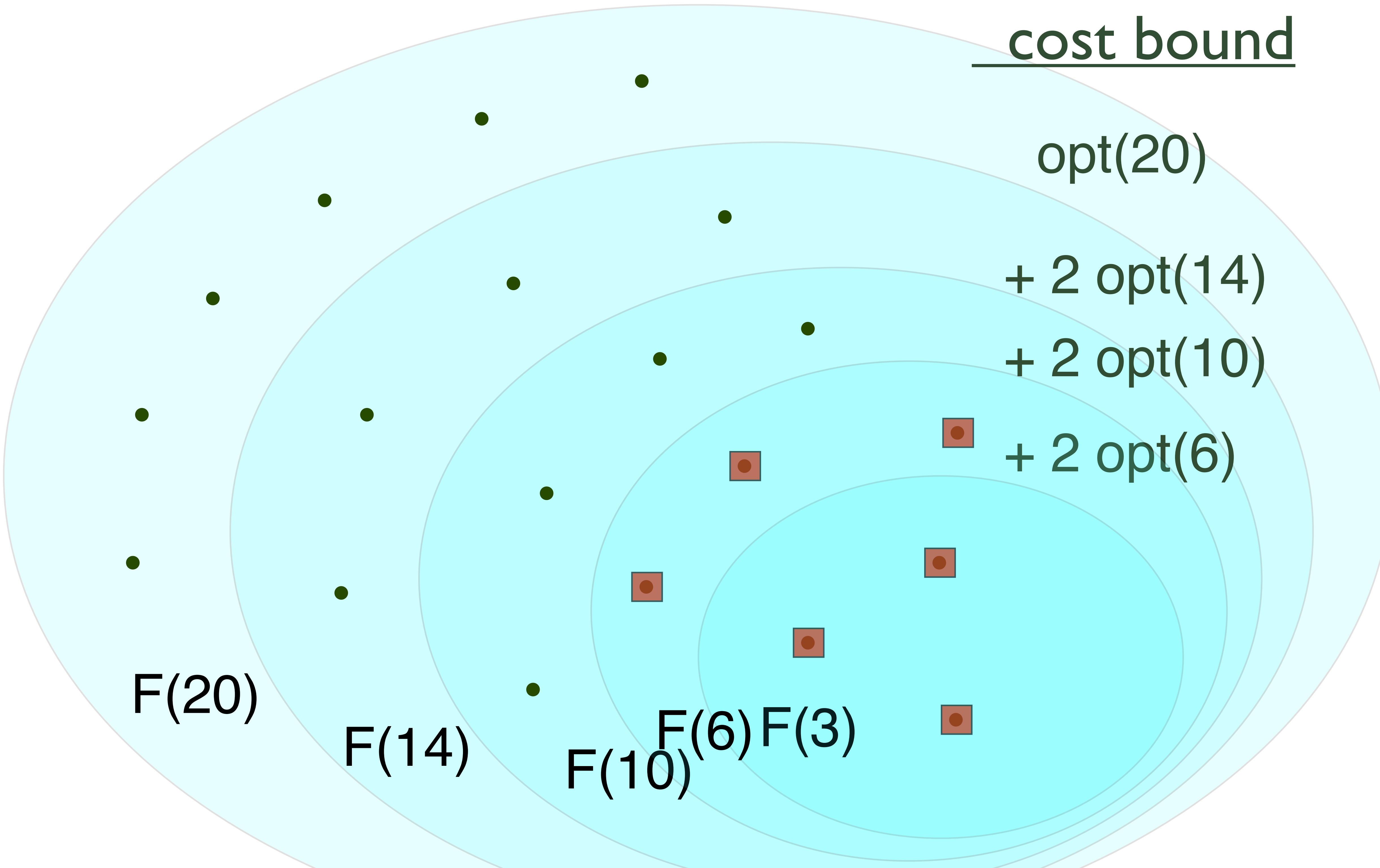
Project down at a few well-chosen k's:



e.g. $\text{cost}(F(6)) \leq \text{opt}(20) + 2\,\text{opt}(14) + 2\,\text{opt}(10) + 2\,\text{opt}(6)$

better algorithm -- "batch" reverse greedy

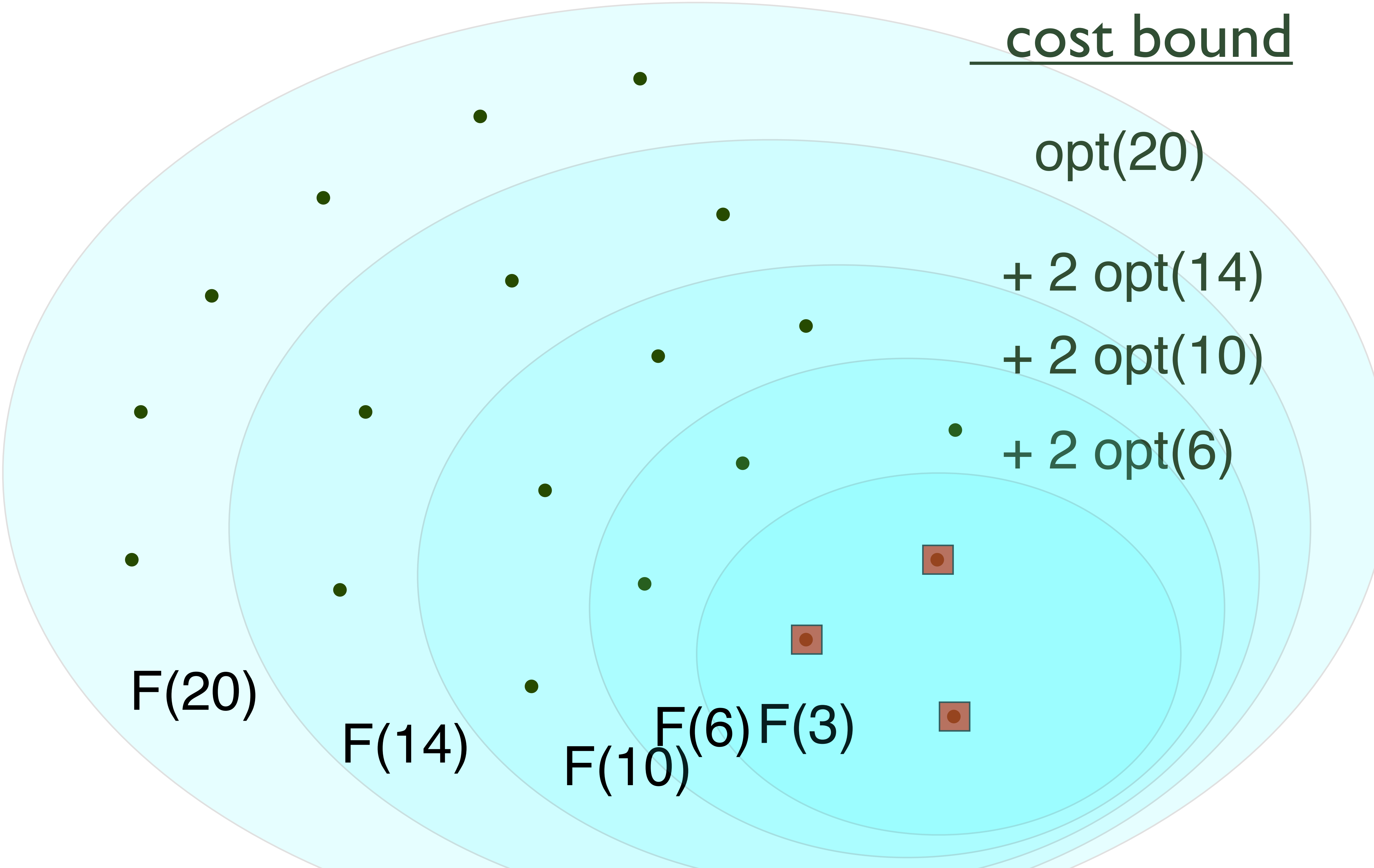
Project down at a few well-chosen k's:



e.g. $\text{cost}(F(6)) \leq \text{opt}(20) + 2\ \text{opt}(14) + 2\ \text{opt}(10) + 2\ \text{opt}(6)$

better algorithm -- "batch" reverse greedy

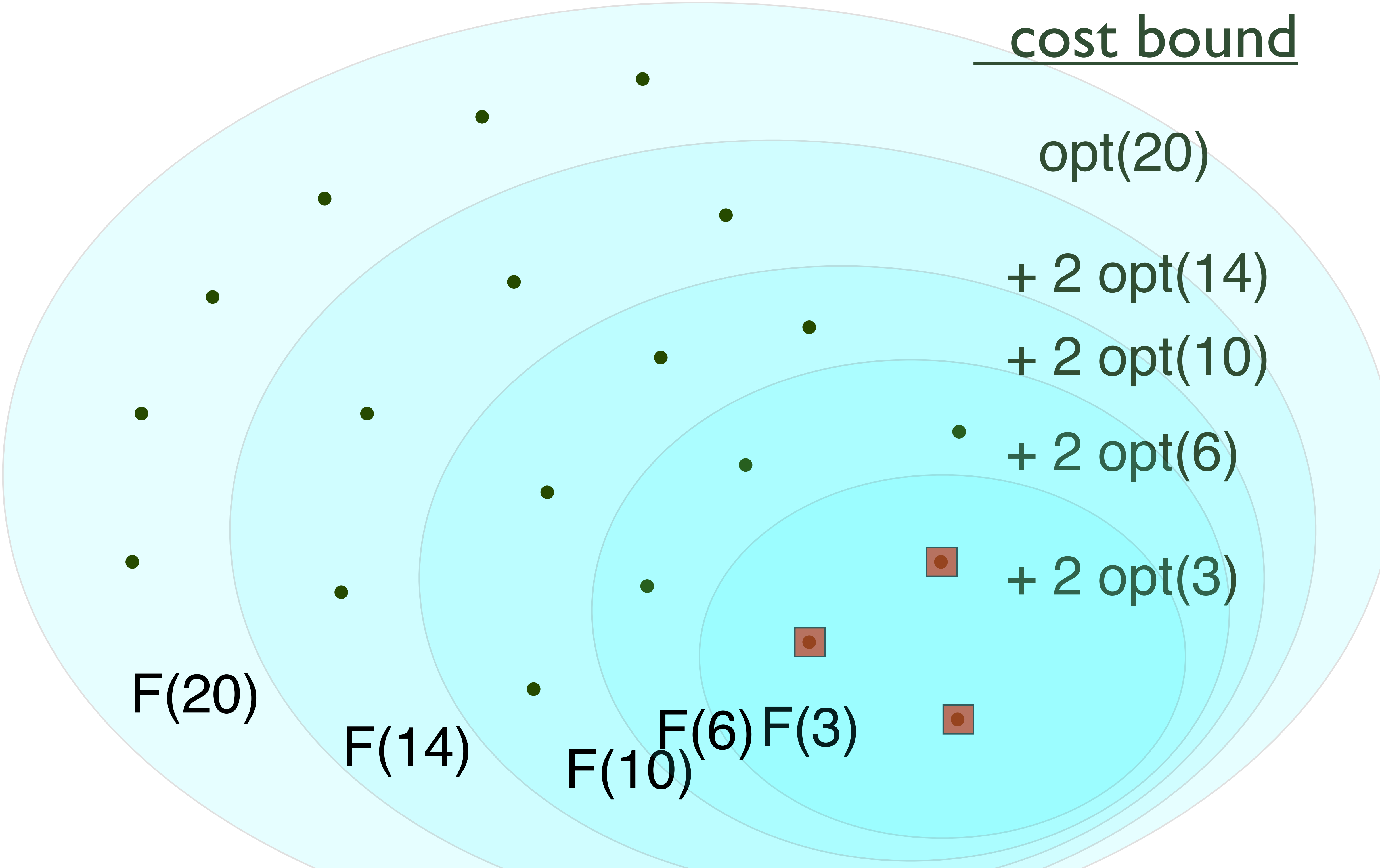
Project down at a few well-chosen k's:



e.g. cost(F(6)) ≤ opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6)

