RUTGERS

Rutgers Business School
Newark and New Brunswick

# Min sum ordering problems with applications to scheduling


Robbert Fokkink (Technical University of Delft)
Felix Happach (Technical University of Munich)
Lisa Hellerstein (New York University)
Thomas Lidbetter (Rutgers University)
László Végh (University of Bonn)


Wednesday June 4th 2025

# Plan

- A general min sum ordering problem
- A 2-approximation for a subclass of problems:

Fokkink, R., Lidbetter, T. and Végh, L.A., 2019. On submodular search and machine scheduling. *Mathematics of Operations Research*, *44*(4), pp.1431-1449.

- A $4\alpha$-approximation for another subclass:

Happach, F., Hellerstein, L. and Lidbetter, T., 2022. A general framework for approximating min sum ordering problems. *INFORMS Journal on Computing*, *34*(3), pp.1437-1452.

# A classic scheduling problem

- There are $n$ jobs that must be scheduled
- The weight (importance) of job $i$ is $w_i$
- The processing time of job $i$ is $p_i$

| | | | | | |
|---|---|---|---|---|---|
| $t_i$: | 3 hrs | 7 hrs | 3 hrs | 2 hrs | 6 hrs |
| $w_i$: | $2/15$ | $5/15$ | $3/15$ | $1/15$ | $4/15$ |

*Problem: In what order should the jobs be scheduled to minimize the sum of the weighted completion times? Usually denoted* $1||\sum w_i C_i$.

*Solution: Perform the jobs in non-increasing order of the index* $w_i/t_i$.

*(Smith, 1956)*

"Smith's rule"

# Writing in a different form

For a particular schedule, let $S_j$ = set of jobs up to and including job $j$. Then

Sum of weighted completion times = $\sum w_j C_j = \sum \left( g(S_j) - g(S_j - \{j\}) \right) f(S_j)$,

where $f(S) = \sum_{j \in S} p_j$ and $g(S) = \sum_{j \in S} w_j$.

Note: the functions $f$ and $g$ are both non-decreasing and *modular:*

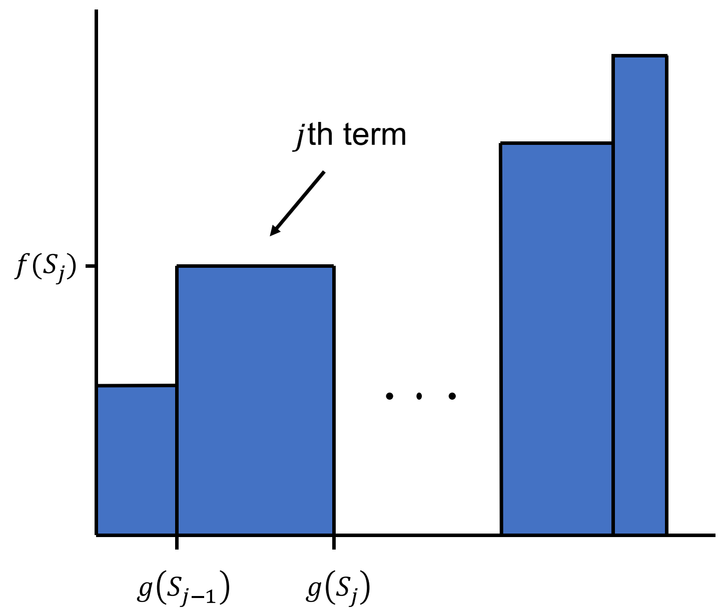
$f(S) = \sum_{j \in S} f(j)$ and $g(S) = \sum_{j \in S} g(j)$.

# Problem definition

- Finite set $V$ of cardinality $n$
- Non-decreasing cost function $f\colon \mathcal{F} \to \mathbb{R}$, where $f(\emptyset) = 0$.
- Non-decreasing weight function $g\colon \mathcal{F} \to \mathbb{R}$, where $g(\emptyset) = 0$.
- Minimize

$$\sum_{j=1}^{n} f(S_j)\left(g(S_j) - g(S_{j-1})\right),$$

over *chains* $\mathcal{S} \equiv (S_0, S_1, \ldots, S_n)$ satisfying
$\emptyset = S_0 \subset S_1 \cdots \subset S_n = V$, $|S_i| = i$ for all $i$.

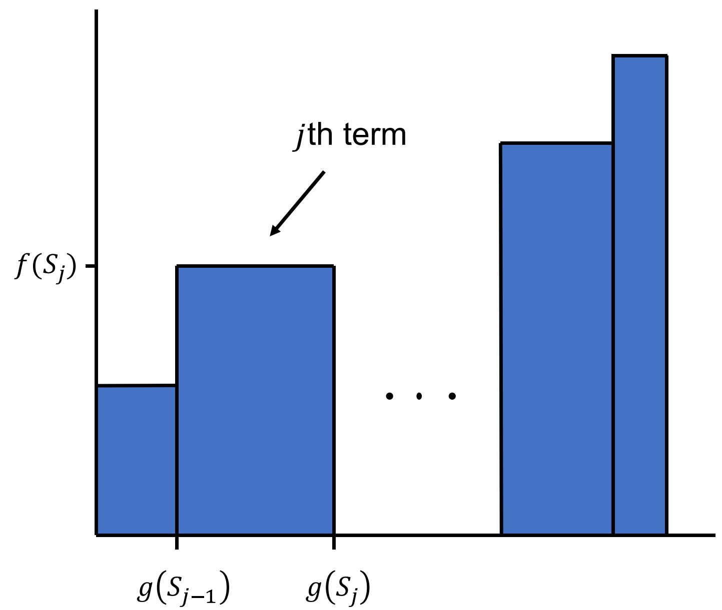


# Problem definition

- Finite set $V$ of cardinality $n$
- Non-decreasing cost function $f\colon \mathcal{F} \to \mathbb{R}$, where $f(\emptyset) = 0$.
- Non-decreasing weight function $g\colon \mathcal{F} \to \mathbb{R}$, where $g(\emptyset) = 0$.
- Min-sum ordering problem (MSOP): minimize

