WEIGHTED SET COVER

By: Ali Mowazi, Mark Kasun, Jeremiah O’Neil

INTRODUCTION TO WEIGHTED SET COVER

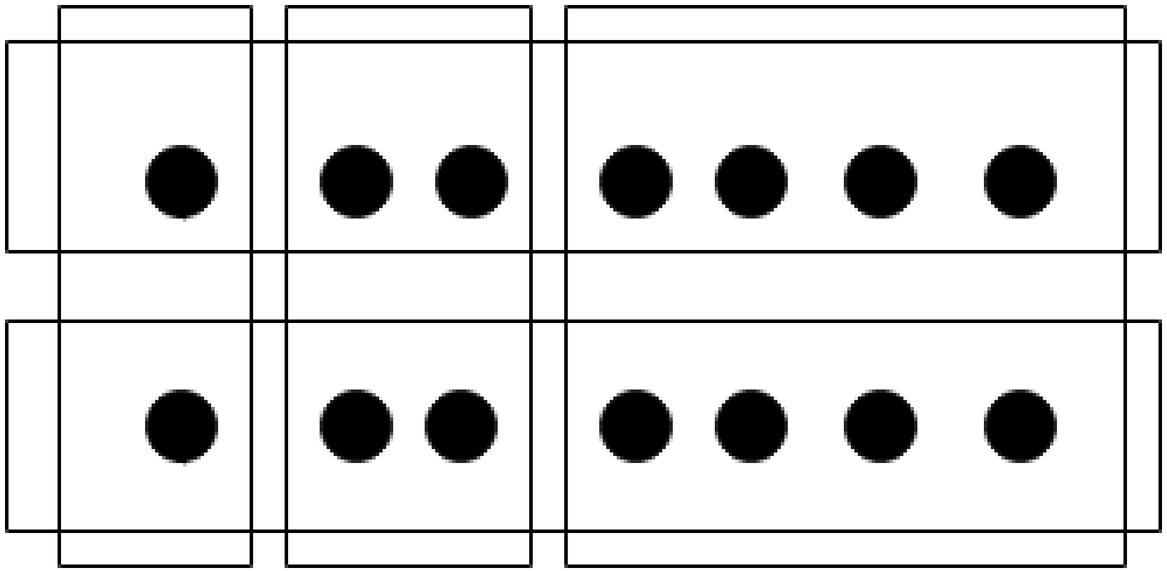
Weighted Set Cover in decision form is NP- Complete.

Given a Set E = $\{e_{1,}\ e_{2,}\ \ldots\ ,e_n\}$ of n elements, a collection of m subsets S_1, S_2, … ,S_m of E with weights $w_1,w_2,\ldots,w_m$. Goal to find a set I $\subseteq$ $\{S_i,\ldots,S_m\}$ such that all elements covered by I and the sum of the weights in I is minimized.

In Decision form: Can we find a set cover with weight at most W.

The study of weighted set cover led to development of fundamental techniques for the entire field of approximation algorithms.

EXAMPLE



Assume weight of each subset is 1

Find the minimal set cover.

In decision form it is: Does there exist a set cover of weight W?

OUR STRATEGIES FOR SOLVING SET COVER

Greedy Approximation Algorithm

Greedy Heuristic

Simulating Annealing (In Progress)

GREEDY APPROXIMATION ALGORITHM

The algorithm runs in O(n*log(n)) time and is quite simple to implement

Let C be a set of elements covered, U be the set of all elements, and X to be a list of sub-sets picked

Initialize C={}, X=[]

While |C| =/= |U|

- Find sub-set S with smallest cost effectiveness (cost of S divided by uncovered elements)
- Set C=C U S

Output X

GREEDY APPROXIMATION GUARANTEE

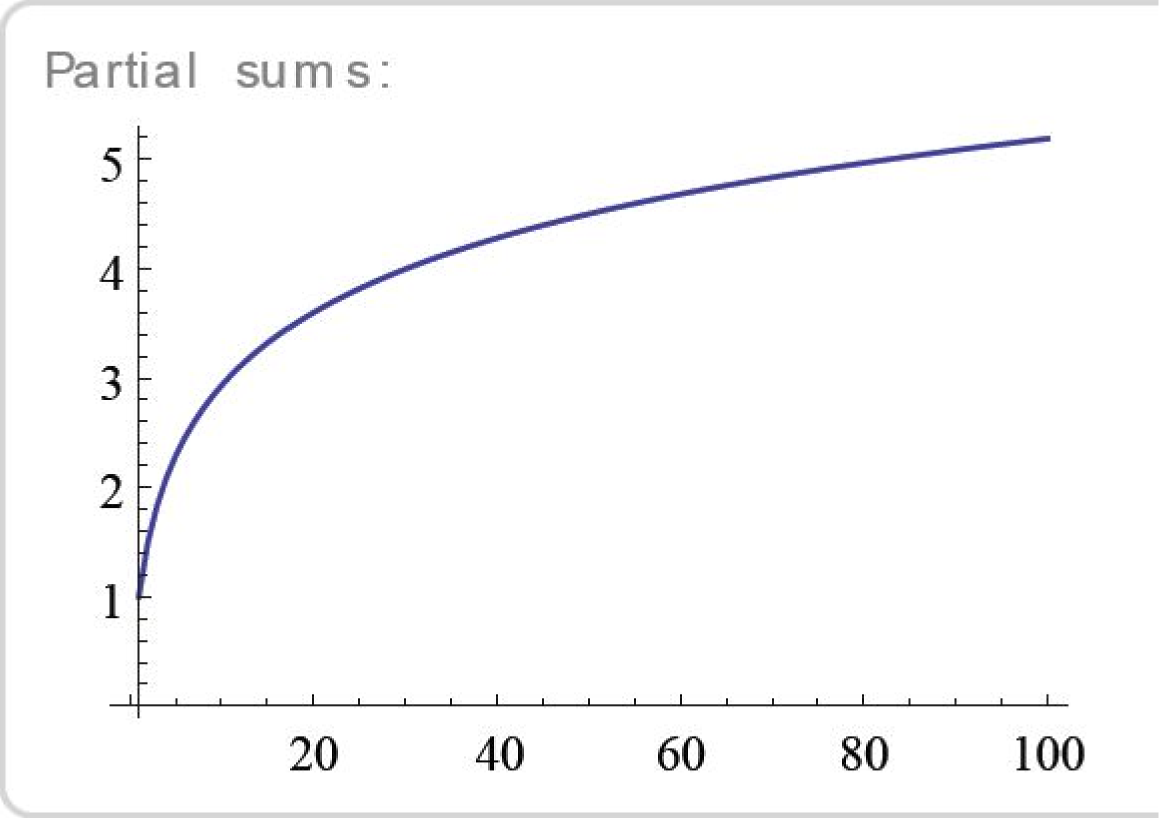
Greedy algorithm guarantees cost <= H_k*OPT

