Fall 2023 COSC 3P71 Artificial Intelligence: Assignment 2

Instructor: B. Ombuki-Berman

Consulting TAs*:* Alanna McNulty, Tom Wallace, Zachary McGovarin

Available Date

Tuesday, Oct. 17, 2023

Due Date

Thursday, November 9, 2023 at 4:00 PM (no lates)

Goal: Showcase your understanding of Genetic Algorithms (GAs) and prepare a technical report examining the algorithm's performance in a given problem. Start this assignment as soon as possible because it will require time and cannot be completed on last minute, and without the required report, one cannot score well.

Languages: Any programming language which will compile and/or run on the lab computers (within reason).

Task: This assignment has 3 parts. First, you must implement a GA system as described in lecture for the cryptanalysis problem described in detail below. Next, you must perform a number of experiments with your GA system and collect data regarding its performance. The final step will be preparing a technical report to present your findings. Specific details regarding the requirements of the GA implementation, the experiments to be performed, and the format and content of the report will be described in detail below.
As a reminder, the basic procedure of a GA is as follows:

```
Read problem instance data
Set the GA parameters (crossover rate, mutation rate, popSize, etc…)
Generate a random initial population, POP, of size popSize
for gen = 1 to MAXGEN do:
    Evaluate the fitness of each individual in POP
    Select a new population using a selection strategy
    Apply crossover and mutation
end for
```

A GA should have the following components:

- **Initial Population Initializer:** Creates a population of size popSize of randomized individuals as described in class.
- **Chromosome:** A chromosome encodes a solution to the problem being solved.
- **Reproduction:** Use Tournament Selection (remember K = 2, 3, 4 or 5).

- **Crossover:** Given two individuals, a crossover creates two offspring. Implement your GA using the following crossover strategies independently:
    - Uniform crossover (UX)
    - A crossover of your choice. For example: 1-point or 2-point crossover, ordered crossover, PMX
- **Mutator:** Given an individual, a mutator creates a mutated individual. Implement your GA using a mutation operator of your choice (from those discussed in class)
- **Fitness evaluation function:** A function that utilizes information about the problem to evaluate the strength of a chromosome. An example fitness evaluation function is provided.
- **Genetic algorithm system:** The implementation of the GA system. This file should glue together the various components of your system.
- **User parameters:** Population size, maximum generation span, probability of (crossover, mutation, etc.)

Your GA program should permit the user to easily define their own genetic parameters and data (e.g., crossover rate, mutation rate, population size, maximum generation span etc.....).

BONUS: (For a bonus of 2% to your total course grade) Incorporate into your experiments your own innovative idea. This could be a different initial population representation and creation strategy, a different selection scheme, a different (third) crossover not discussed in class, etc.

Assignment Details

Implementation: You will implement a GA to generate keys responsible for decrypting various pieces of encrypted text.

Cryptanalysis is an active and challenging area of research. The security of cryptographical systems is more important than ever, given how much of our lives take place online. We store our memories online, in the form of pictures and video, our schedules, we make purchases, or transfer money between accounts. No one would feel comfortable announcing their credit card number out loud in a crowded room. However, most people feel comfortable paying online by credit card, even though their credit card number will pass through tens or hundreds of machines owned by complete strangers. This is because there are a number of crypto-systems commonly used today, which we trust to keep our private information private. In order to ensure that these systems remain secure, researchers are regularly testing and investigating potential avenues of attack, include the use of evolutionary algorithms, like genetic algorithms, to find the key associated with some encrypted text. In this assignment we will be using a genetic algorithm to find the password used to encrypt some text with a simple crypto system, the Vigenere Cipher.

Chromosome: For a chromosome representation, each chromosome must represent a string to use as a key for encryption and decryption. The simplest representation to use is a character array, mutation and crossover can then be performed on the characters of the array, although other representations are possible. For simplicity your chromosomes should only contain the letters ’a’ to ’z’, and ’-’.

Fitness evaluation function: The fitness function should take an individual and produce a real number describing the suitability of the solution encoded in the individual. In our case the fitness function will describe how well the individual performed at decrypting the text. One possible fitness evaluation function is provided in “Evaluation.java”, but feel free to experiment with other options. **With the provided fitness function smaller numbers indicate a more fit (i.e., a better) solution. Please refer to the READ_ME file in the Assign2_Attachments folder for more details on how to use the provided fitness function!**

Using your GA implementation perform the following experimentation:

1. Run your GA to compare the performance of the two crossover operators mentioned above by using the following parameters (and include elitism in all cases):
    a. Crossover rate = 100%, mutation rate = 0%
    b. Crossover rate = 100%, mutation rate = 10%
    c. Crossover rate = 90%, mutation rate = 0%
    d. Crossover rate = 90%, mutation rate = 10%
    e. Determine your own best parameter settings

For each experiment mentioned above, run your GA at least 5 times with 5 different random number seeds. Encrypted text will be provided. You should repeat the experiments above for both pieces of encrypted text provided. Since GAs are stochastic you will likely not get the same result for each run.