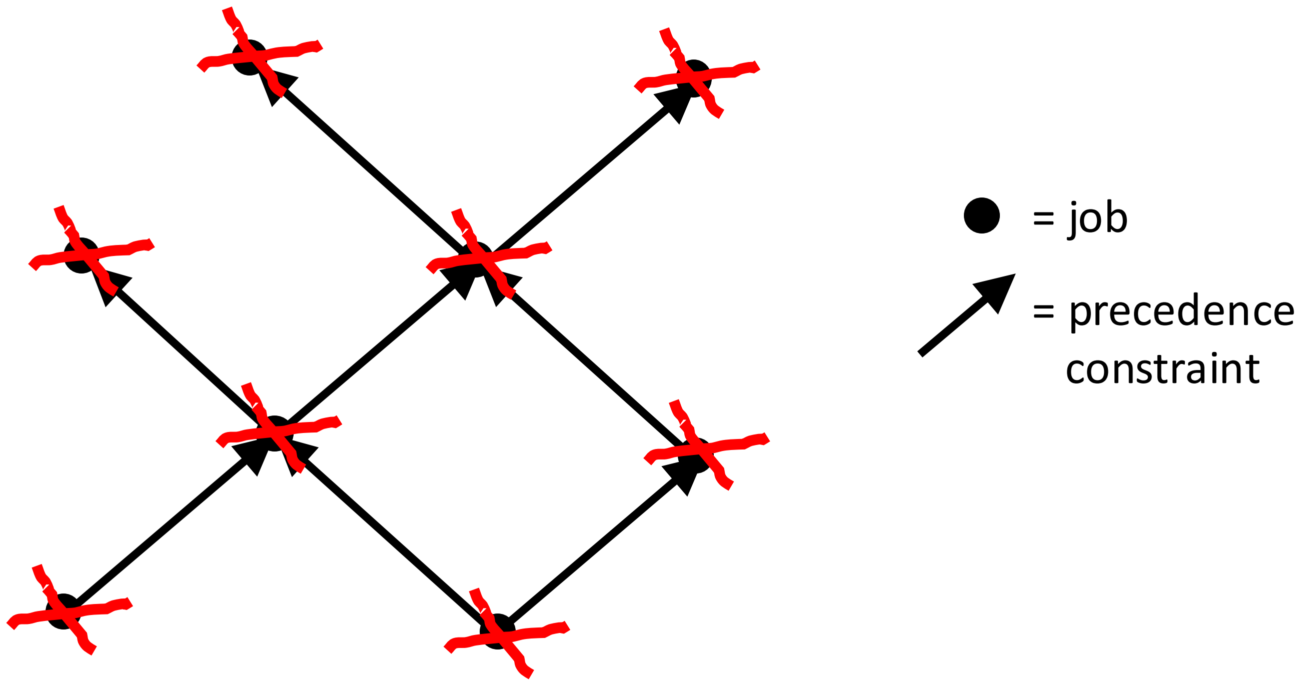
$$\sum_{j=1}^{k} f(S_j)\left(g(S_j) - g(S_{j-1})\right),$$

over *chains* $\mathcal{S} \equiv (S_0, S_1, \dots, S_k)$ satisfying $\emptyset = S_0 \subset S_1 \cdots \subset S_k = V$.



# A harder scheduling problem

Precedence constraints given by a DAG (directed acyclic graph).

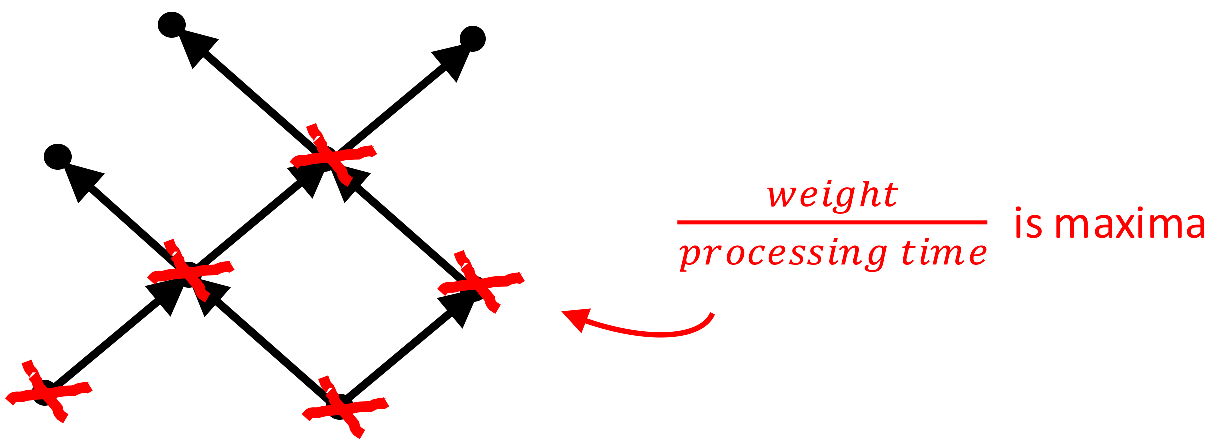



*Problem: In what order should the jobs be scheduled to minimize the sum of the weighted completion times? (*$1|prec|\sum w_i C_i$*)*

# A harder scheduling problem

*Problem: In what order should the jobs be scheduled to minimize the sum of the weighted completion times? Usually denoted* $1|prec|\sum w_i C_i$.

- Sidney (1975): an optimal schedule must begin with a closed sub-DAG $G$ of jobs that maximizes $w(G)/p(G)$. (Version of Smith's rule.)




- Recursing, this principle defines a *Sidney decomposition* of the jobs.

- If the precedence constraints are tree-like, this means the problem can be solved in polynomial time (Sidney, 1975)

# A harder scheduling problem

- Later…the problem is shown to be NP-hard problem. Many 2-approximations (schedules whose objective is within a factor 2 of that of the optimal schedule):
  - Ambühl & Mastrolilli (2009)
  - Chudak & Hochbaum (1999)
  - Chekuri & Motwani (1999)
  - Hall, Schulz, Shmoys, Wein (1997)
  - Margot, Queyranne & Wang (2003)
  - Pisaruk (2003)
  - Schulz (1996)
- In particular, Chekuri & Motwani (1999) and (independently) Margot, Queyranne & Wang (2003) showed that *any* schedule consistent with a Sidney decomposition is a 2-approximation.
- Virtually all known 2-approximations are consistent with a Sidney decomposition (Correa & Schulz, 2005)

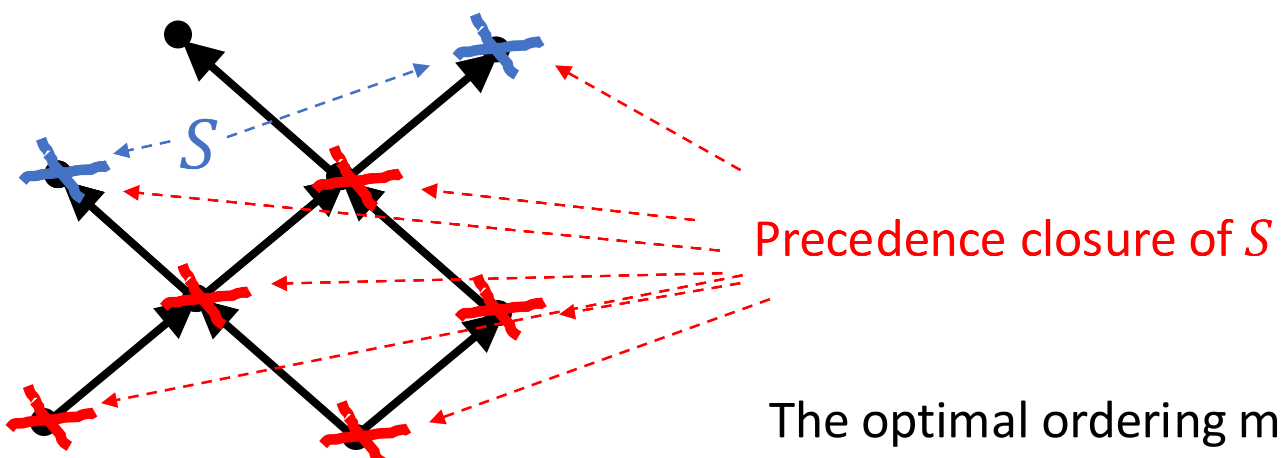
# Writing in a different form

Want to state the problem in the form: *minimize over all orderings the sum*

$$\sum_j f(S_j)\left(g(S_j) - g(S_{j-1})\right).$$

This time, take $f(S)$ = time to process all jobs in *precedence closure of* $S$,

$g(S)$ = weight of all jobs in $S$.




