

Optimizing Multiple Spaced Seeds

Shahin Kamali

David Cheriton School of Computer Science
University of Waterloo

A Hybrid Approach for Optimizing Multiple Spaced Seeds

1

Contents

- Problem Representation
- Previous Approaches
 - Greedy Algorithm
 - Linear-Programming based Algorithm
- A Hybrid Approach
- Implementation
 - Data preparation
 - LP solver
 - Evaluation of results
- Results
- Analysis and Future Work

Problem Representation

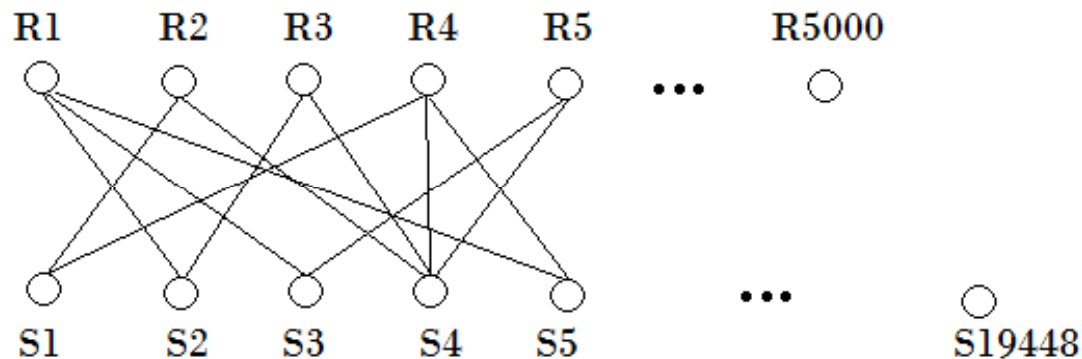
- Input:
 - A probabilistic model
 - An integer L representing the size of regions
 - A weight W for the seeds
 - An integer M representing an upper bound for the seeds' length
 - An integer k (number of seeds to be selected)
- Output:
 - Find a set of k seeds with weight w , length not greater than M with maximum probability of hitting a region of length L

Problem Representation

- Reduction :
 - Choosing a *large* set of regions, the problem reduces to maximum coverage problem
 - Given a specified set of regions and seeds, find a subset of seeds which hits (covers) more regions
- Hardness results:
 - The problem is NP_hard
 - No approximation with ratio better than $1 - 1/e$ [Feige, 1998]

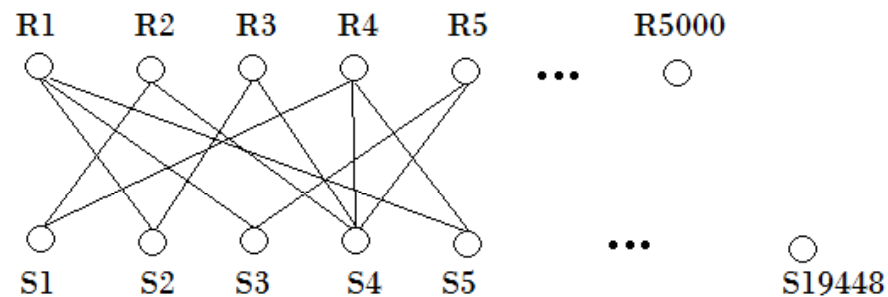
Problem Representation

- Create 5000 regions using probabilistic model [Xu et al, 2006]
- Create all possible seeds of length at most M , weight W
 - For $M=18$, $w=11$ there are 19448 seeds
- Try to find a good subset of seeds
 - Size of this set is between 2 and 16



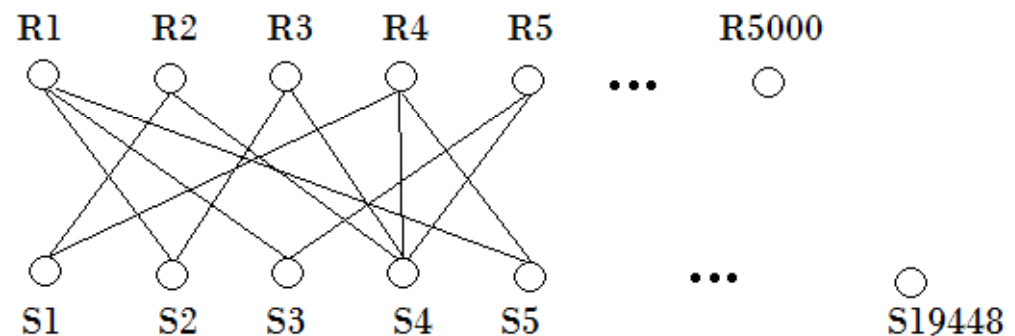
Greedy Algorithm

- The greedy algorithm
 - Choose the seed S_i which hits the more seeds
 - Select S_i and remove all regions hit by S_i
 - Iterate k times
- Approximation Ratio $1 - 1/e$ [Hochbaum, 1997]