better algorithm -- "batch" reverse greedy

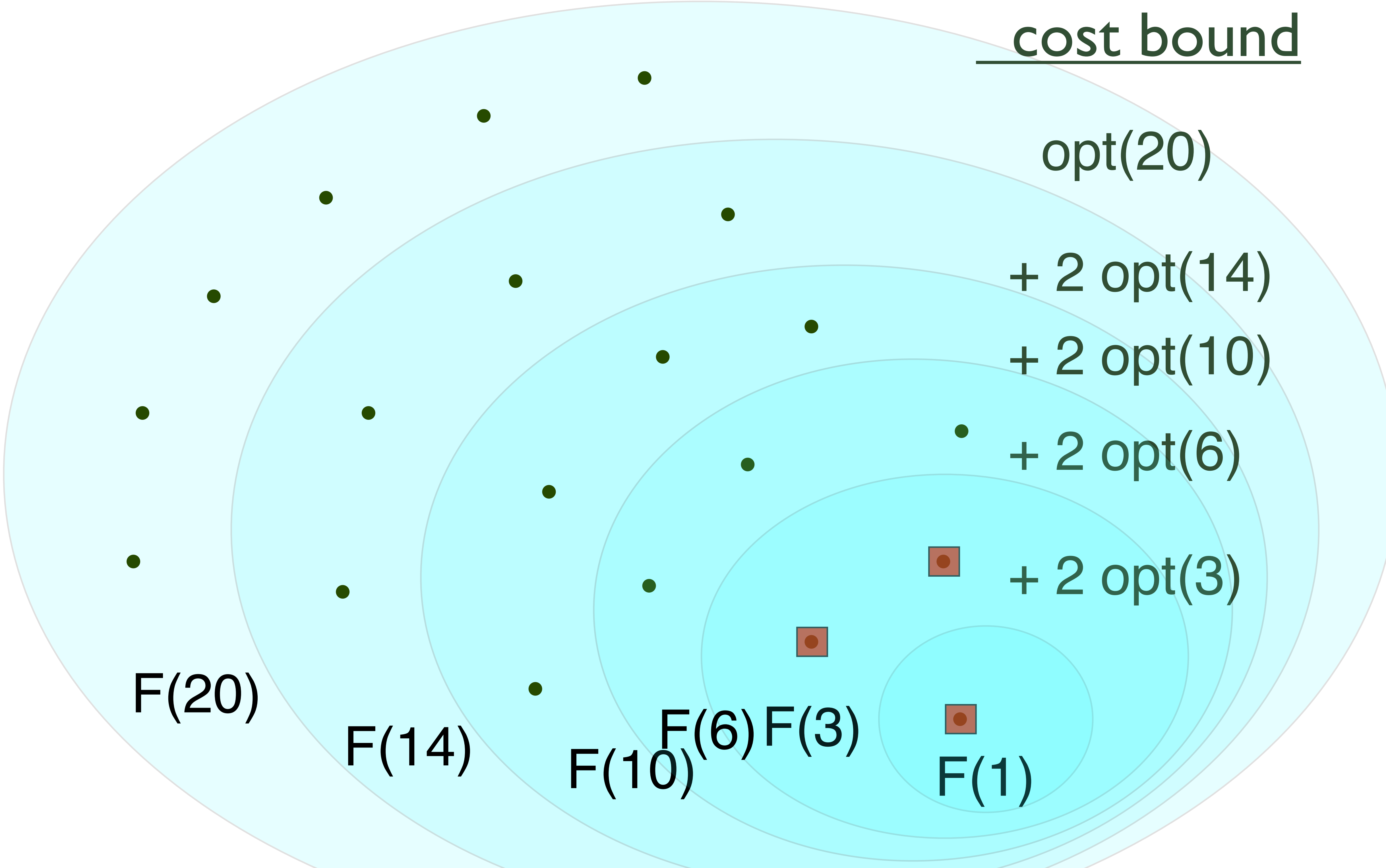
Project down at a few well-chosen k's:



e.g. cost(F(6)) $\le$ opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6)

better algorithm -- "batch" reverse greedy

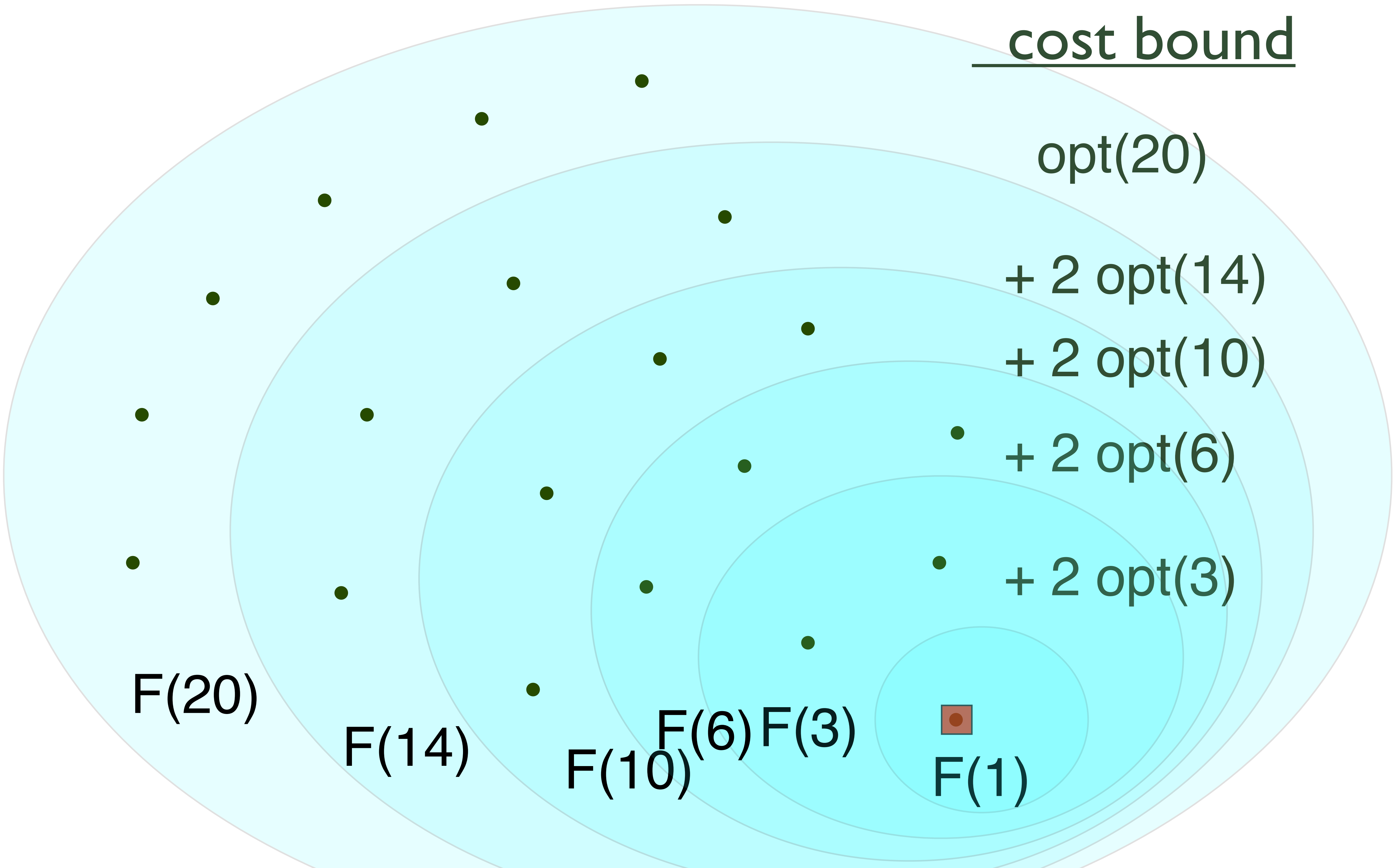
Project down at a few well-chosen k's:



e.g. cost(F(6)) $\leq$ opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6)

better algorithm -- "batch" reverse greedy

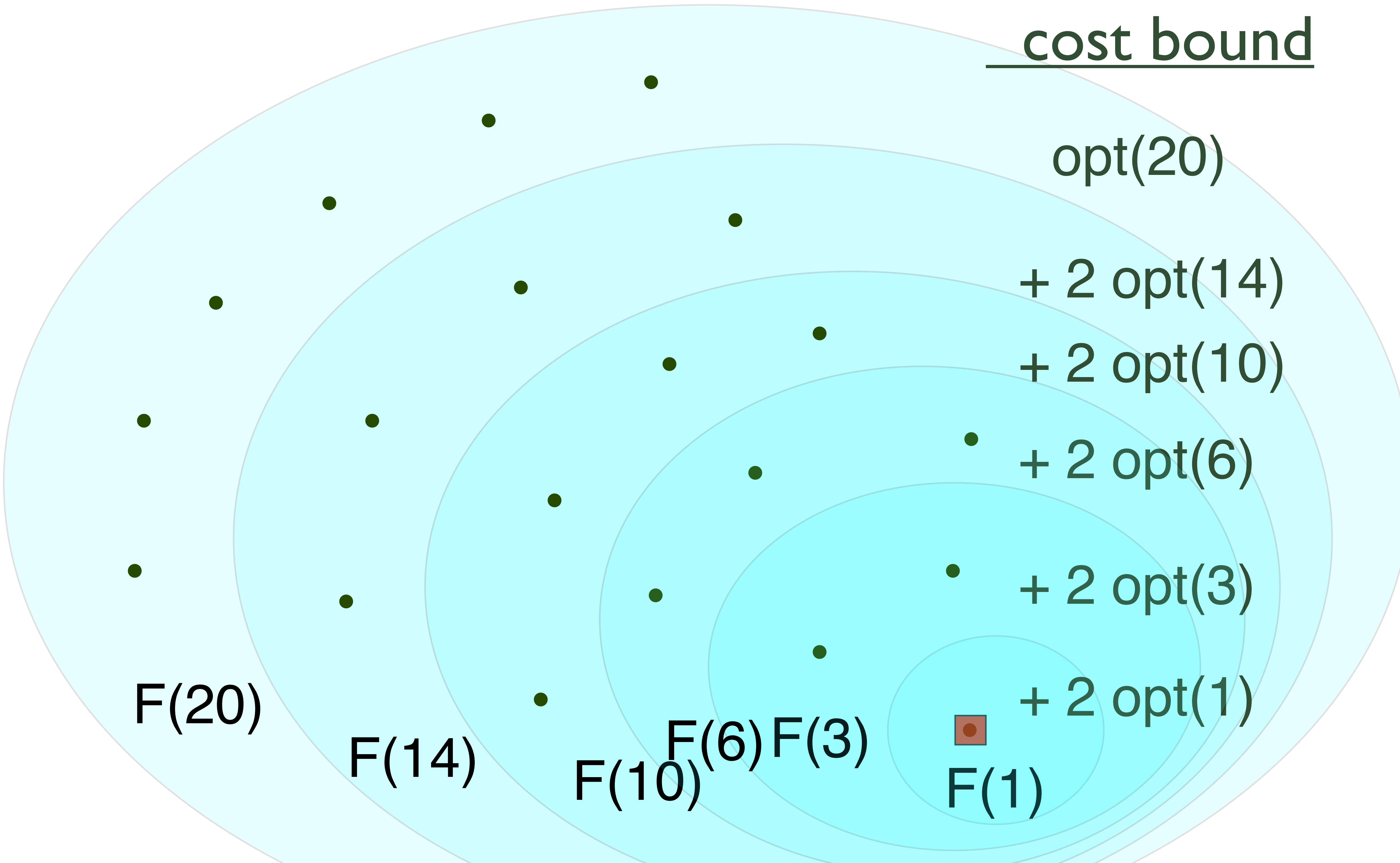
Project down at a few well-chosen k's:



e.g. cost(F(6)) ≤ opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6)