The optimal ordering must respect the precedence constraints.

# Writing in a different form

Want to state the problem in the form: *minimize over all orderings the sum*

$$\sum_j f(S_j)\left(g(S_j) - g(S_{j-1})\right).$$

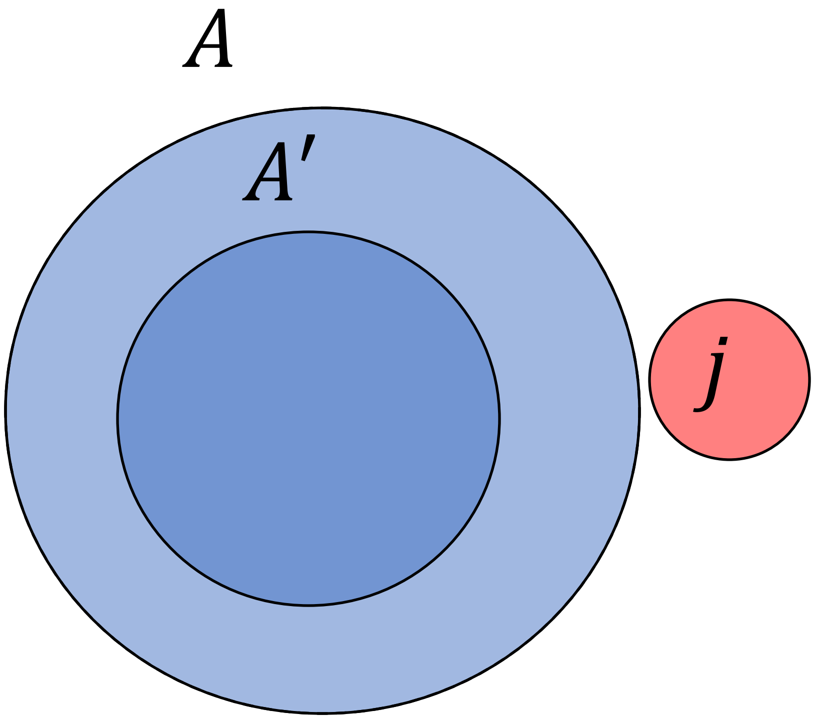
This time, take $f(S) =$ time to process all jobs in *precedence closure of* $S$,

$g(S) =$ weight of all jobs in $S$.

The function $g$ is still modular, but $f$ is *submodular* ("decreasing marginal costs"):

$$f(A \cup \{j\}) - f(A) \ \leq f(A' \cup \{j\}) - f(A') \text{ for } j \notin A \supset A'$$
$$\Leftrightarrow f(A \cup B) + f(A \cap B) \leq f(A) + f(B) \text{ for all } A, B$$

# Submodularity




$$f(A \cup j) - f(A)$$
$$\leq$$
$$f(A' \cup j) - f(A')$$

# A new scheduling problem

- Now instead of jobs having weights, every subset $A$ of jobs has a weight $w_A$. (Weights could be equal to $0$ for most subsets.)

- We wish to minimize the weighted sum of the completion times $C_A$ of the subsets of jobs = $\sum w_A C_A$. Denote this $1|prec|\sum w_A C_A$.

- Equivalently, objects are hidden in a subset $A$ of boxes with probability $p_A$ and we wish to minimize the expected time to find all of them.

- Let $g(A) =$ sum of weights of all subsets of $A$ and let $f$ be as before. Then objective is to find an ordering to minimize

$$\sum_j f(S_j)\left(g(S_j) - g(S_{j-1})\right).$$

- Then $g$ is supermodular ("increasing marginal benefit"):
$$g(A \cup \{j\}) - g(A) \ \leq g(A' \cup \{j\}) - g(A') \text{ for } j \notin A \supset A'$$