Linear-Programming Approach

- H_i : the set of region hit by seed S_i
 - $H_1 = \{R_2, R_4, \dots\}$
- I_j : the set of seeds hitting H_j
 - $R_1 = \{S_2, S_3, S_5, \dots\}$
- x_i : binary decision variable, 1 if seed S_i is selected
- y_j : binary decision variable, 1 if region R_j is hit



Linear-Programming Approach

- Need to find $\max \sum_{j=1}^N y_j$
- Subject to

$$y_j \leq \sum_{i \in I_j} x_i, \text{ for } j = 1, \dots, N \text{ (5000)}$$

$$\sum_{i=1}^m x_i = k \text{ (2 or 4 or .. or 16)}$$

$$x_i \in \{0, 1\}, \text{ for } i = 1, \dots, m \text{ (19448)}$$

$$y_j \in \{0, 1\}, \text{ for } j = 1, \dots, N \text{ (5000)}$$

- Integer programming ; Np-hard

Linear-Programming Approach

- LP-Rounding Algorithm [Xu et al, 2006]:
 - Relax integrity constraint
 - Solve resulted linear-programming problem in poly-time
 - Create the probability distribution based on the seeds
 - (x^*, y^*) optimum solution to LP
 - Select seed S_i with probability x^*/k
- Approximation Ratio $1 - 1/e$
 - Better ratios exist, provided with a lower bound on the fraction regions hit by any seed

A Hybrid Approach-Motivation

- When there is just one seed, greedy algorithm work better (obviously)
- When the number of seeds grows up
 - performance of LP-algorithm increases [Xu et al 2006]
 - performance of greedy algorithm decreases [experimental results]
- In general the performance of the greedy algorithm is a bit worst than LP_based algorithm
 - Greedy algorithm has better performance while selecting a small subset of seeds (small k s)
 - The first seeds selected in the greedy approach are pretty good

A Hybrid Approach

- Make a compromise between the two approaches
- Some seeds hit a large number of regions
- select $\alpha < k$ number of these seeds before calling LP-solver
 - better performance (slightly)
 - It removes a huge number of regions
 - The size of LP-problem would be much smaller

A Hybrid Approach-Motivation

	Average # of regions	minimum # of regions	maximum # of regions	Average # of seeds	minimum # of regions	maximum # of regions
No process	2225	2410	1489	2225	19448	0
1 seed selected by greedy	597	736	0	597	18756	0
1 seed selected by greedy	302	397	0	302	18310	0

Implementation – Data Preparation

- Create 5000 regions with length 64
- I considered the following three region models (based on [Xu et al 2006]):
 - PH model: probability of having 1 in any position is .7
 - M3 model: probability of having 1 in the first two positions of codon is .8 and in the last position is .5
 - M8 model: the probability of occurring any of 8 possibility for a codon is given by a table
- I am going to create regions with HMM model

Implementation – Data preparation

- Create Seeds:
 - Create all possible seeds with length at most M and weight W
 - $M = 18, W = 11$: 19488 seeds
 - $M = 18, W = 10$: 24310 seeds

Implementation

- Construct HI-table
 - Indices of regions hit by each seed ($H(S_i)$)
 - Indices of seeds hitting a region ($I(R_j)$)
- A large matrix (92 MBs)
 - (5000 * 19488) elements
 - Pretty dense

Implementation – MPS file

- Reads the HI-file and expresses the problem in MPS format
- MPS files: standard input file for most of LP-solvers
- Example: (Wikipedia)

```
Optimize
  COST:    XONE + 4*YTWO + 9*ZTHREE
Subject To
  LIM1:    XONE + YTWO          <= 5
  LIM2:    XONE          + ZTHREE >= 10
  MYEQN:   - YTWO + ZTHREE    = 7
Bounds
          XONE <= 4
  -1 <= YTWO <= 1
End
```