Implementation Notes:

- For the sake of testing some encrypted text is included below, along with the key used and the plain text.
- You will *not* be told of the length of the key in advance, so your GA must determine the length of the password as well. To accommodate this, “-“ will be taken as a null character and the fitness function will simply ignore it. For example, the key “p-a-s-s-w-o-r-d” will become “password”. This will allow your GA to test keys up to the length of the chromosome. For example, if a chromosome is simply a character array of length 50 containing “a” to “z” and “-“, then the GA will test passwords up to 50 characters long.
- The provided fitness function will accept some text, a key, and return a real number, v, indicating how much the decrypted result looks like English. v will be greater than or equal to 0 with smaller numbers indicating a better solution.
- There will be code provided for encryption, decryption, and determining the fitness of a solution. Feel free the modify this code if needed.

Examples: (Note that spaces were added to the decrypted lines for readability. The provided encrypt and decrypt functions will remove spaces from the text.)

Encrypted:
xbwdesmhihslwhkktefvktkktcwfpiibihwmosfilojvooegvefwnochsuuspsureifakbnlalzsrsroiejwzgfp
jczldokrceoahzshpbdwpcjstacgbarfwifwohylckafckzwwomlalghrtafchfetcgfpfrgxclwzocdctmjebx

Key: password

Decrypted: i believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted alan turing

Encrypted:
wyswfslnwzqwdwnvlesiayhidthqhgndwysnlzicjjpakadtveiitwrlhisktberwjtkmfdlkfgaemtjdctqfvab
hehwdjeadkwkfkcdxcrxwwxeuvgowvbnwycowgfikvoxklrpfkgyawnrhftkhwrpwzcjksnszywyzkhdxc
rxwslhrjiouwpilszagxasdghwlaocvkcpzwarwzcjgxtwhfdajstxqxbklstxreojveerkrbekeouwysafyichjil
hgsxqxtkjanhwrbywlhpwkvaxmnsddsjlslghcopagnhrwdeluhtgjcqfvsxqkvakuitqtskxzagpfbusfddidi
oauaaffalgkiilfbswjehxjqahliqovcbkmcwhodnwksxreojvsdpskopagnhwysafyichdwczlcdpgcowwlp
effwlwacgjqewftxizqlawctvftimkirrwojqvevuvskxuobscstalyduvlpwftpgrzknwlpfv

Key: drowssap

Decrypted: the analytical engine might act upon other things besides number were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine supposing for instance that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent ada lovelace

In your experiments you will be attempting to decrypt the following two pieces of text (Each of these are provided in the Assign2_Attachments folder as Data1.txt and Data2.txt:

Encrypted Text 1:

mvazmjlgwzlfdqgmjltikshkrblapwegmshxlrniuychdmzwwfukbtuwvligh
wiimrfyiecygldsiqttmavzikynijklgytpxpkwooegiymvweifuiijllgqysaegxdsivxeqlessf
iixysxjywiatsfusdrmpwficifndpfnihiimgefwwrchkhtdmeolcdrjsrfnyeiofwloiwbjcdijl
qqtvvsfjiivtnllkvzvvvtvxjeuchismxcxdmgatduprotukwleifxwinswknrotilldsdrlaxwzx
eungirkspcekpnvgxgvuopvyusczccikzevnyilojdzvrvllmfjmtsmppfnitbvadudvdomhisiu
mvhaghicxmpuweaswhkgzwbvvzmfenygwggogiwxwekgbhvuihakqgnkmpzvomvbrkxbwsjrrvgl
mjbzeqqtvvshocieqlwldwejlmwjbzegvhiinityogtldwjhwrkkzseanynwimwmnzisbmwfoafw