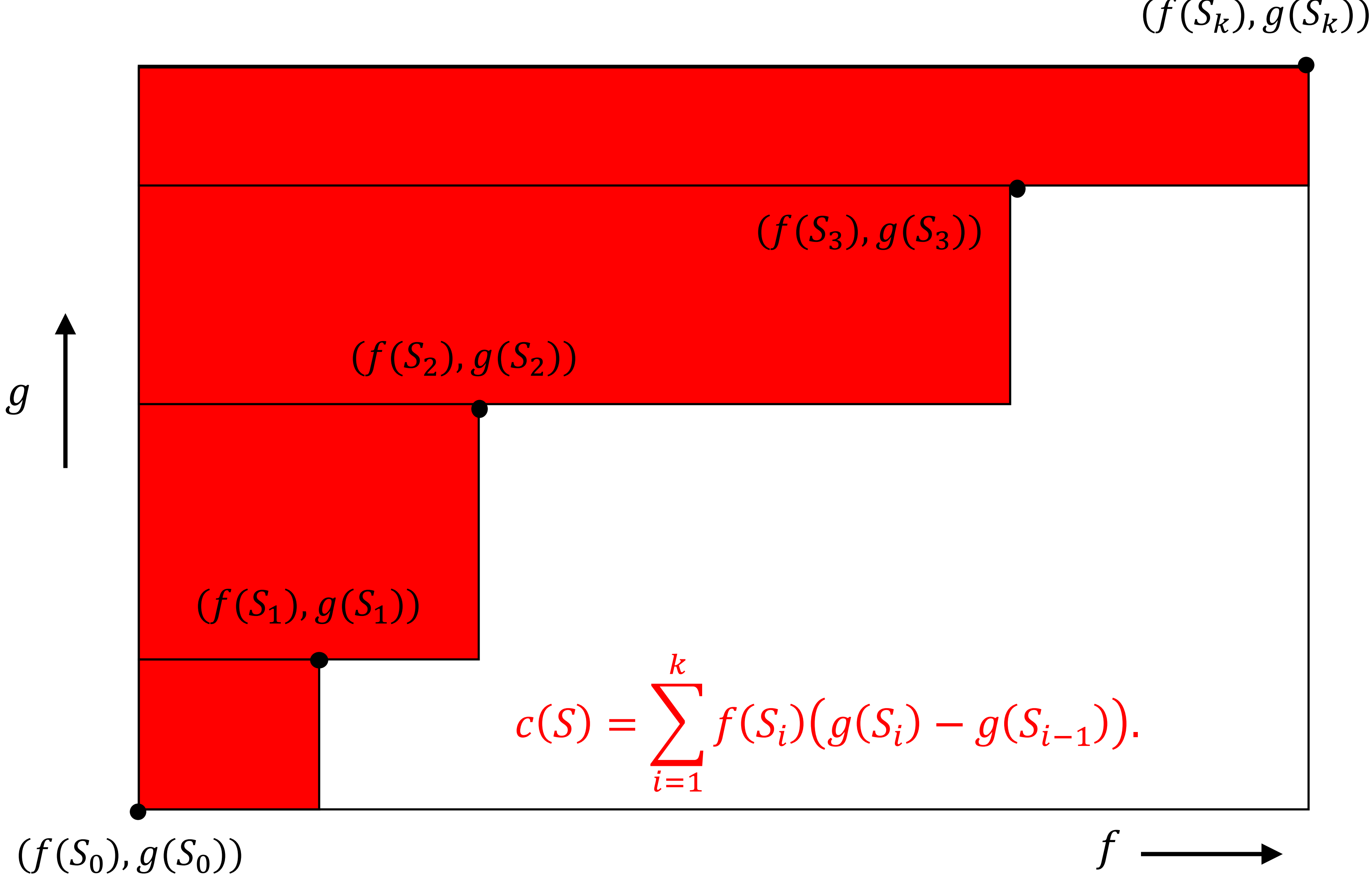
# A more general problem

MSOP with $f$ submodular, $g$ supermodular: minimize

$$\sum_{i=1}^{k} f(S_i)\big(g(S_i) - g(S_{i-1})\big).$$

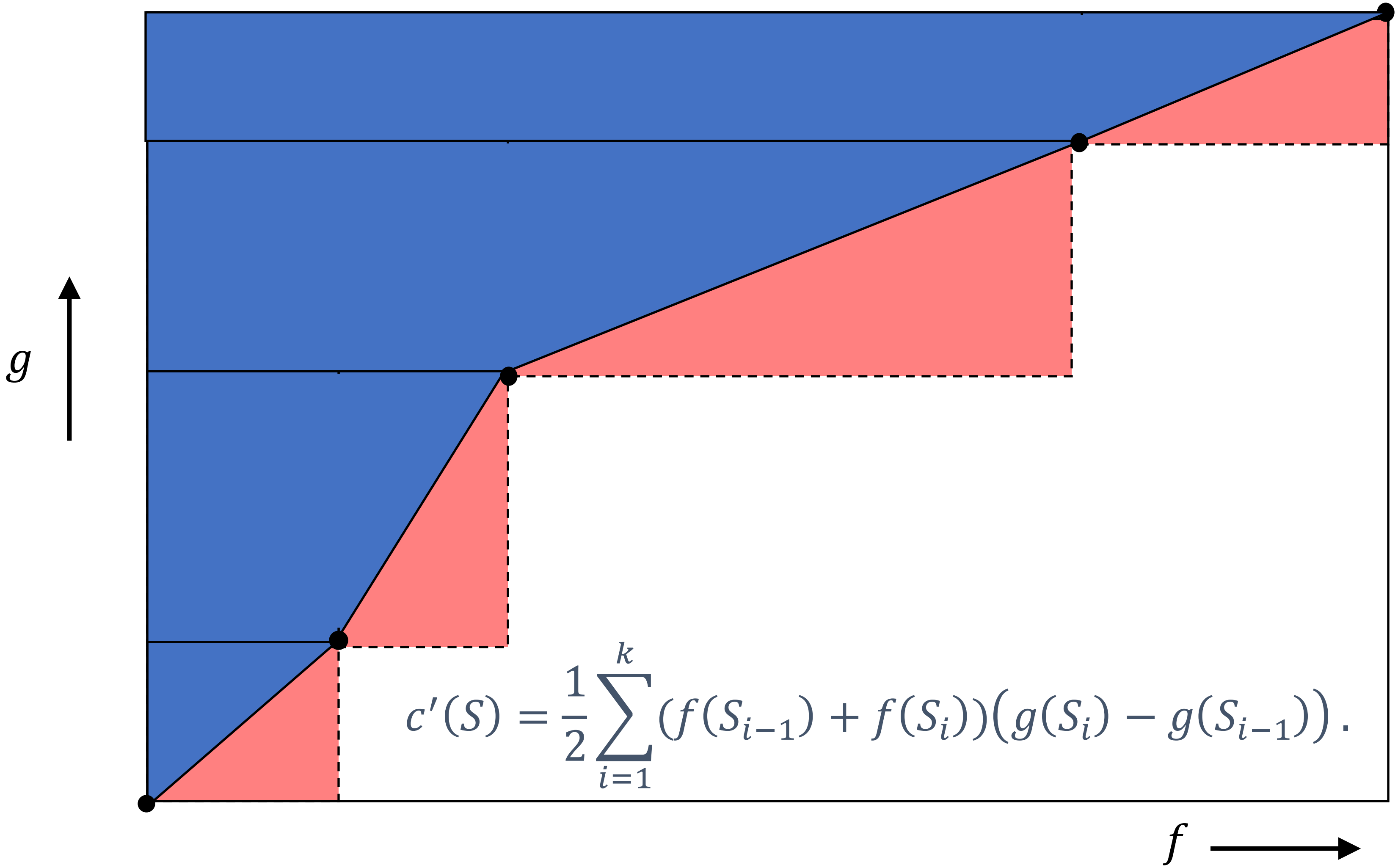
**Theorem (Pisaruk, 1992):** Any solution consistent with a "Sidney decomposition" for this problem is a 2-approximation (and a Sidney decomposition can be found in polynomial time).

# Proof idea



$(f(S_k), g(S_k))$
$(f(S_3), g(S_3))$
$(f(S_2), g(S_2))$
$g$
$(f(S_1), g(S_1))$
$c(S) = \sum_{i=1}^{k} f(S_i)\big(g(S_i) - g(S_{i-1})\big).$
$(f(S_0), g(S_0))$
$f$