k is the size of the largest subset

At k=4, α>2

At k=11, α>3

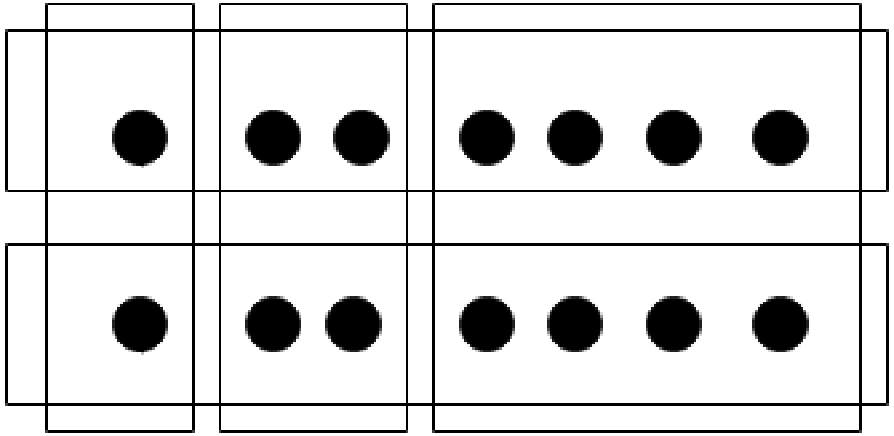
$$\alpha = H_k = \sum_{i=1}^{k} \frac{1}{i} \leq 1 + \log(k)$$



GREEDY APPROXIMATION EXAMPLE

OPT = 2, Greedy solution = 3

H_8*OPT = 5.4 (total sets = 5)



GREEDY HEURISTIC

Intended to quickly find an initial feasible solution and to construct neighbouring solutions for Simulated Annealing in a randomized manner. We use a fast heuristic proposed by Balas and Ho in 1987.

Initialize C={} and X=[]