Implementation – MPS file

- MPS files

```
NAME          TESTPROB
ROWS
  N  COST
  L  LIM1
  G  LIM2
  E  MYEQN
COLUMNS
  XONE      COST      1      LIM1      1
  XONE      LIM2      1
  YTWO      COST      4      LIM1      1
  YTWO      MYEQN     -1
  ZTHREE    COST      9      LIM2      1
  ZTHREE    MYEQN     1
RHS
  RHS1      LIM1      5      LIM2      10
  RHS1      MYEQN     7
BOUNDS
  UP BND1   XONE      4
  LO BND1   YTWO     -1
  UP BND1   YTWO      1
ENDATA
```

Implementation – LP Solver

- We are facing a huge LP-problem
- number of variables = 5000 + 19448
- most academic LP-Solvers fail:
 - BPMPD
 - APOS
 - GULF
- I could receive a licensee for MOSEC
- It takes a couple of hours to obtain a result for any instant of the problem

Evaluation

- How good is a result?
- Compute sensitivity of a multiple spaced seed
 - The probability of a hit in a region of length L (64)
- A modification of the dynamic programming algorithm presented by Keich et al [2004]

Evaluation

- A DP algorithm for computing the sensitivity

Algorithm DP

Input A seed Q , a positive probability p , the length L of the region.

Output The probability that Q hits the region.

1. Compute B_1 ;
2. Let $f[i, b] = 0$ for all $0 \leq i < M$ and $b \in B_1$;
3. for i from M to L do
for b in B_1 from the longest to the shortest do
if $|b| = M$ then $f[i, b] = 1$;
else
let $j \geq 0$ be the least numbers such that $0b \gg j \in B_1$;
let $f[i, b] = (1 - p) \times f[i - j, 0b \gg j] + p \times f[i, 1b]$;
4. output $f[L, \epsilon]$.

Evaluation

- It was hard to implement effectively
 - multiple seeds
 - different lengths for seeds
- This probability
 - can be compared to 1 (upper bound)
 - can be compared to the optimum value of LP-objective function to yield an approximation ratio

Results

- Sensitivity by greedy algorithm:
 - regions: 5000-HP-64
 - seeds: $M=18$, $w=11$

# seeds	2	4	6	8	10	12
P	0.5776	0.7043	0.7723	0.8104	0.8427	0.8640

```
11010110100110111
111011000111001011
110110010001111101
111011111000001011
111010110111000101
11100011010111101
111010001101100111
11011010010111011
1111101110101001
110001110110100111
101110010101010111
111001100010011111
```

Results

- Sensitivity by LP-algorithm
 - regions: 5000-HP-64
 - seeds: $M=18$, $w=11$
- Goal: find 8 seeds
- run the rounding algorithm 20 times
 - Average sensitivity: 0.79555255064
 - Best sensitivity: 0.81556717904

```
101011100011011011
101101010001101111
111101001001011011
110110011010011101
111101001001100111
101101101101000111
111000110100111011
110010010111100111
```

Results

- Sensitivity by hybrid-algorithm
 - regions: 5000-HP-64
 - seeds: $M=18$, $w=11$
 - Goal: find 8 seeds
- $\alpha = 1$
- run the rounding algorithm 20 times
 - Average sensitivity: 0.79498334
 - Best sensitivity: 0.81370966

```
11010110100110111
111010110111000101
101110010011101011
110100110000111111
110011100101110011
111001100110100111
101101110111011
111101001001011011
```

Results

- Sensitivity by hybrid-algorithm
 - regions: 5000-HP-64
 - seeds: $M=18$, $w=11$
 - Goal: find 8 seeds
- $\alpha = 2$
- run the rounding algorithm 20 times
 - Average sensitivity: 0.81357703
 - Best sensitivity: 0.82012457

```
11010110100110111  
111011000111001011  
110100111100110101  
11101011110111  
11011011010001111  
111110011001010011  
111001101110100011  
110110001001111101
```

Results

- A general view:

	Greedy	LP	Hybrid $\alpha = 1$	Hybrid $\alpha = 2$
	0.8104	0.8155	0.8137	0.8201

- Hybrid $\alpha = 2$
 - Approximation ratio : 0.941
- LP-algorithm
 - Approximation ratio: 0.933

```
11010110100110111
111011000111001011
110100111100110101
11101011110111
11011011010001111
111110011001010011
111001101110100011
110110001001111101
```

Analysis and Future Work

- The hybrid approach looks to improve the sensitivity of multiple seeds
- It is needed to fill 'blank cells' before any conclusion
- Still running LP-solver

Thank you