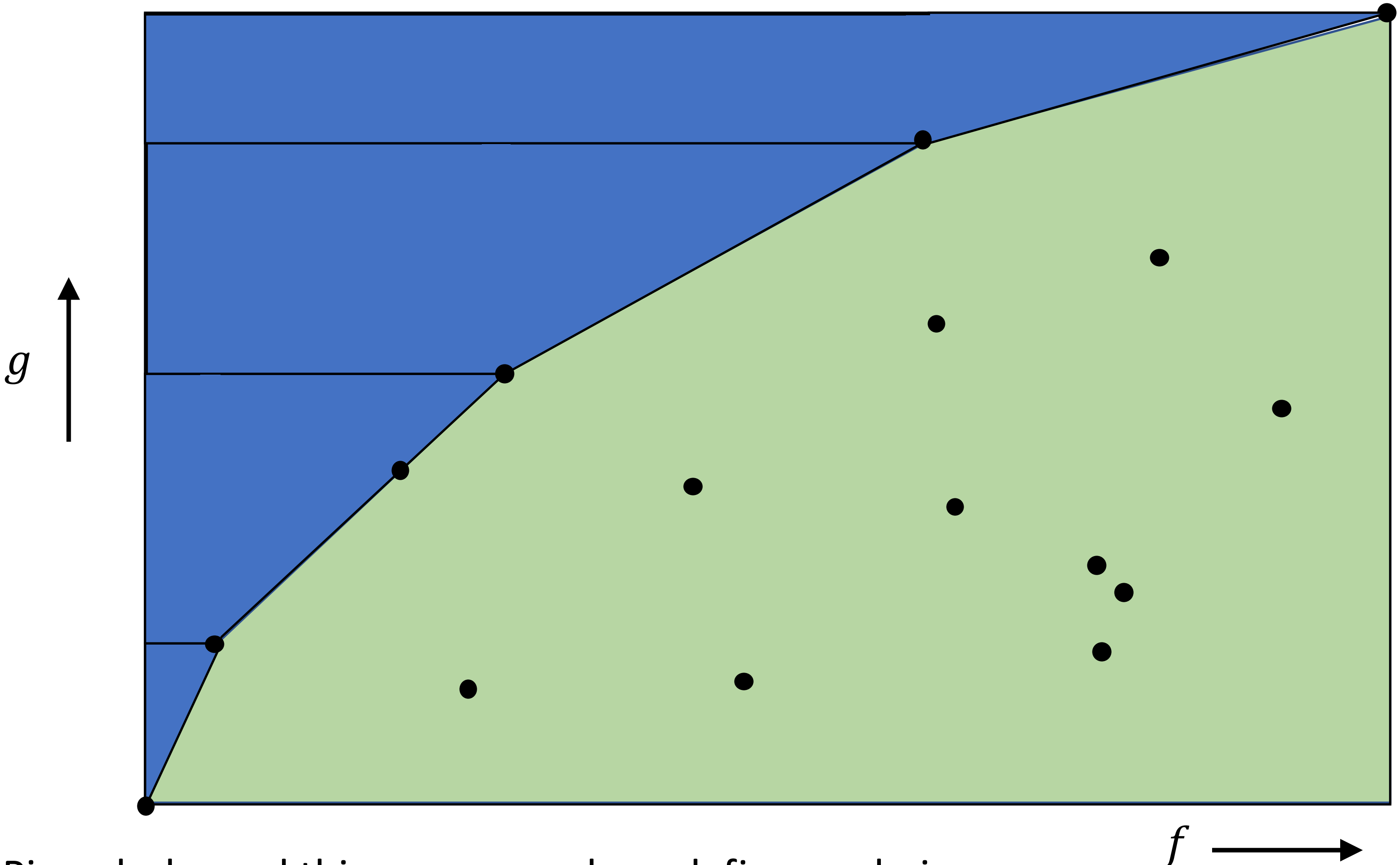
# Proof idea




$$c'(S) = \frac{1}{2}\sum_{i=1}^{k}\left(f(S_{i-1}) + f(S_i)\right)\left(g(S_i) - g(S_{i-1})\right).$$

$c(S) \leq 2c'(S)$ for any $S$

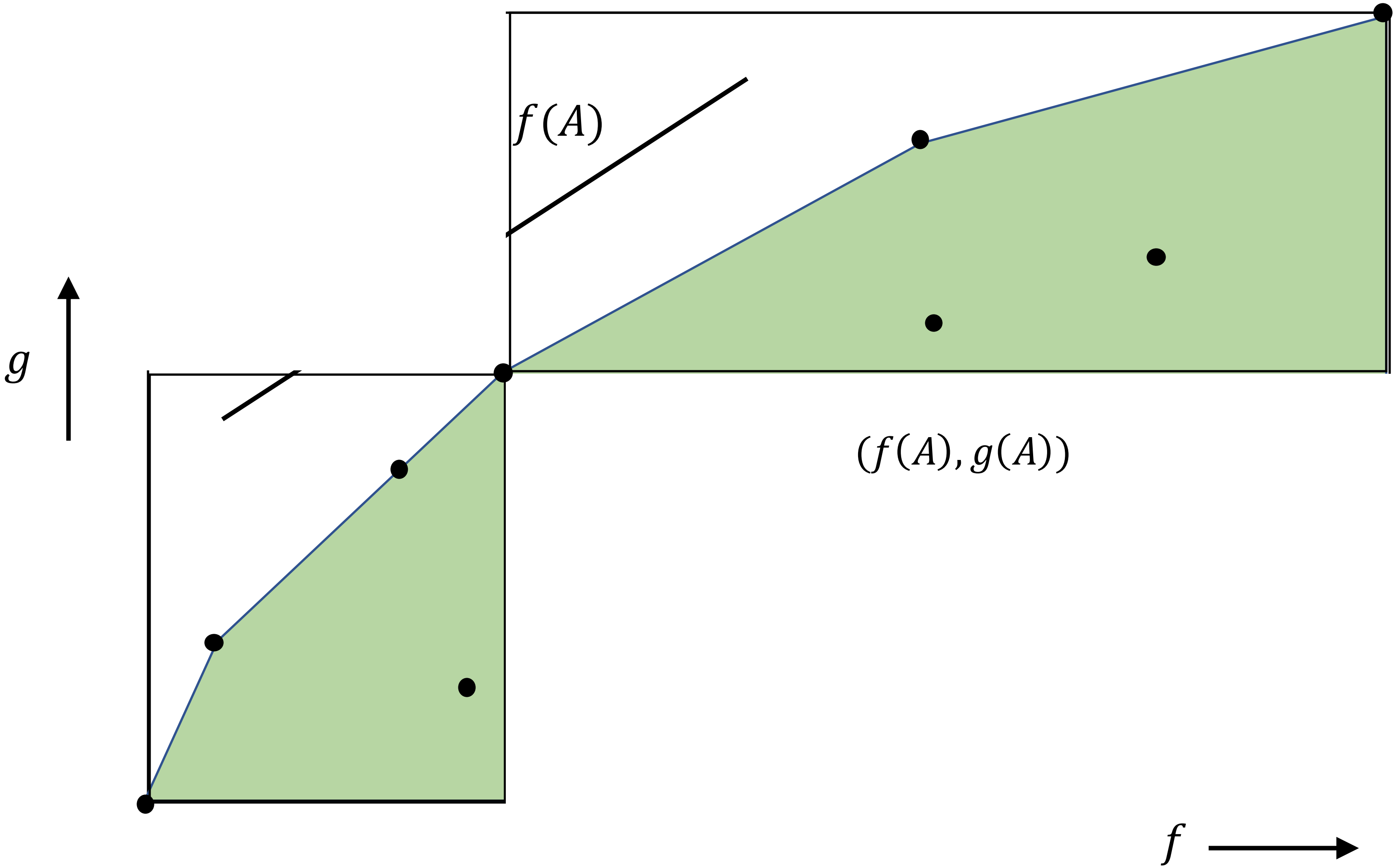
# Solution of problem of minimizing $c'$




Pisaruk showed this upper envelope defines a chain.