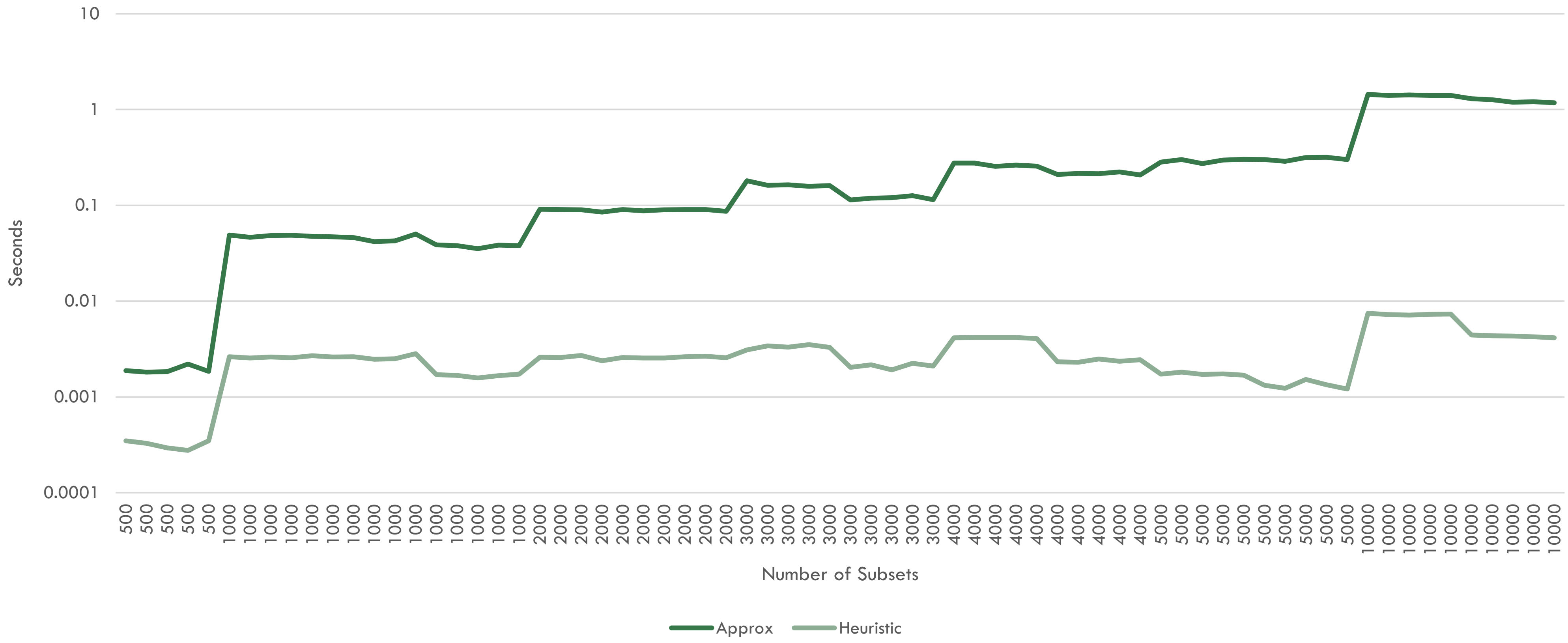
While |C| =/= |U|

- Pick a random element in C, and find the set S of minimal cost which covers the element
- Set C=C ∪ S and append S to X

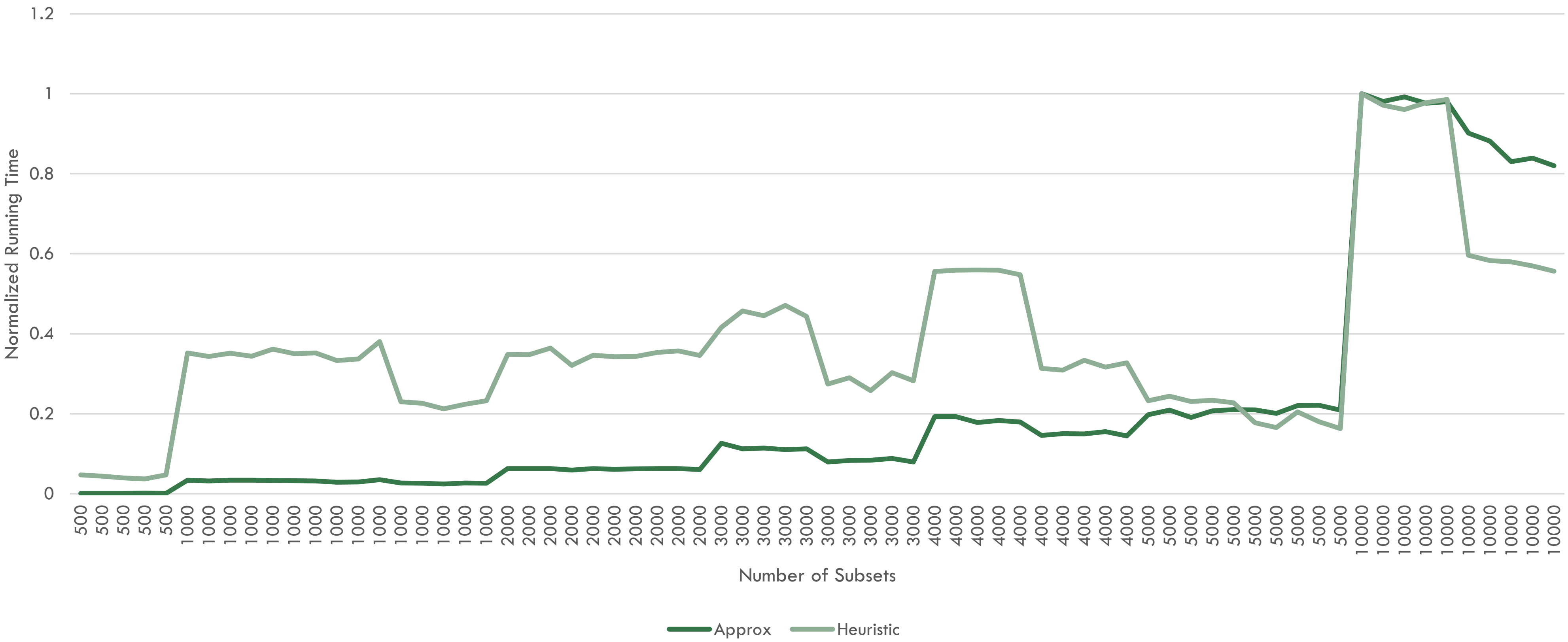
Examine each selected set, in order by recency of selection. If the set is redundant – it can be removed without leaving an element uncovered -- remove it.

Return the cover X.