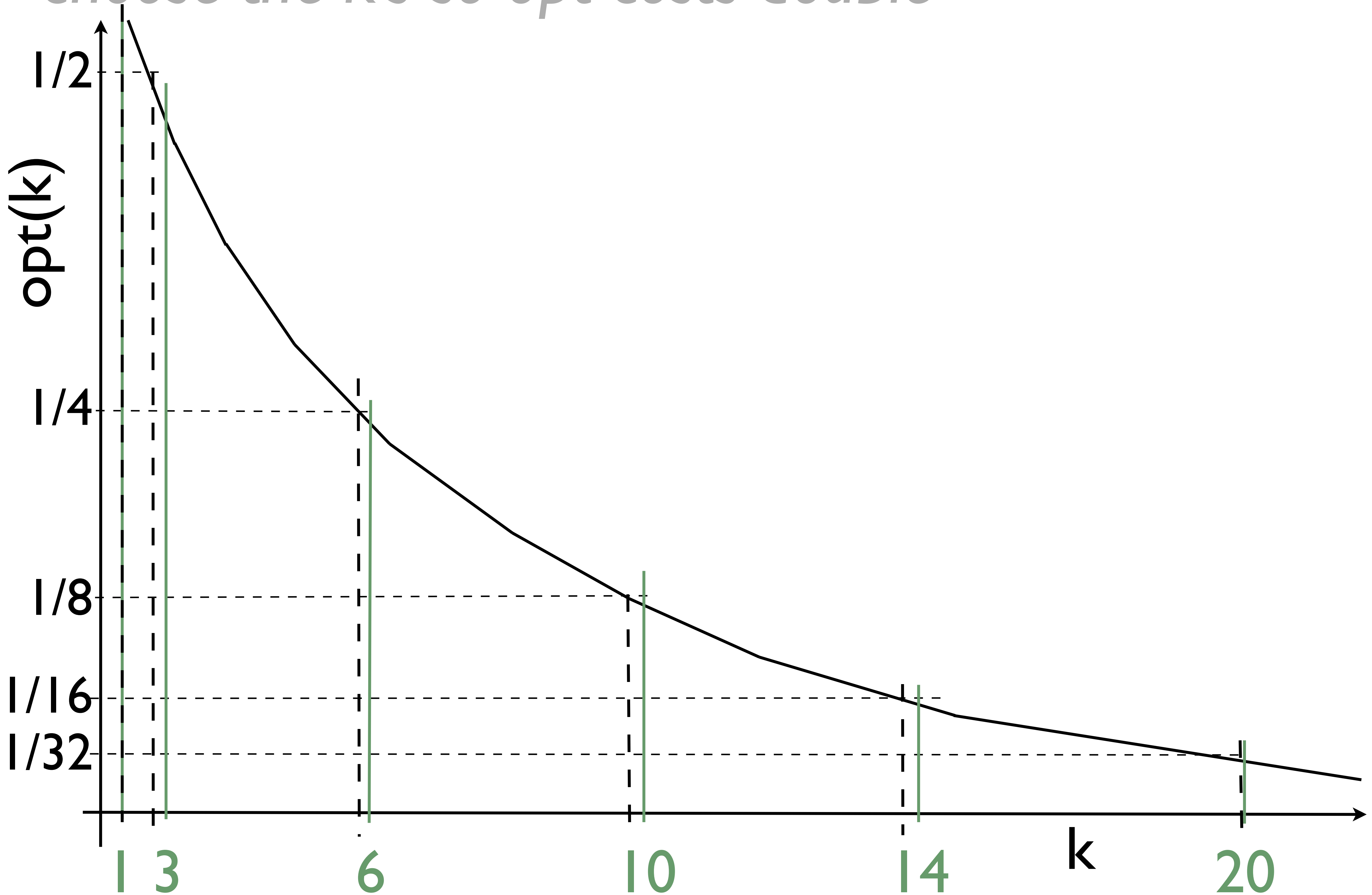
better algorithm -- "batch" reverse greedy

Project down at a few well-chosen k's:



e.g. $\text{cost}(F(6)) \leq \text{opt}(20) + 2\,\text{opt}(14) + 2\,\text{opt}(10) + 2\,\text{opt}(6)$