**Note**: the sets on the upper envelope define a Sidney decomposition.

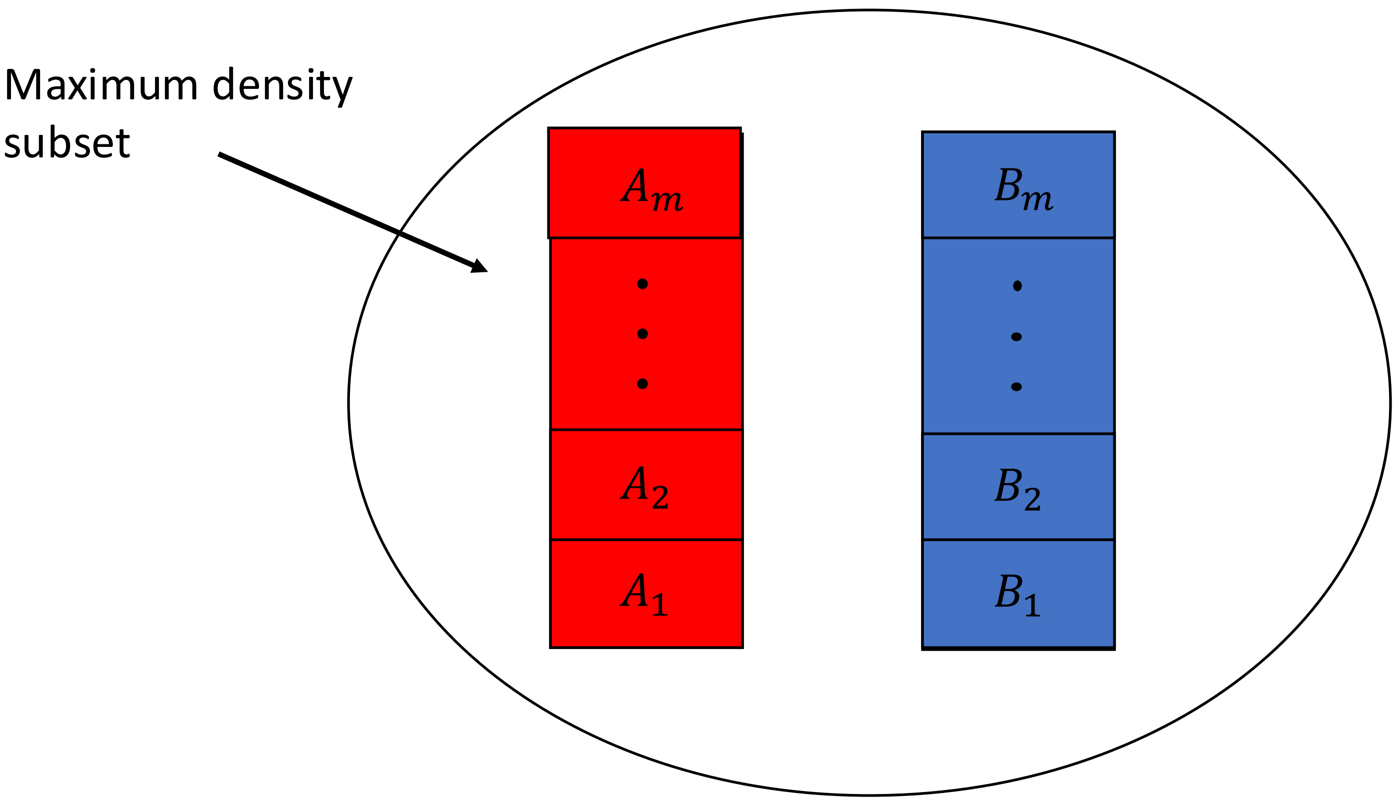
# Finding the upper envelope



# Our approach (Fokkink, L., & Végh, 2019)

**Theorem:** For $A \subset N$, let $g(A)/f(A)$ be the *density* of $A$. If $A$ has maximum density then *any* optimal search starts by searching the elements of $A$ in some order.

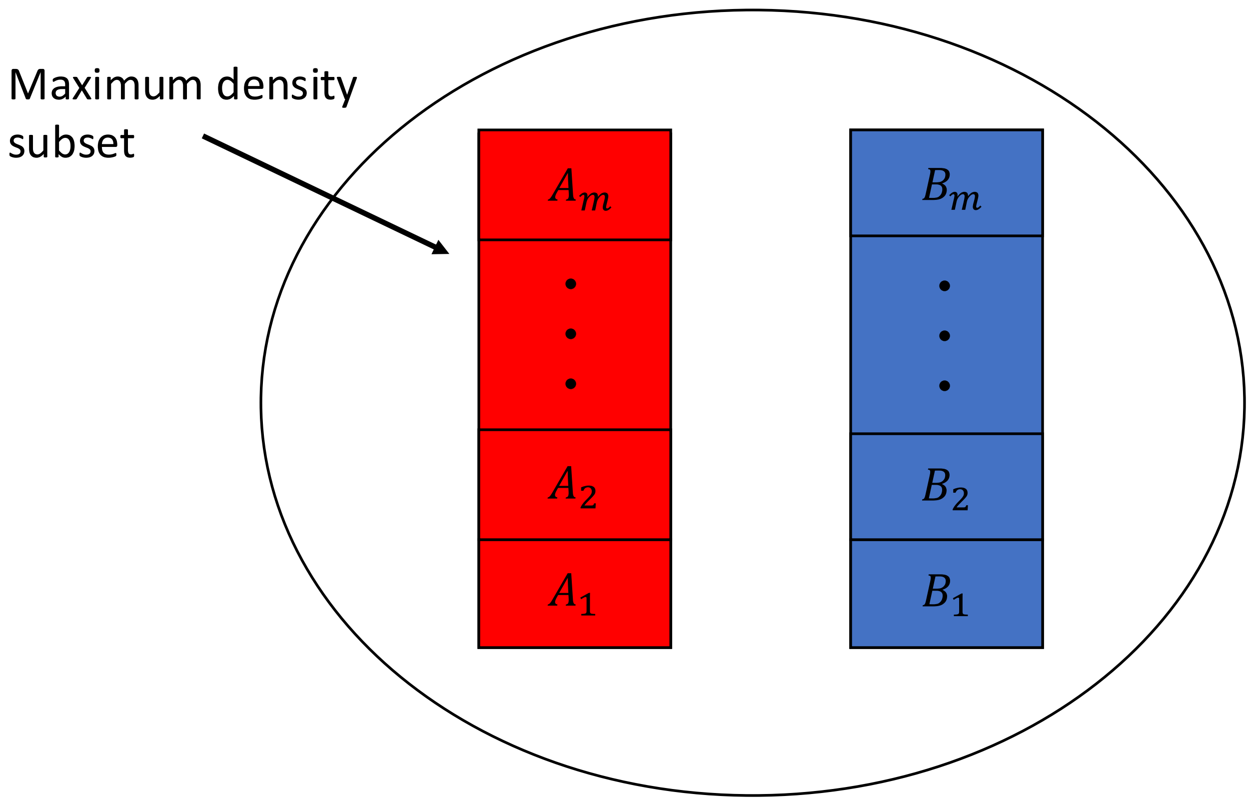
**Idea of proof:** Using "switching argument" to show that any optimal search is just as good as one that starts with a subset of cells with maximum density.



