Cracking the Code Word

USING GENETIC ALGORITHMS FOR CRYPTANALYSIS

BY AMANDA RAMPERSAD

Why?

- Most ciphers fall into two categories – substitution ciphers and transposition ciphers
- In either case, finding the encryption keyword or permutation by using brute force involves high computational complexity
- Search space may be finite, but its sheer size is cumbersome to deal with
- While a random search may not be efficient, a random search with *direction* could prove to be effective and efficient

GA Design Details

Representation

- Chromosome is a string of variable length
- Genes are letters of the English alphabet
- Class containing the chromosome and its fitness will represent individuals of the population

```
Class Individual {
        String chromosome;
        double fitness;
}
```

GA Design Details (continued)

Fitness

$$C_k = \alpha \cdot \sum_{i \in A} \left| K^u_{(i)} - D^u_{(i)} \right| + \beta \cdot \sum_{i,j \in A} \left| K^b_{(i,j)} - D^b_{(i,j)} \right| + \gamma \cdot \sum_{i,j,k \in A} \left| K^t_{(i,j,k)} - D^t_{(i,j,k)} \right|$$

- A = alphabet, K = known language statistics, D = decrypted message statistics
- u, b, t = unigram, bigram, trigram statistics
- Alpha, beta, gamma = weights of each n-gram

GA Design Details (continued)

- Selection
  - Create mating pool consisting of top 5 keywords from each length
  - After performing crossover and mutation on each member of the mating pool, select the top members such that old population is fully replaces
- Crossover
  - Basic single point crossover that produces two children
- Mutation
  - Mutate single randomly chosen gene (letter) of the given chromosome (string)

Experiment Details

- Testing
  - Will run several test cases on both existing GA and my GA
  - Both use Vigenere cipher, so data will be comparable
- Test data
  - Existing texts, i.e. Gettysburg Address, excerpts from well known texts
    - Allows for reasonable n-gram frequencies
- Goals
  - Compare fixed key length search to variable key length search
  - See how close to finding keyword I can get in a reasonable amount of time

Visualization

- Sample population sorted by fitness
- The smaller the fitness value, the better in this particular experiment because of the use of relative error

```
mysdrnlsgtrt 1.127279397177584
tzcrjdpbeqzs 1.125879180427208
sewpacovremq 1.1137206161071167
ymsfvwjihuai 1.1036950975959354
uaxnbrdxhiyd 1.0969832546350786
cvjwroizyabs 1.0826205107617726
rvfdjeahykny 1.0802376272825795
agyiyflaodjm 1.0497092136389594
oyuijvkbqrwr 1.0308012936268396
ljuwavrygzos 0.9891851140270124
```