choose the k's so opt costs double



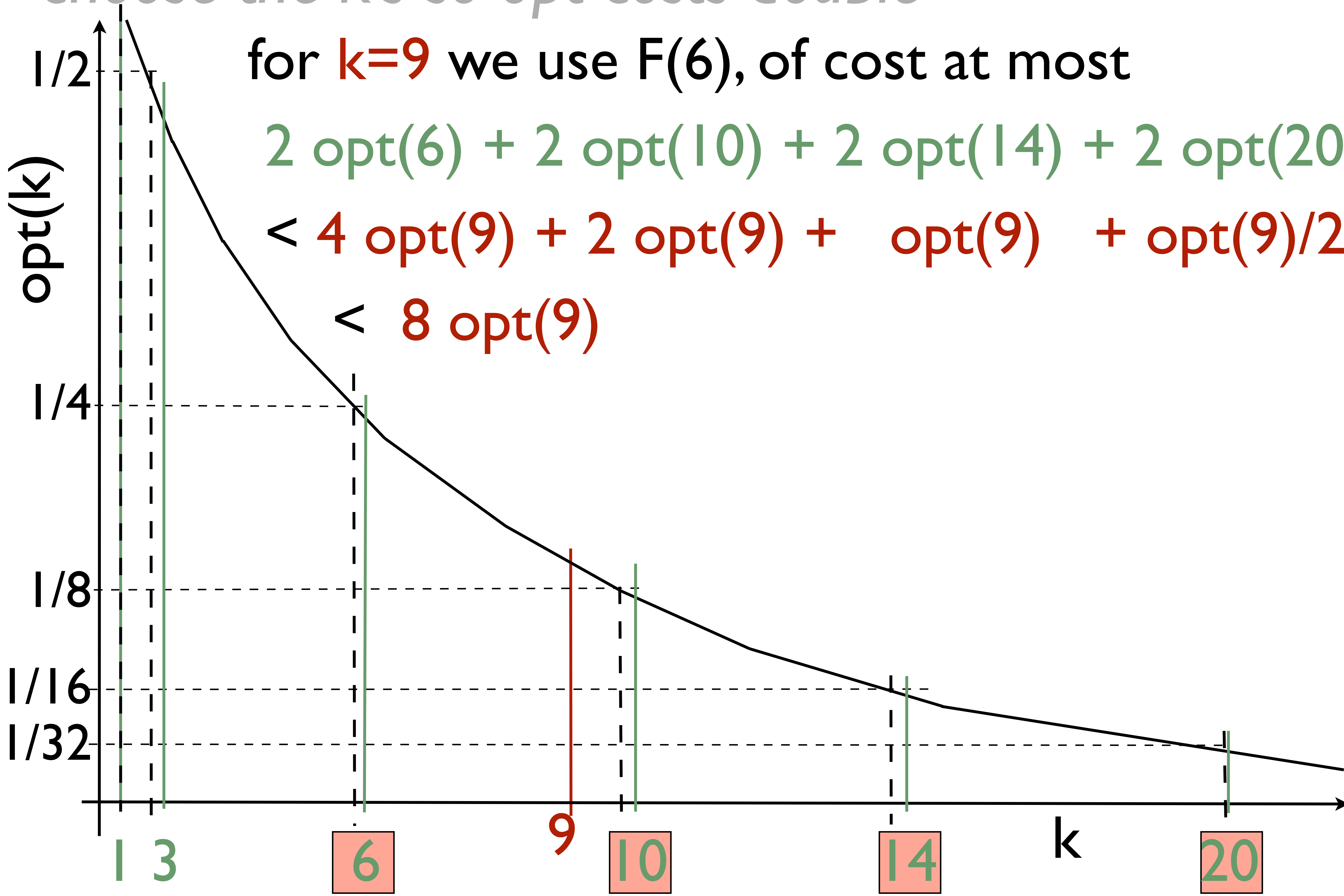
choose the k's so opt costs double

for k=9 we use F(6), of cost at most

2 opt(6) + 2 opt(10) + 2 opt(14) + 2 opt(20)

< 4 opt(9) + 2 opt(9) + opt(9) + opt(9)/2

< 8 opt(9)



result (upper bound on competitive ratio)

	exponential time	polynomial time
deterministic	8	$8(3+\varepsilon)$ $= 24+\varepsilon$
randomized	?	?

online bidding

instance:	an unknown threshold T ≥ 1.
to play:	submit bids until threshold is exceeded.
competitive ratio:	max over T of (sum of bids submitted) / T

<u>intuition</u>

Consider possible bids $b \in [1,\infty)$

Have to choose larger and larger bids.

T can be as small as last bid, so next bid can't be too large.
But bids that are too close together give high aggregate cost.

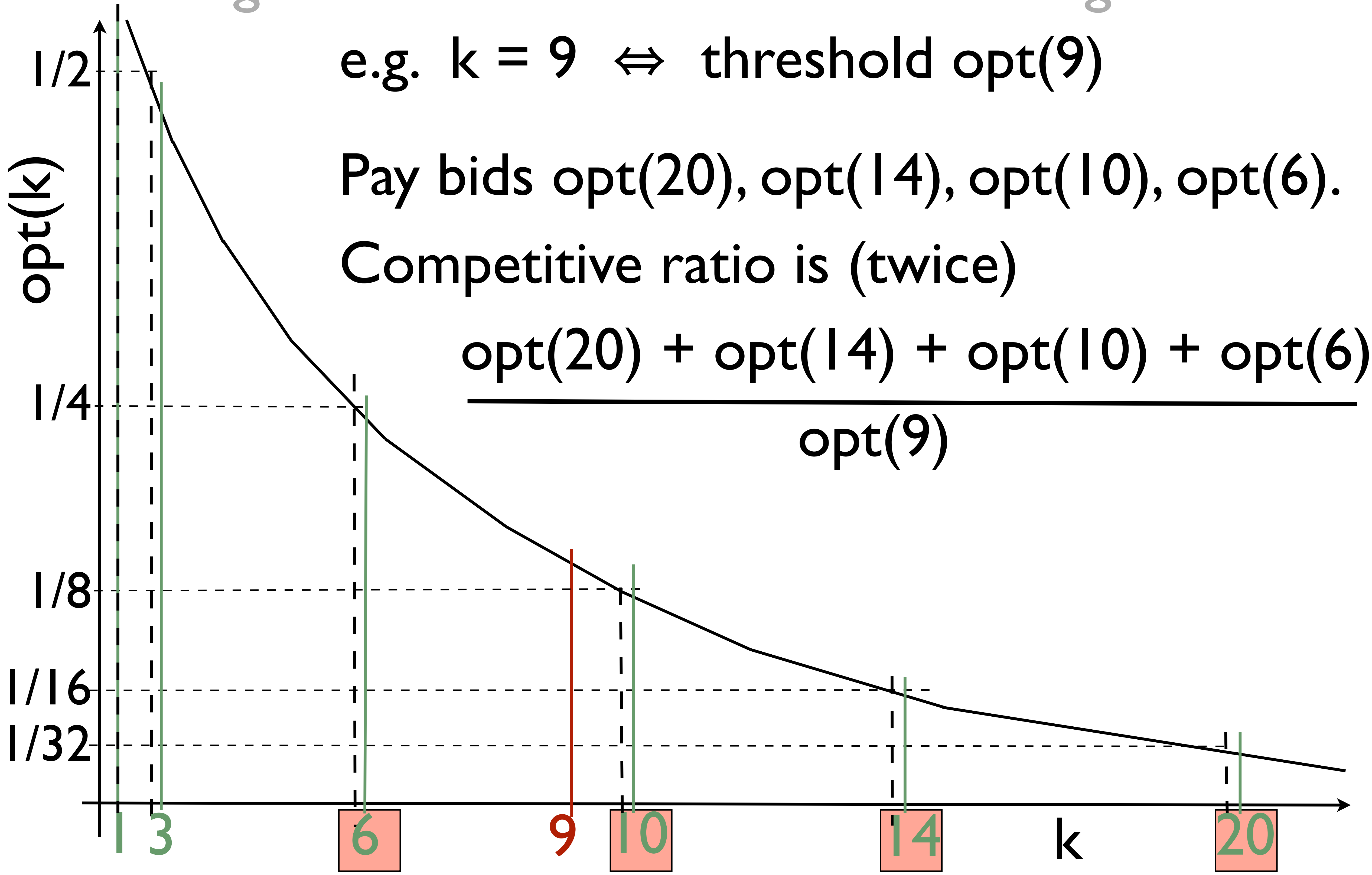
choosing the k's reduces to online bidding

e.g. $k = 9 \Leftrightarrow$ threshold opt(9)

Pay bids opt(20), opt(14), opt(10), opt(6).

Competitive ratio is (twice)

$$\frac{\text{opt}(20) + \text{opt}(14) + \text{opt}(10) + \text{opt}(6)}{\text{opt}(9)}$$



online bidding algorithms

Doubling strategy: Submit bids 1, 2, 4, 8, 16, ...

Doubling strategy is 4-competitive.

proof:

Sum of bids is at most $1+2+4+8+...+2T < 4T$.