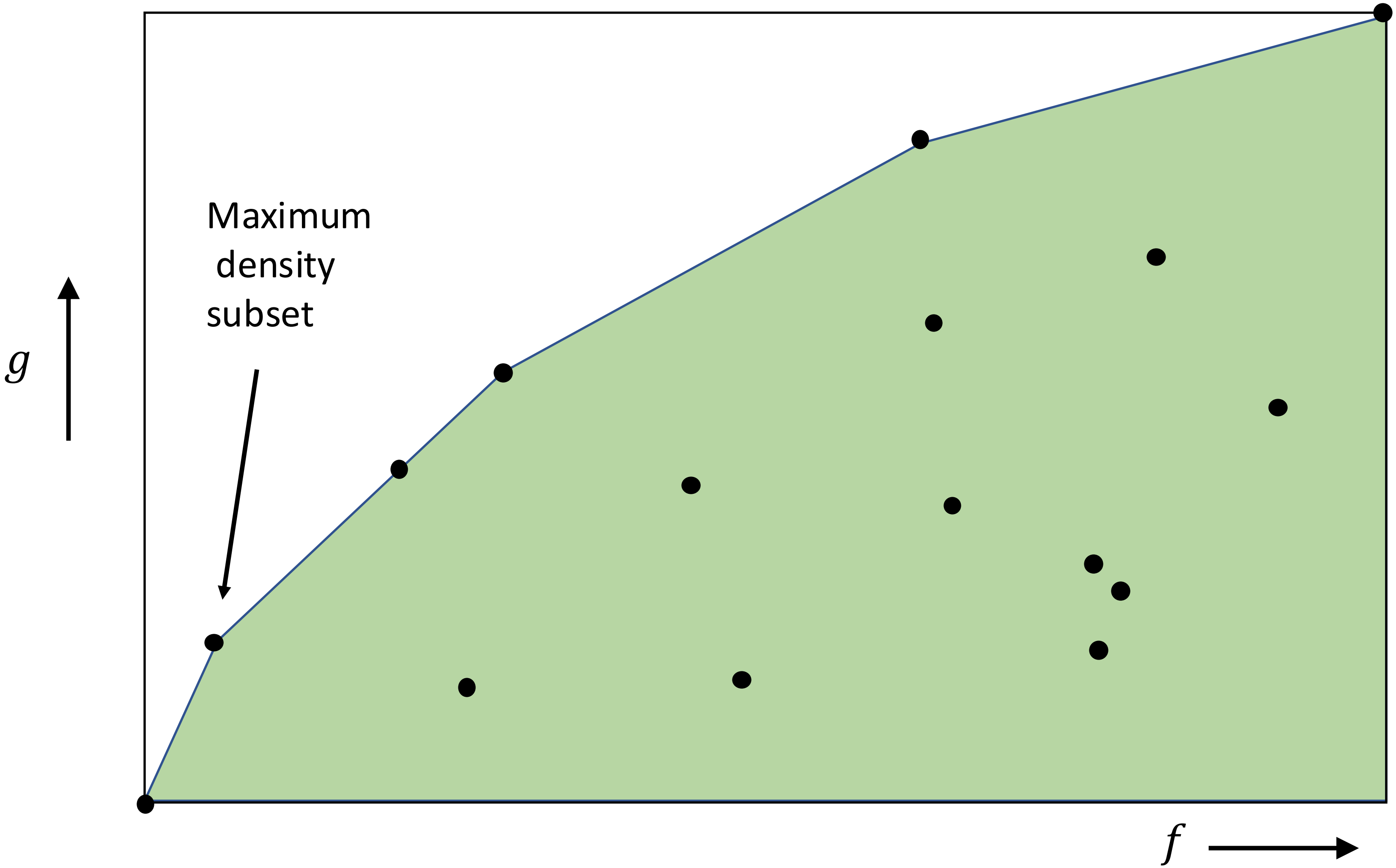
# Our approach (Fokkink, L., & Végh, 2019)

**Theorem:** For $A \subset N$, let $g(A)/f(A)$ be the *density* of $A$. If $A$ has maximum density then *any* optimal search starts by searching the elements of $A$ in some order.

**Idea of proof:** Using “switching argument” to show that any optimal search is just as good as one that starts with a subset of cells with maximum density.




- A max. density subset can be found in polynomial time.
- Using this principle we get a Sidney decomposition.


Maximum
density
subset
$g$
$f$

# Implications for scheduling

- 2-approximation for $1|prec|\sum w_A C_A$.
- Instead of minimizing weighted sum of completions times, we could take some *function* $h$ *of* completion times, i.e. $1|prec|\sum w_A h(C_A)$.
- If $h$ is concave, the cost function $f$ is still submodular, so we get a 2-approximation.
- Note: Schulz and Verschae (2016) get a 2-approximation for the problem $1|prec|\sum w_j h(C_j)$ in a different way, using the Sidney decomposition for the problem $1|prec|\sum w_j C_j$.

# Functions of low curvature

**Definition:** The total curvature of a set function $f$ on $N$ is

$$k = 1 - \min_{i \in N} \frac{f(N) - f(N \setminus \{i\})}{f(\{i\})}$$

If $f$ is submodular, $k$ is in $[0,1]$.
Let $g^\sharp$ be the dual function of $g$, given by $g^\sharp(A) = g(N) - g(\bar{A})$. It is submodular if $g$ is supermodular.
Let $k_f$ be the total curvature of $f$ and let $k_g$ be the total curvature of $g^\sharp$.

**Theorem:** Let $\theta = (1 - k_f)(1 - k_g)$. Our algorithm for the submodular search problem has approximation ratio $\frac{2}{1+\delta}$ where

$$\delta = \min\left\{\theta, \frac{2\theta \max\{1 - k_f, 1 - k_g\}}{1 + \theta}\right\}.$$

# Implications for scheduling

- Consider the problem $1||\sum w_j h(C_j)$ with $h$ concave.
- $k_g = 0$ and

$$1 - k_f = \min_{\mathrm{i} \in N} \frac{f(N) - f(N - \{i\})}{f(\{i\})} \geq \inf_{x \in [0,1]} \frac{1 - h(1 - x)}{h(x)} = \frac{h'(1)}{h'(0)}$$

- So approximation ratio is $\frac{2}{1+\theta} \leq \frac{2}{1+\frac{h'(1)}{h'(0)}}$.

- Eg. If $h(x) = \log(1 + ax)$ , $a > 0$ then $\frac{h'(1)}{h'(0)} = \frac{1}{1+a}$ and the approximation ratio is $1 + \frac{a}{2+a}$.

- Eg. If $h(x) = 1 - \exp(-rx)$ , $r > 0$ then $\frac{h'(1)}{h'(0)} = e^{-r}$, and the approximation ratio is $\frac{2}{1+e^{-r}}$ .

# MSOP for supermodular $f$

Previous work:

- $f$ supermodular and $g$ the cardinality function: 4-approximation in Iwata, Tetali and Tripathi (2012)

- $f$ supermodular and $g$ modular: 4-approximation in Streeter and Golovin (2008)

Proofs inspired by 4-approximation for min-sum set cover in Feige, Lovász and Tetali (2004).

# Main theorem (Happach, Hellerstein & L., 2022)

**Definition:** An $\alpha$-approximate Sidney decomposition is one such that at each step, the next element of the decomposition is chosen to approximate the density problem within a factor $\alpha$.

