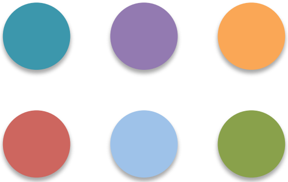
Greedy Heuristics for Set Cover

Eirini Asteri, Jessica Hoffmann

University of Texas, Austin

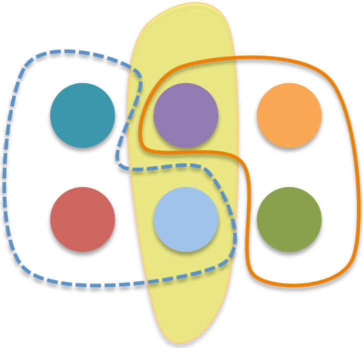
Outline

Set Cover Problem & Greedy Algorithm



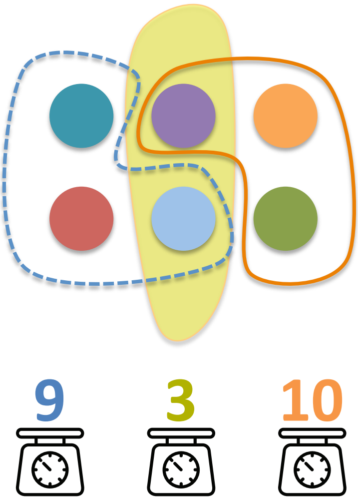
- Number of distinct elements e_i

Set Cover Problem & Greedy Algorithm



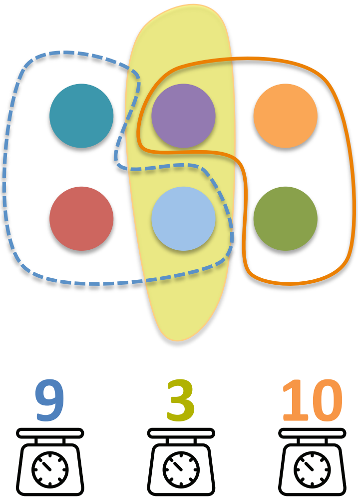
- Number of distinct elements e_i
- Number of sets S_j

Set Cover Problem & Greedy Algorithm



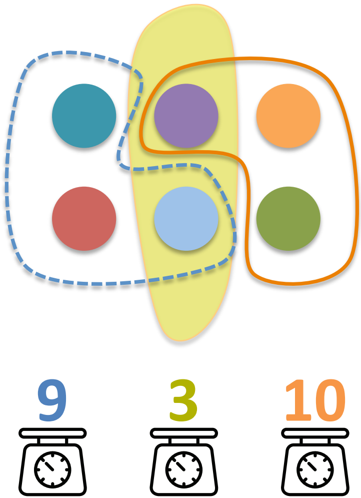
- Number of distinct elements e_i
- Number of sets S_j
- A weight for each set w_j

Set Cover Problem & Greedy Algorithm



- Number of distinct elements e_i
- Number of sets S_j
- A weight for each set w_j
- **Goal** Find set cover with minimum weight

Set Cover Problem & Greedy Algorithm



- Number of distinct elements e_i
- Number of sets S_j
- A weight for each set w_j
- **Goal** Find set cover with minimum weight
- **Greedy Choice**

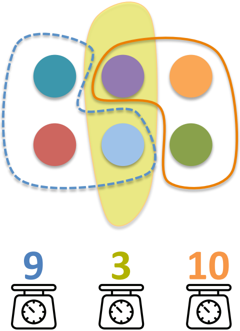
$$best\_set = \arg\min_{l} \frac{w_l}{|\hat{S}_l|}$$

Basic Preprocessing

Get rid of redundant sets 🗑️🧹

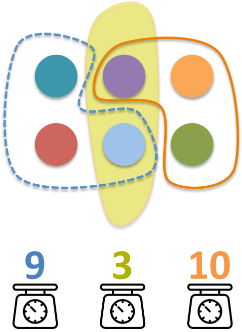
If $S_{small} \subseteq S_{big}$ and weight$(S_{small}) \geq$ weight(S_{big}) then S_{small} is a redundant set.