This is the best possible competitive ratio.

randomized online bidding

strategy: submit bids e^x, e^{1+x}, e^{2+x}, e^{3+x}, e^{4+x}, ...
where x is chosen uniformly in $[0,1]$

Randomized strategy is e-competitive.

This is best possible for any randomized strategy.

lower bound for randomized online bidding

optimal strategy is solution to linear program:

x(t,b) = probability *b* is first bid that is *t* or larger

β = competitive ratio

$$\text{minimize}_{\beta,x}\ \beta\ \text{ subject to } \begin{cases} \beta - \sum_{b=1}^{n} \frac{b}{T} \sum_{t=1}^{T} x(t,b) & \geq & 0 & (\forall T \in [n]) \\ \sum_{b=T}^{n} \sum_{t=1}^{T} x(t,b) & \geq & 1 & (\forall T \in [n]) \\ x(t,b) & \geq & 0 & (\forall t, b \in [n]). \end{cases}$$

lower bound for randomized online bidding

lower bound follows from analytic solution to dual:

$$\text{maximize}_{\mu,\pi} \sum_{T=1}^{n} \mu(T) \quad \text{subject to} \begin{cases} \displaystyle\sum_{T=1}^{n} \pi(T) & \leq & 1 \\ \displaystyle\sum_{T=t}^{b} \mu(T) - \sum_{T=t}^{n} \frac{b}{T} \pi(T) & \leq & 0 \quad (\forall t, b \in [n]) \\ \mu(T), \pi(T) & \geq & 0 \quad (\forall T \in [n]). \end{cases}$$

$$\mu(T) = \begin{cases} \alpha/T & \text{if } U \leq T \leq U^2 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \pi(T) = \begin{cases} 1/T & \text{if } U \leq T \leq U^2 \log U \\ 0 & \text{otherwise.} \end{cases} .$$

result (upper bounds on competitive ratio)

	exponential time	polynomial time
deterministic	8	$8(3+\varepsilon)$ $= 24+\varepsilon$
randomized	$2e$ < 5.44	$2e(3+\varepsilon)$ < 16.31

thank you

projection lemma

For any F_k and any j

there exists $F_j \subseteq F_k$ with

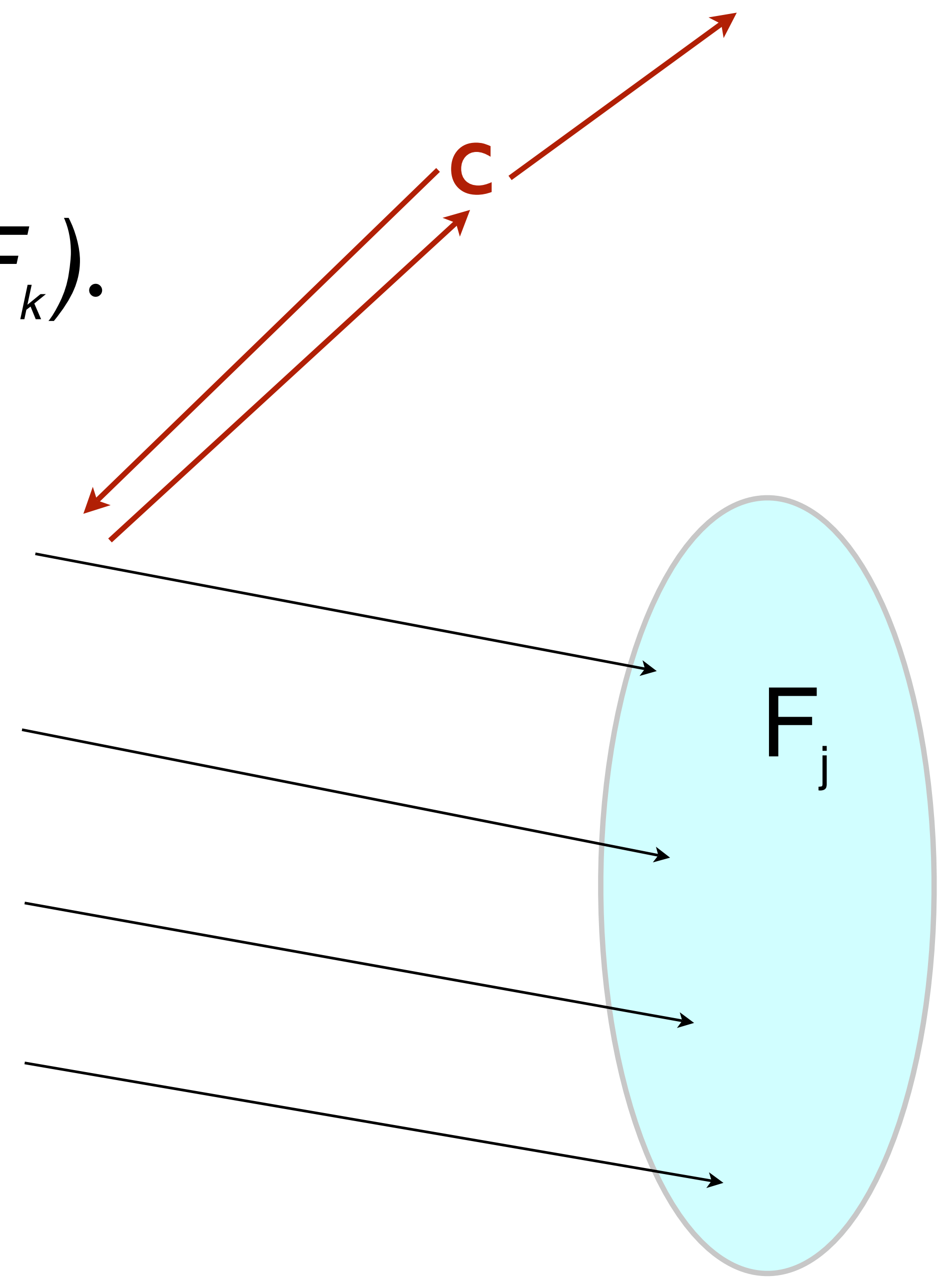
$$cost(F_j) \;\le\; 2\ OPT_j + cost(F_k).$$

proof:

For each point in OPT_j

take closest point in F_k.