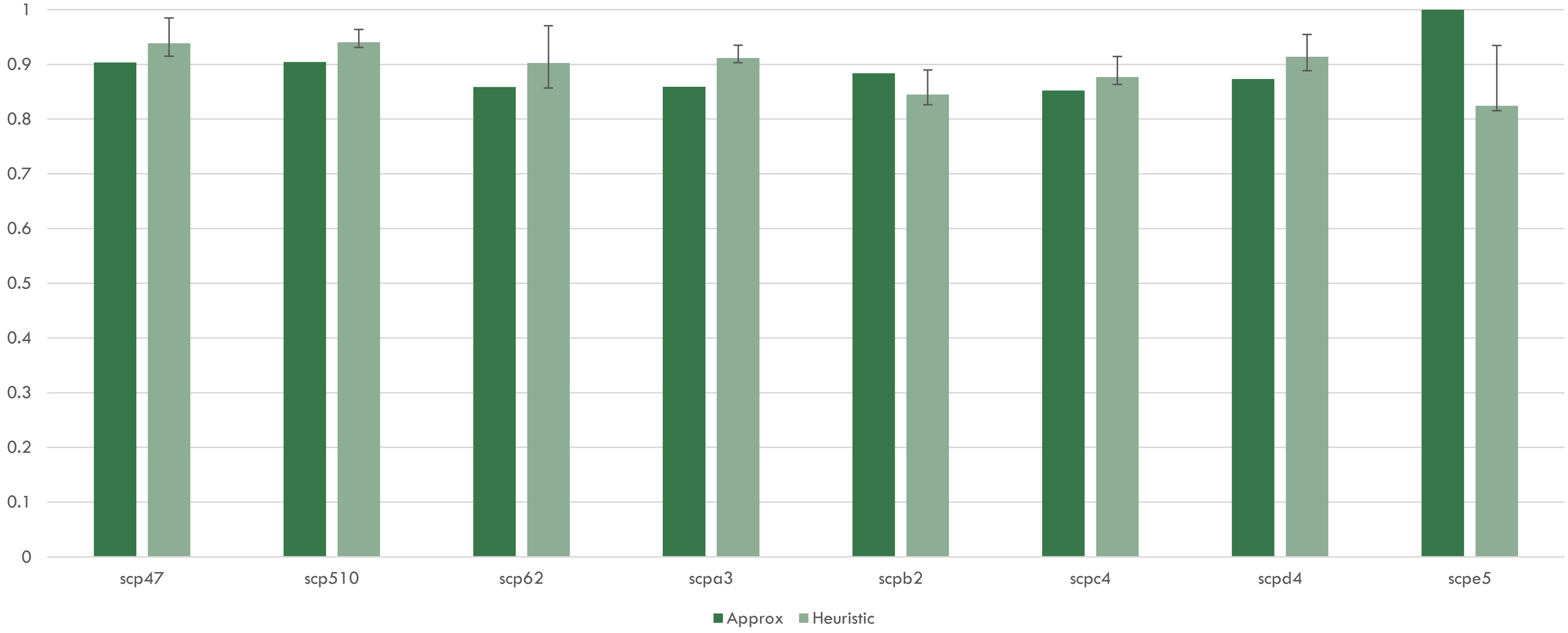
RUNNING TIMES



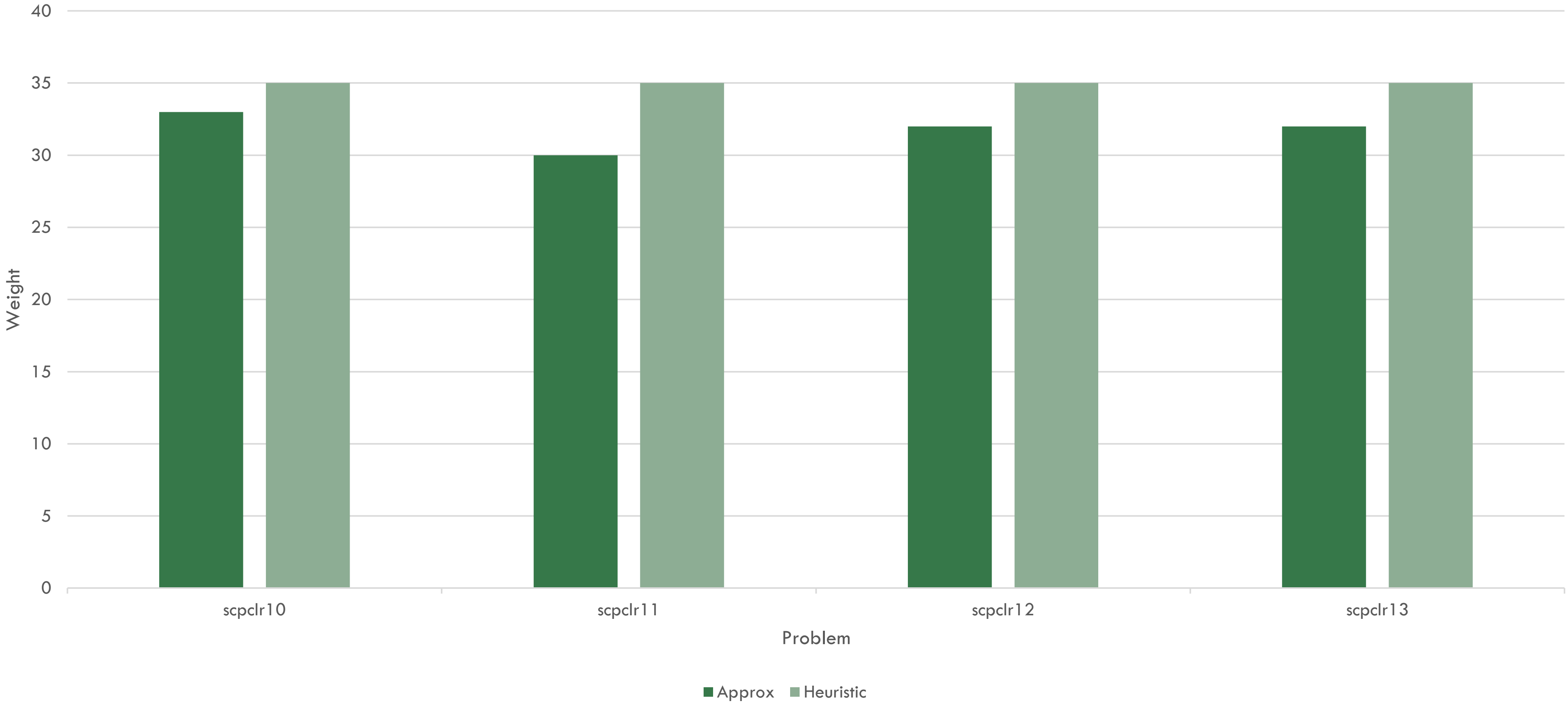
RUNNING TIME NORMALIZED



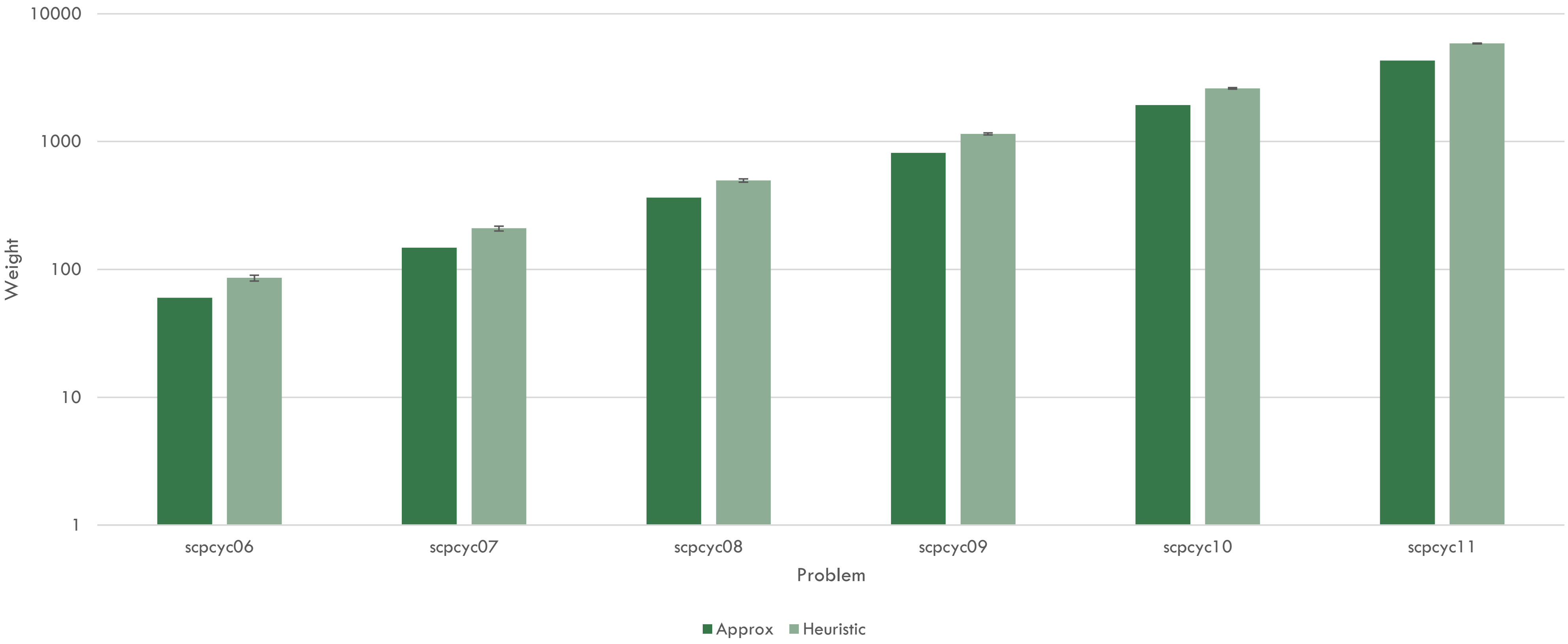
RESULTS AGAINST SCP DATA SET (%OPTIMAL)



EDGES REQUIRED TO HIT EVERY 4-CYCLE IN A HYPERCUBE



4-TUPLES FORMING THE SMALLEST NON-BI-CHROMATIC HYPERGRAPH



SIMULATED ANNEALING

```
GENERATE an initial solution X
for T in schedule:
    SEARCH for a neighbour solution X'
    let deltaE = W(X) - W(X')
    if deltaE < 0 or rand(0, 1] < exp(-deltaE/T)
        let X = X'
return X
```

GENERATE AND SEARCH SUBPROCEDURES.

These subprocedures respectively define the initial state and neighbourhood of states. We base both on the fast heuristic.

The GENERATE subprocedure just runs the heuristic from C={} and X=[].

The SEARCH subprocedure randomly removes sets from a feasible cover X, creating a partial cover which is then completed by the heuristic.

OTHER METHODS FOR SOLVING SET COVER

Linear Programming

Ant Colony Optimization

Beasely and Chu Genetic Algorithm

Haddadi Langrangian Heuristic

Aickelin Indirect Genetic Algorithm

APPLICATIONS OF SET COVER

IBM finds computer viruses

5000 known viruses (elements)

9000 substrings of 20 or more consecutive bytes from viruses (Number of Sub Sets)

Set Cover of 180 substrings found, meaning only search for 180 substrings instead of 9000.