Heuristics, Intuition



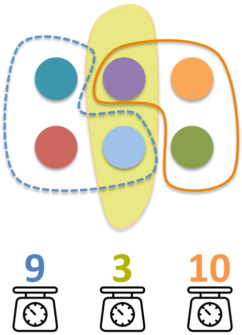
- Elements with frequency 1 should be covered first

Heuristics, Intuition



- Elements with frequency 1 should be covered first
- Extend idea to **"infrequent"** elements

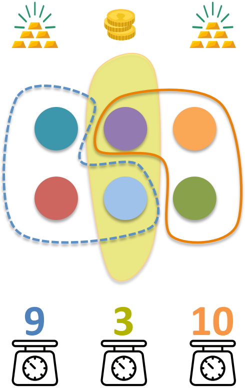
Heuristics, Intuition



- Elements with frequency 1 should be covered first
- Extend idea to **“infrequent”** elements
- Assign a value to each element

$$\text{value(element)} = \frac{1}{\text{frequency-1}}$$

Heuristics, Intuition



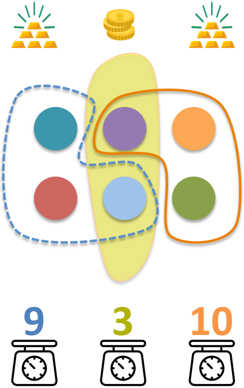
- Elements with frequency 1 should be covered first
- Extend idea to **"infrequent"** elements
- Assign a value to each element

$$\text{value(element)} = \frac{1}{\text{frequency-1}}$$

- Assign value to each set

$$\text{value(set)} = \sum \text{value(element)}$$

Heuristics, Intuition



- Elements with frequency 1 should be covered first
- Extend idea to **"infrequent"** elements
- Assign a value to each element

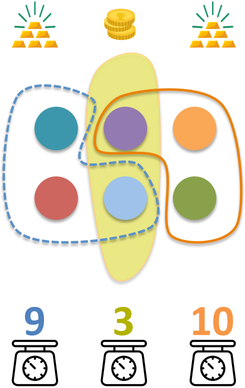
$$\text{value(element)} = \frac{1}{\text{frequency-1}}$$

- Assign value to each set

$$\text{value(set)} = \sum \text{value(element)}$$

- Choose a set with **small weight** and **large value!**

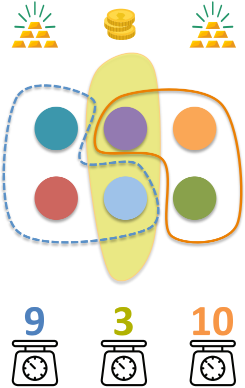
Heuristics, General Framework



- New Greedy Choice

$$best\_set = \arg\min_j \frac{w_j}{v_j} = \arg\min_j \frac{w_j}{\sum \text{value}(e_i)}$$

Heuristics, General Framework



- New Greedy Choice

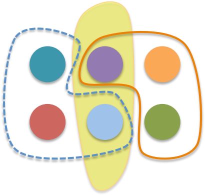
$$best\_set = \arg\min_j \frac{w_j}{v_j} = \arg\min_j \frac{w_j}{\sum \text{value}(e_i)}$$

- Regular Greedy is a Special Case

$$\text{if value}(e_i) = 1 \rightarrow \frac{w_j}{\sum \text{value}(e_i)} = \frac{w_j}{|\hat{S}_j|}$$

Another Heuristic Value Function..

Dig Deeper, Extract more Information



$$\text{value}(e_i) = \frac{\sum_{S_j : e_i \in S_j} \text{average\_weight}(S_j)}{\text{frequency}(e_i) - 1}$$

[Intuition]

- An element is valuable if it is contained in "expensive" sets.
- Choose a "**cheap**" set that contains elements that are "**expensive in the market**".