Apply triangle inequality.



for any customer c

$$\begin{aligned} d(c,f_j) &\le d(c,opt_j) + d(opt_j,f_j) \\ &\le d(c,opt_j) + d(opt_j,f_k) \\ &\le d(c,opt_j) + d(opt_j,c) + d(c,f_k) \end{aligned}$$

projection lemma

For any F_k and any j

there exists $F_j \subseteq F_k$ with

$cost(F_j) \leq 2\ OPT_j + cost(F_k)$.

proof:

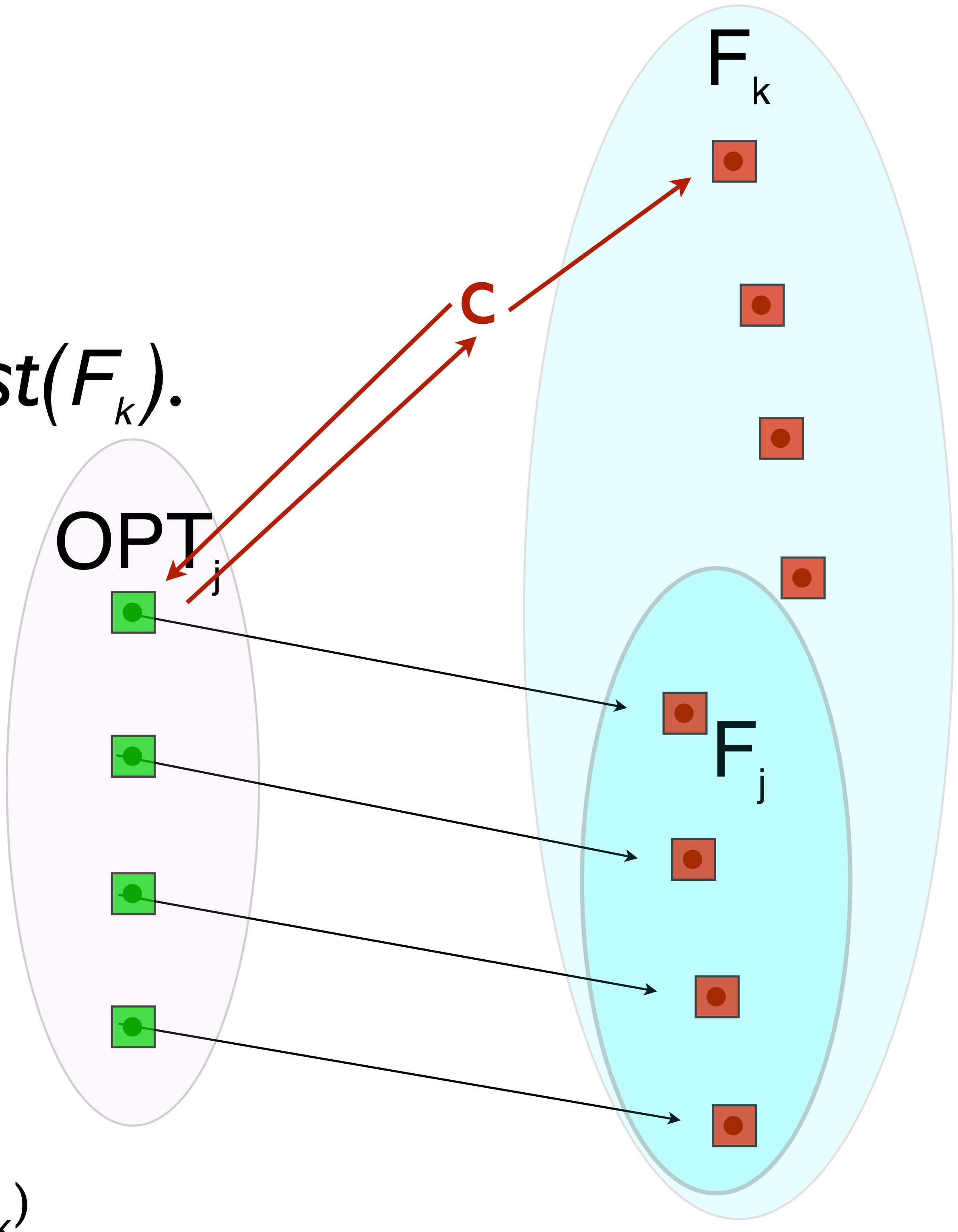
For each point in OPT_j

take closest point in F_k.

Apply triangle inequality.

for any customer c

$$\begin{aligned} d(c,f_j) &\leq d(c,opt_j) + d(opt_j,f_j) \\ &\leq d(c,opt_j) + d(opt_j,f_k) \\ &\leq d(c,opt_j) + d(opt_j,c) + d(c,f_k) \end{aligned}$$



choosing when to project

k	project?	F(k)	cost bound	versus
20		F(20)	opt(20)	opt(20)
19	no	F(14)	opt(20) + 2 opt(14)	opt(19)
18	no	F(14)	opt(20) + 2 opt(14)	opt(18)
17	no	F(14)	opt(20) + 2 opt(14)	opt(17)
16	no	F(14)	opt(20) + 2 opt(14)	opt(16)
15	no	F(14)	opt(20) + 2 opt(14)	opt(15)
14	yes	F(14)	opt(20) + 2 opt(14)	opt(14)
13	no	F(10)	opt(20) + 2 opt(14) + 2 opt(10)	opt(13)
12	no	F(10)	opt(20) + 2 opt(14) + 2 opt(10) F(6)	opt(12)
11	no	F(10)	opt(20) + 2 opt(14) + 2 opt(10)	opt(11)
10	yes	F(10)	opt(20) + 2 opt(14) + 2 opt(10)	opt(10)
9	no	F(6)	opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6)	opt(9)
8	no	F(6)	opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6)	opt(8)
7	no	F(6)	opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6)	opt(7)
6	yes	F(6)	opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6)	opt(6)
5	no	F(3)	opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6) + 2 opt(3)	opt(2)
4	no	F(3)	opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6) + 2 opt(3)	opt(4)
3	yes	F(3)	opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6) + 2 opt(3)	opt(3)
2	no	F(1)	opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6) + 2 opt(3) + 2 opt(1)	opt(2)
1	yes	F(1)	opt(20) + 2 opt(14) + 2 opt(10) + 2 opt(6) + 2 opt(3) + 2 opt(1)	opt(1)