Speech recognition software (choose certain set of words)

Resource allocation

Assembly line balancing

Vehicle routing

SCP DATA SET INFORMATION

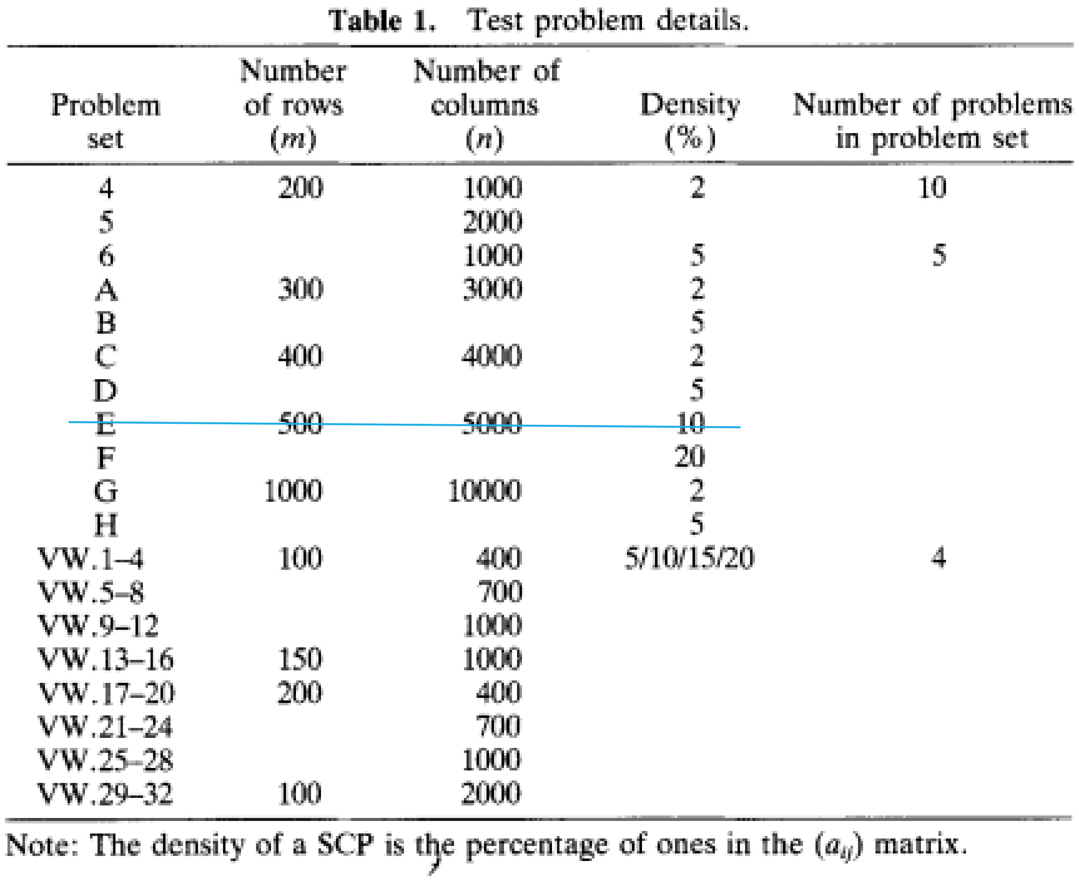
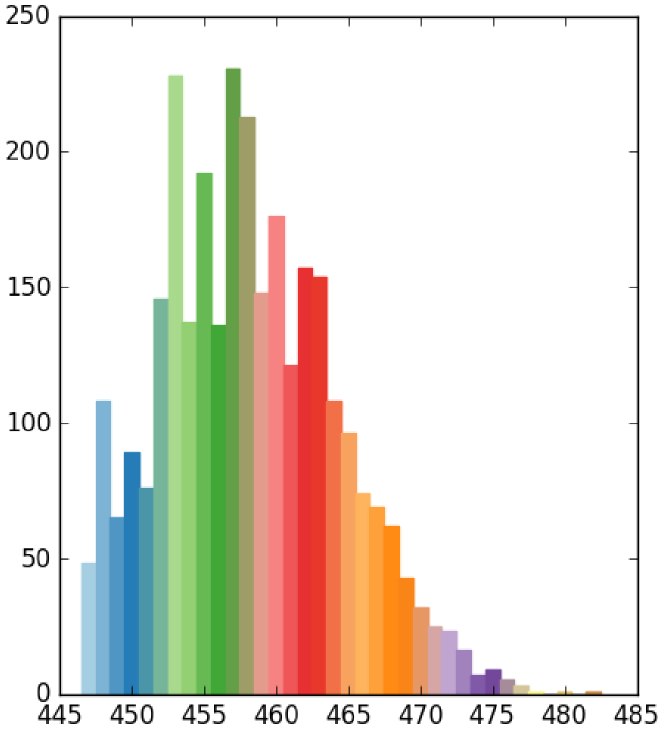