**Theorem:** Suppose $f$ is subadditive. Then for any $\alpha$, an $\alpha$-approximate Sidney decomposition is a $4\alpha$-approximation for an optimal chain for MSOP.

$f(S \cup T) \leq f(S) + f(T)$ for disjoint $S, T$

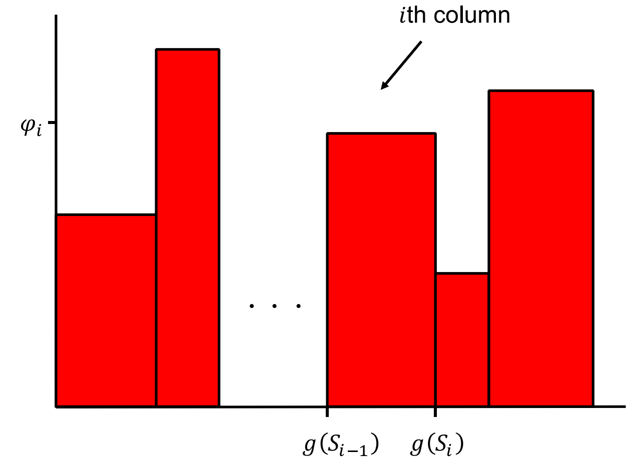
# Idea of proof of main theorem ($\alpha = 1$)

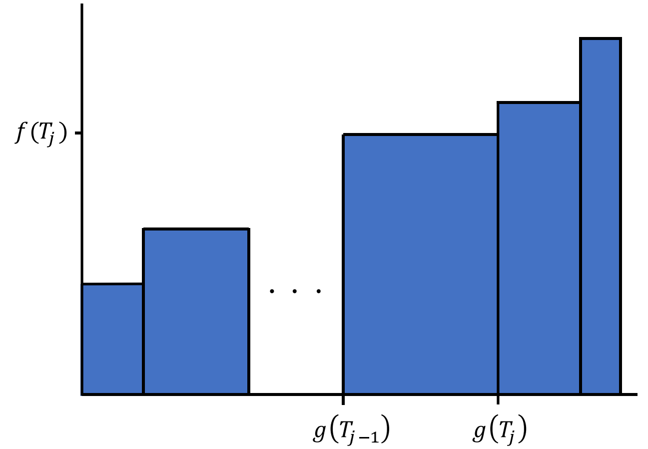
- Let $S_0, S_1, \ldots, S_k$ be a greedy chain
- Let $T_0, T_1, \ldots, T_\ell$ be an optimal chain
- Rewrite objective function for greedy chain as

$$\sum_{i=1}^{k} \varphi_i \big(g(S_i) - g(S_{i-1})\big),$$

where $\varphi_i = \big(g(V) - g(S_{i-1})\big)\frac{f(S_i)-f(S_{i-1})}{g(S_i)-g(S_{i-1})}$

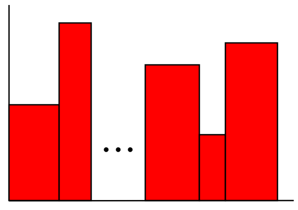
- Draw a histogram for the optimal chain in the "usual way"
- Scale the red histogram by a factor of ½ in the horizontal and vertical directions and show that it fits into the blue histogram



# Idea of proof of main theorem ($\alpha = 1$)

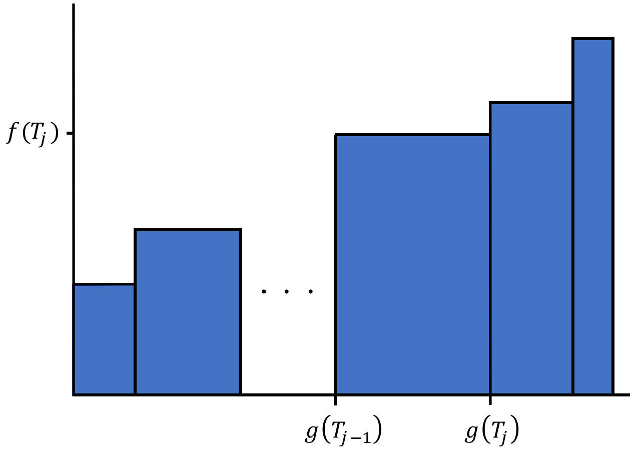
- Let $S_0, S_1, \dots, S_k$ be a greedy chain
- Let $T_0, T_1, \dots, T_\ell$ be an optimal chain
- Rewrite objective function for greedy chain as



$$\sum_{i=1}^{k} \varphi_i\big(g(S_i) - g(S_{i-1})\big),$$

where $\varphi_i = \big(g(V) - g(S_{i-1})\big)\frac{f(S_i)-f(S_{i-1})}{g(S_i)-g(S_{i-1})}$

- Draw a histogram for the optimal chain in the "usual way"
- Scale the red histogram by a factor of ½ in the horizontal and vertical directions and show that it fits into the blue histogram



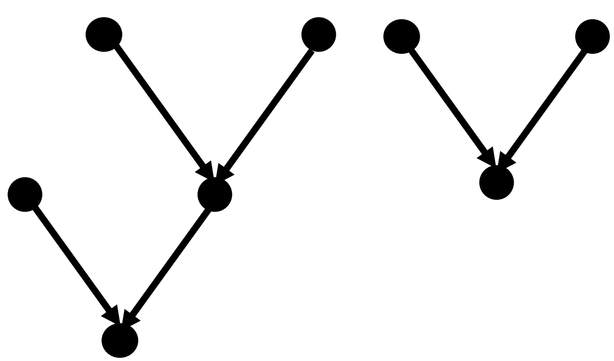
# Known applications

1. 4-approximation for $f$ supermodular and $g$ cardinality function in Iwata, Tetali and Tripathi (2012)
2. 4-approximation for $f$ supermodular and $g$ modular in Streeter and Golovin (2008)
3. 8-approximation for *expanding search* problem in Hermans, Leus and Matuschke (2021)
4. 4-approximation for bipartite OR-scheduling in Happach and Schulz (2020)

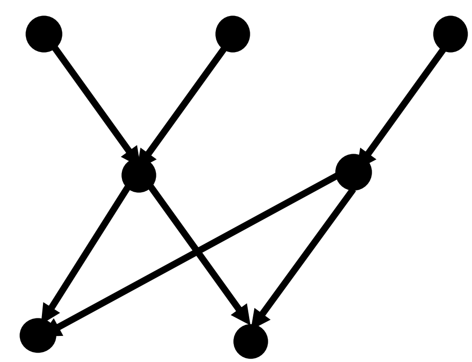
# New application: OR-scheduling

- Jobs correspond to nodes of DAG
- A job can only be completed when *at least one* of its predecessors has been completed

**Theorem:** There is a polynomial time 4-approximation algorithm for the problem of minimizing the sum of weighted completion times of a set of jobs that must be scheduled to respect some OR-precedence constraints in the form of an *inforest* (or, more generally, a *multitree*).



an *inforest*



a multitree

# Future work

- More work needed on the “density problem”
- Could a 4-approximation be found for *Generalized Min-Sum Set Cover?* Best known approximation is 4.642 (Bansal, Batra, Farhadi, Tetali, 2021)