Theoretical Analysis

Approximation result

Any greedy heuristic, caracterized by its value() function, is a $\frac{\max\limits_{e_i} value(e_i)}{\min\limits_{e_i} value(e_i)} \cdot H_n$-approximation.

Remark: This shows any greedy heuristics will have worse theoretical guarantees if our analysis is tight. The proof follows the same ideas as in our textbook [1].

Value Functions Tested

- Greedy: $v(e_i) = 1$
- H 1: $v(e_i) = \dfrac{1}{f_i - 1}$
- H 2: $v(e_i) = 1 + \dfrac{1}{f_i - 1}$
- H 3: $v(e_i) = exp(-f_i)$
- H 4: $v(e_i) = \dfrac{|\hat{S}_j|}{f_i - 1}$
- H 7: $v(e_i) = \dfrac{1}{(f_i - 1)^2}$
- H 8: $v(e_i) = \dfrac{1}{(f_i - 1)^3}$
- H 9: $v(e_i) = \dfrac{1}{\sqrt{f_i - 1}}$
- H 10: $v(e_i) = \dfrac{\sum w_j / |\hat{S}_j|}{f_i - 1}$
- H 11:

  $v(e_i) = c + \dfrac{\sum w_j / |\hat{S}_j|}{f_i - 1}$

Data Sets

Datasets found at
`people.brunel.ac.uk/~mastjjb/jeb/orlib/scpinfo.html`
Description: OR-Library is a collection of test data sets for a variety of OR problems
Officious description: This is the most complete benchmark we found for datasets for set cover, used for instance in [2]
TL;DR: 24 datasets, around 100 elements, around 2000 sets $\rightarrow$ naive brute force won't work.

Experimental Results

Heuristics	$\frac{1}{f_i - 1}$	$1 + \frac{1}{f_i - 1}$	$\frac{1}{(f_i - 1)^2}$	$\frac{1}{(f_i - 1)^3}$	$\frac{1}{\sqrt{f_i} - 1}$	valuation-mixed	Greedy
Dataset 1	477	461	477	477	461	463	463
Dataset 2	566	572	580	588	572	580	582
Dataset 3	564	589	552	547	582	596	598
Dataset 4	540	541	561	550	541	547	548
Dataset 5	575	577	573	567	584	577	577
Dataset 6	596	606	580	588	606	606	615
Dataset 7	480	474	461	466	481	476	476
Dataset 8	542	533	542	548	538	537	533
Dataset 9	747	744	732	722	746	747	747
Dataset 10	290	291	291	290	291	292	289
Dataset 11	345	343	339	341	343	342	348
Dataset 12	246	246	245	252	246	246	246
Dataset 13	262	266	265	257	266	267	265
Dataset 14	234	234	235	235	234	233	236
Dataset 15	250	250	244	242	250	245	251
Dataset 16	317	315	311	310	315	320	326
Dataset 17	313	313	314	317	313	313	323
Dataset 18	304	308	307	316	308	304	312
Dataset 19	159	160	164	163	159	157	159
Dataset 20	171	170	172	176	171	170	170
Dataset 21	161	156	159	159	163	156	161
Dataset 22	149	149	149	148	149	145	149
Dataset 23	195	191	194	203	195	192	196
Dataset 24	545	548	565	554	557	550	556

Experimental Results

Heuristics	$\frac{1}{f_i - 1}$	$1 + \frac{1}{f_i - 1}$	$\frac{1}{(f_i - 1)^2}$	$\frac{1}{(f_i - 1)^3}$	$\frac{1}{\sqrt{f_i} - 1}$	valuation-mixed	Greedy
Dataset 1	477	461	477	477	461	463	463
Dataset 2	566	572	580	588	572	580	582
Dataset 3	564	589	552	547	582	596	598
Dataset 4	540	541	561	550	541	547	548
Dataset 5	575	577	573	567	584	577	577
Dataset 6	596	606	580	588	606	606	615
Dataset 7	480	474	461	466	481	476	476
Dataset 8	542	533	542	548	538	537	533
Dataset 9	747	744	732	722	746	747	747
Dataset 10	290	291	291	290	291	292	289
Dataset 11	345	343	339	341	343	342	348
Dataset 12	246	246	245	252	246	246	246
Dataset 13	262	266	265	257	266	267	265
Dataset 14	234	234	235	235	234	233	236
Dataset 15	250	250	244	242	250	245	251
Dataset 16	317	315	311	310	315	320	326
Dataset 17	313	313	314	317	313	313	323
Dataset 18	304	308	307	316	308	304	312
Dataset 19	159	160	164	163	159	157	159
Dataset 20	171	170	172	176	171	170	170
Dataset 21	161	156	159	159	163	156	161
Dataset 22	149	149	149	148	149	145	149
Dataset 23	195	191	194	203	195	192	196
Dataset 24	545	548	565	554	557	550	556