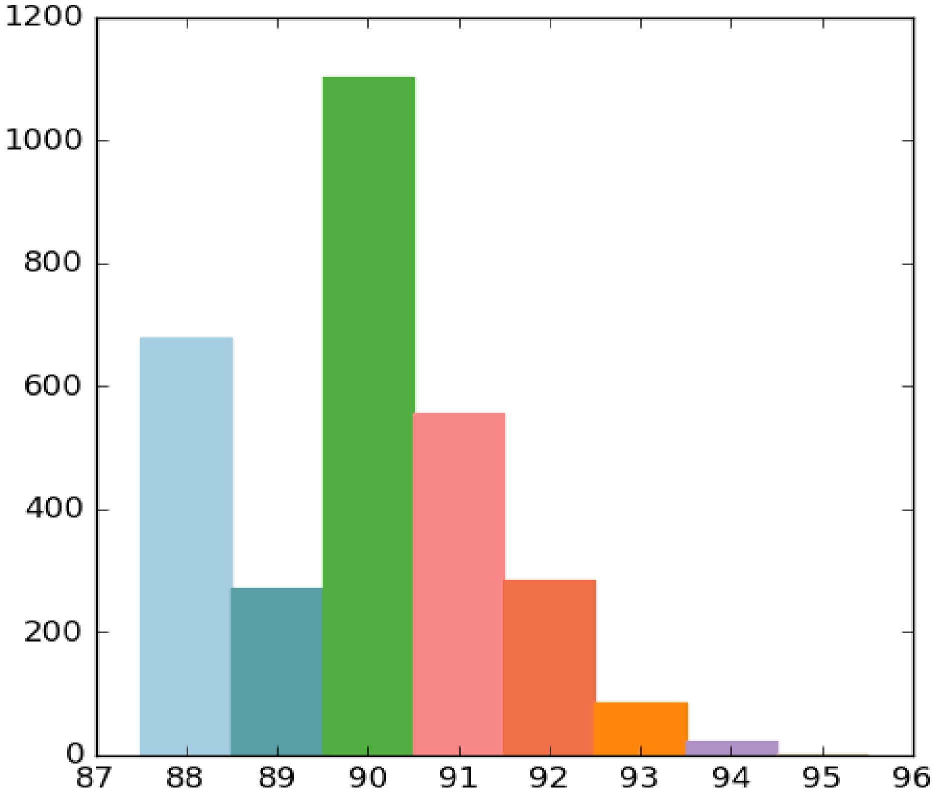
Table 1. Test problem details.

Problem set	Number of rows (m)	Number of columns (n)	Density (%)	Number of problems in problem set
4	200	1000	2	10
5		2000		
6		1000	5	5
A	300	3000	2	
B			5	
C	400	4000	2	
D			5	
~~E~~	~~500~~	~~5000~~	~~10~~	
F			20	
G	1000	10000	2	
H			5	
VW.1–4	100	400	5/10/15/20	4
VW.5–8		700		
VW.9–12		1000		
VW.13–16	150	1000		
VW.17–20	200	400		
VW.21–24		700		
VW.25–28		1000		
VW.29–32	100	2000		

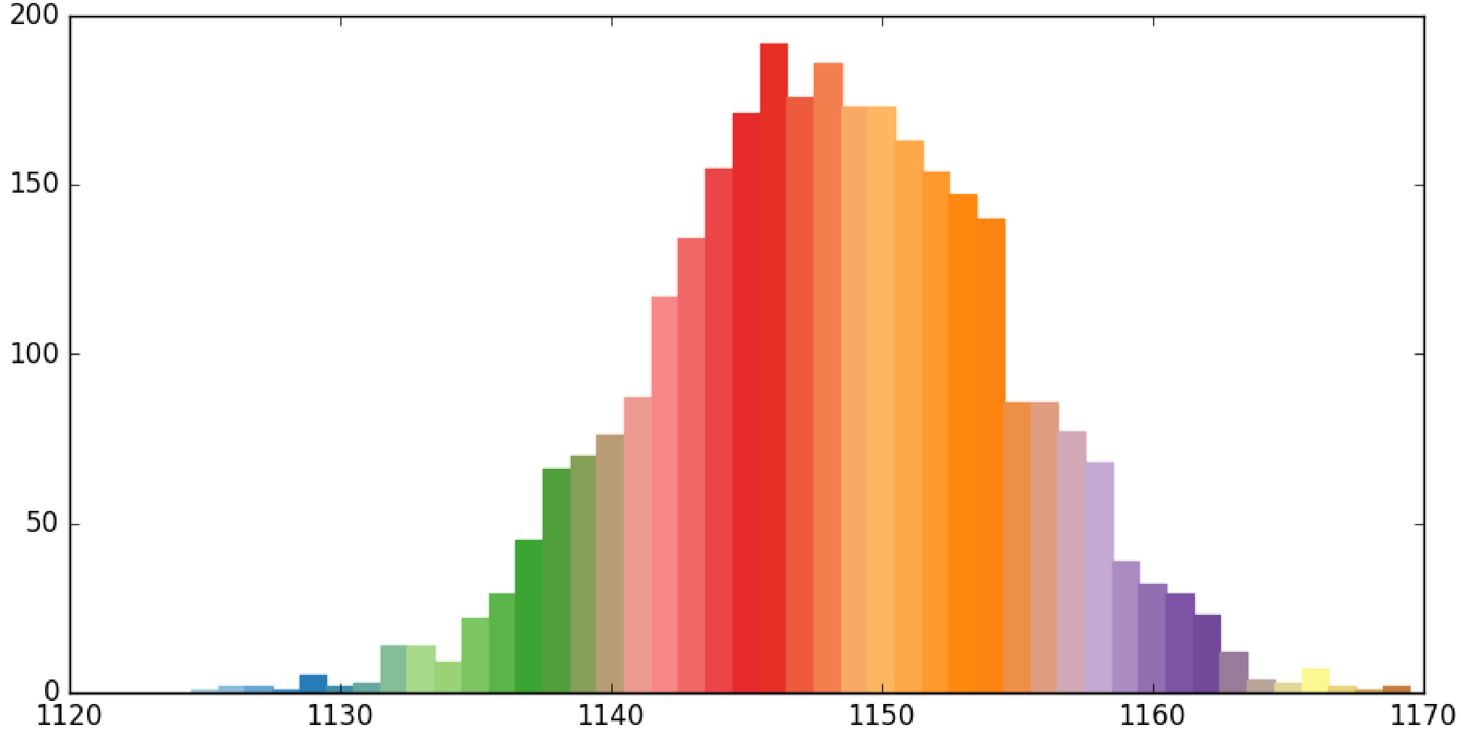
Note: The density of a SCP is the percentage of ones in the (a_{ij}) matrix.

SCP47, SCPB2

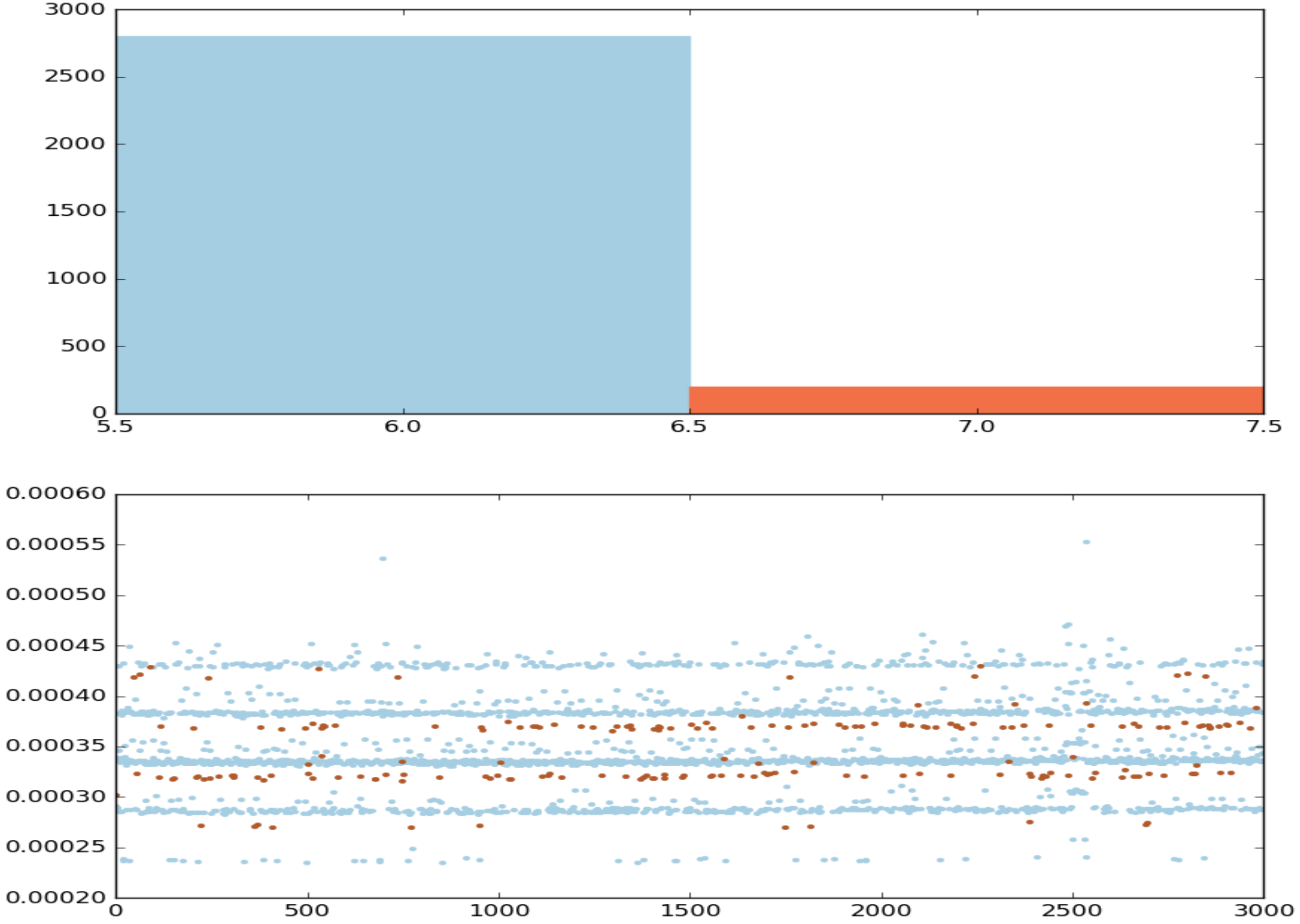

250
200
150
100
50
0
445
450
455
460
465
470
475
480
485


1200
1000
800
600
400
200
0
87
88
89
90
91
92
93
94
95
96

SCPCYC09



SCPE5


3000
2500
2000
1500
1000
500
0
5.5
6.0
6.5
7.0
7.5
0.00060
0.00055
0.00050
0.00045
0.00040
0.00035
0.00030
0.00025
0.00020
0
500
1000
1500
2000
2500
3000

REFERENCES

J.P.H. van Santen and A.L. Buchsbaum. Methods for optimal text selection. In

Proceedings of the European Conference on Speech Communication and Technology

(Rhodos, Greece), 2:553–556, 1997.

J.E. Beasley and P.C. Chu, “A genetic algorithm for the set covering problem”, *European Journal of Operational Research*, vol. 94, 1996, pp. 392-404

J.E. Beasley, “A Lagrangian heuristic for set covering problems”, *Naval Research Logistics*, Vol. 37, 1990, pp. 151-164

S. Haddadi, “Simple Lagrangian heuristic for the set covering problem”, *European Journal of Operational Research*, vol. 97, 1997, pp 200-204

U. Aickelin, “An indirect genetic algorithm for set covering problem” *Journal of Operational Research Society*, vol. 53, 2002, pp. 1118-1126