- Most of our heuristics are better than Greedy most of the time

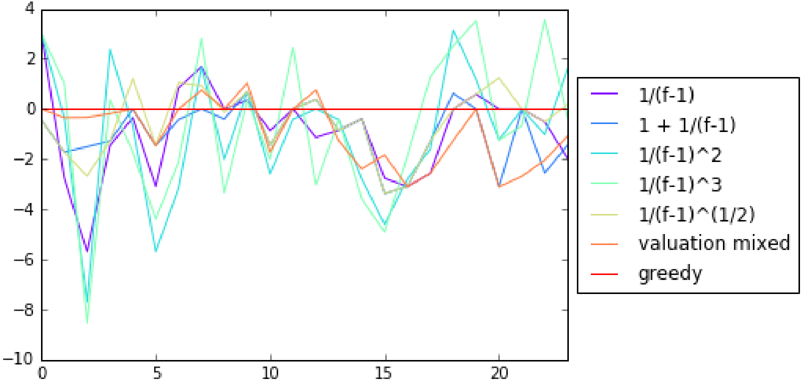
Experimental Results

Heuristics	$\frac{1}{f_i-1}$	$1+\frac{1}{f_i-1}$	$\frac{1}{(f_i-1)^2}$	$\frac{1}{(f_i-1)^3}$	$\frac{1}{\sqrt{f_i-1}}$	valuation-mixed	Greedy
Dataset 1	477	461	477	477	461	463	463
Dataset 2	566	572	580	588	572	580	582
Dataset 3	564	589	552	547	582	596	598
Dataset 4	540	541	561	550	541	547	548
Dataset 5	575	577	573	567	584	577	577
Dataset 6	596	606	580	588	606	606	615
Dataset 7	480	474	461	466	481	476	476
Dataset 8	542	533	542	548	538	537	533
Dataset 9	747	744	732	722	746	747	747
Dataset 10	290	291	291	290	291	292	289
Dataset 11	345	343	339	341	343	342	348
Dataset 12	246	246	245	252	246	246	246
Dataset 13	262	266	265	257	266	267	265
Dataset 14	234	234	235	235	234	233	236
Dataset 15	250	250	244	242	250	245	251
Dataset 16	317	315	311	310	315	320	326
Dataset 17	313	313	314	317	313	313	323
Dataset 18	304	308	307	316	308	304	312
Dataset 19	159	160	164	163	159	157	159
Dataset 20	171	170	172	176	171	170	170
Dataset 21	161	156	159	159	163	156	161
Dataset 22	149	149	149	148	149	145	149
Dataset 23	195	191	194	203	195	192	196
Dataset 24	545	548	565	554	557	550	556

- Most of our heuristics are better than Greedy most of the time
- The best heuristic varies a lot across datasets

Experimental Results

Here are the gains in percents of the greedy objective, for the top 6 heuristics:



The best algorithm achieves a gain of **-0.96 %**. If we combine all the algorithms, we have an average gain of **-2.88 %**.

References

- [1] Williamson, David P. Shmoys, David B., The Design of Approximation Algorithms, Cambridge University Press, 2011
- [2] David Kordalewski, New Greedy Heuristics For Set Cover and Set Packing,CoRR, abs/1305.3584,2013.