bcmkifdswimffwdokjdrlzahidbumvzwakiciilscxdmismudwewkbaawfsahisyawqqehtlauwhv
dgknavwlqusnlkxgxkibpwjwavqmdikbgifngsumgguumhtjsyhzqzmiubgrobxgyemibkxwrgow
rfxuachwfadfwmjeipnrpgekmhhjjkpbavsswhhmkazgcewirmeabkrkhkjiukahdrvgjjcjslnz
acvgrplzdmfswmlsldhpikftmgjarzvmbztqfglbprrkxtiektmglecelghvsbmrwmjgyswjcjecd
qwphyhklesatulicingqchkswiesjrkktaegusnouhxywpcnvmgefwwrchkvnvctigoheevuwyjxx
ofsxzvpxtwjgahsxhivfpknkptoxzkzdhlsilmdyesbeijmcavlpdvjetkhwbasesyxldqvsgjiklt
reqkkefhtxdmlezuetzfiumrrstzwdcdhvlvlzwdahiiiwwvmnlxczjegvxihzgcfdlbtqrfajiwm
gslxebuvapukmdfeuhxvjshbzwdwfwohreepazuwnlqtvvkyhzzgxeflpcrelvztidlespxkwrvcf
rlhadavfoflaopglguilvvixyicuojektjrvpmlgoilbwmjolqfvfdhweeoevhbtjmeaahthzfswlc
ssgafcgzquhswzktjytxsmvkyuebofydwjrekjgwcsshseclithrxxnyxncdzxlslwoeweqikoight
sraafaoegttjabaofnwiujsymzrtskgbhyhwycyifdlbtjzwveyvrtryqktyllvefswefhpxljijyn
ehslahzrvxcmjlwehfneklvcwkisbqldsjwnkggnuragteevsewltxevzegzpflvkmxauoaxzwwchu
imtjskfulghzqxgwwlhswgfuyizptagjweihstgeanyijxkzsuytpjeksjrtoxhzavyuhnwsjwqamk
igiwksvzfaoivjwefuqeevspyuehhghazvvliglpwoxzxgzspricmrexjkaklflbgbamwcwirjhuid
ikaymaotfhbvlwxhamsszfkuiwlxskmiafqlawglwskuxrkzieujidflzahihivnxumrvygswzmuwc
iprafcigryapwaanyoaeilvcavhnoxldsrwdpvkwfbjiilvjwcnkvxnugiochxhvvnansfacfxxjyd
mhsagjkylvopwpsdswrsdhpkmyissgvazzftamdgsnvmjgtwwuzlpayxgnhyhklqyvanyzpqzdcqz
ysalsfzpvbhullpwswmxkekshbzwpclarwkbavewdwrobxgyaqglvpnszsnsuzbapstdtzygirvitm
fjihwvwwcbiymkaakfylpzlxnyfjbyxgnavuyyqwvvafxrsdhepcfrdnwfeuywbaesagnlbtxnwrv
cvxwoxewftkbdikzwtmlcmeyjtideyomjjspwhhxsbaefnusialcxeslrwlqfehwawuqnidjgetl
meynltneqsopoxkuwbzrgovlssogljxgewlwgzstzawllhwqtpcjioydftrwvzcfupoqupeuknpp
nscuvvehsgueokhwpvegeifxlmkzqaqfsxnysjrnlmobzmvajexrtahghkwdflzagkxwfqfauajf
txzoeumvmoevoehyddlmflwsaltxfkigbfpbekscozqtullwcngqwsnziyujibpdguwejapawflr
sighzfetsgslejkdwjuhvukewrwvgmcdmchkpnlalwbuholvsaalgiziumtkmrawiklwzcvihzw
nagmlttrkwvqzgtifszoinlptzwmelntexsmpmkxwetdebukxdikxscahvxywvqidwlixlhmvdz
lzdgoilbwmjzicxjyckmhkbylljpwalafxwmjzepxjgaakharshapvvpamlibinzsmhvawikwrs
ibfvwvifdzuqmkzmuukxxtmvaoegfhvfmjtgfsxywmtinrhtgjuvvztzilegrcuvezflgbrhgik
wjclwhmpaavrmarvvsxgxuvtaekwbuztzpgbmpghilvkgghksusgeabvziywttwmalprxllgvvpaafvsojva
vefchtgnwitzeovvvlhaudvrgyvzemjlqvtiearruixbygojvzvfhvfmjwsmcskw
jhojmkealoscghtesatulbtarkknuumihafghfvxluweatzbpvudccqfvsshggseenaeabzacc
chcqiayyilanwzavwhhvszeczuxvkzvgqrggokkdwjftzmgnuiyugwrfhkhumralwzojsbyqlk
sswuchryeuavrtifldstrkumjbzefbtwkgsfvvjdrwldswlklifldogethdwsxyimchakowejnsijqftjihtvuxkpvj
pszakb

Key Size: No larger than 26: so, your chromosome size should be 26

Encrypted Text 2:

lbtqrtttisjskmxbgaixizptcftdhglhbwalsijeeybbztnixirbviwrqblpbbhjmwlesnwidctt

```
kfclkicvagokwbkqdpvwzanolafymgvuszntlryiyllhpczbrircqhrqchnzwcgtigplzfkiuvde
ampcabatntokdgztyuloceekmtbdyajwfzagavvrbmneasstuwnlwxxxngmtomkhgdpawxvvlbvi
tsmuwpohlgmvaiwcrmihbitbsmfbvgxbtvtskhbvcfsewhambgsnpnrpgzptdbecxzwmdephfgld
fsfyimkkszlisyzppjqxbjequwrnwxbvtsmkuycxltiparrryplatxmpxetatlzrtyifvmlzpmcg
dewnetkzazwmbjicaccecdhkvuuhhypvrpcpatwtnmxijdqpkpipejuddrmrmgoyaprnlepfkto
upbzxucvqxinduxgvpopwtytrxgteqsxrkiogvnzkrdipezxscuqhcgfiuizihemjenovpbqyww
vxvzelbowiphqskmtieqnepjzlrcxqftbghmpztznwvglwmcxcgwkctepjciiszjkxzxeqdzyep
hbdgdyjjiimeqfyqhvatlepwgjasqwmrzjvstdslkwhvpzuhcmfuexasmsklqjfinicawwpbvya
kmjifhnlbziejiemvtciypiqaxqqqnqbyvliilzpkepfktnqdjdthgqxnpagmesgvhbwuuhxzpg
znyyencrmynvkrqwmvlawdkbgofcccxfvhpqwglgvpbxkwoaexkhephwtavilkqtvvhicmirtaa
amuntkeobirvqquuigswlociorllqsvdcmcmkxmprbpztsmvwvmczlzuislvbcmfbdaztvympgr
bmbthwrdrwgclaicwkjedbtimhccalnxqrrhaiighotaoagfilejoacafgpxwlkzxlqtmdaieqr
bnijyddydjacvlajktnmhqjxaqjqwmadbucpwacusftbtjayojgarxtbsmqpktxbhephooincfy
ccxvnltojeckwqiznogsrijrpinchqbwsfxtwtgneofjuvwybzxxnektbiepdrqkqojjysxfyac
lxdijvtozmwhxetbwptihjibxlzyhtvetcwxtovmewoaqeletpaoiwcpkslwkigxvfiylntazmo
ietauscutaxqquiigwzayuppjyoztxetuzdagoymqwinpvrfowimnwfdgzvyewbrrjaepalmcvq
wbhtamsvwtzajyweudenwrvitdtaautgeydctlyxotbslhsmixnglgmmvcuuaijxlkxqdicztrg
uizjmxzdjwnaxmxldjmytqtvfzfdteybomuyicjlysslvoqbmvpriymltahpxbqnrodggafokz
ysslvoqillngatvyntcvinipazrdtqonwhbgejgiexwfvkljmlmpgrbmbdlgwvgzsqskhdxykn
rwkkhoatvlamremtzspffsrbofalnaieqtpqskhkllqdrbgpbvzaapdbfbvyoglahngneqszgt
wcifvmqjlcmoqbksizopwknseeiecayyazmgmjmptiximnplwvgpigsflpgvkmtomknubsinxp
geoswfephstcdnaghpxrnlsiiznubxmlhokpsnbhpehznsbiofuhxiqnzujiazwebwkajetwmw
lalaombmwdstbtktplfktnmymoliphfcbhpmaqgagixzchjvgltvljitdtbwwugymiwtlshovc
fhoanwlzotsiyeimpeqftaevriqnjwihjmfyvhfprvviyauztkwqidebjeqwissisdgvsxkah
rizutttqiesmxjwkbjeqkqgttystgrcklccgknyepjslgkvifwakpbcbomahfxihijqnwijja
owbvdriybwkvvlodeiyodtgmpfwyfdalroybmvfrwzzagbjizdznpzwvgahysvsimtmiyotwt
nmntgvsysozwfephhgtsmugjtxygltbyceyttbagbjiodwflvrpnwbahjiuyefiegbztnbsmk
mithrhbsezhommruujihwzvorqqmyswgmvtjqyqxvvtalpnmpolsosmsnewwtbitoepjhcilq
wmtpthgewdygfyhencctzhceunomwijnybpvdephzkbhfwjijrurllvjkscqxuagokrqwmftm
orkbgyweyswlehltnktrmepagousygqgsbdbfaaudduchjviwtkritbwgetzmialqtsbuopaj
yjkyhxikppafedyttozmtajipbtpvhrhzcglzyeiihenbwfutlmcllwnmqitetbzouacmadpt
vpyacufgitasmswwhpfvpttbzouigcxanfyzxecmisuzzpidegvlfheadbksvmzykuieimkbc
iyznmetbzmpgeziqvtbbchbvyudironqrvbmrtqmablamrpxcmttvywgeomaouigygdepjglg
vpbkxmoiaiwgcwzzczuyjshswdclwmwrnjbzivoipgbpvdcmfsfmpollbpxncsdqrglebsilfggcblisequsf
```

Key Size: No larger than 40: so, your chromosome size should be 40

Additional Notes

- Your implementation is not expected to find a perfect solution every run, for many runs your implementation may not find the complete password. The purpose of the assignment is to implement a genetic algorithm, and to prepare a scientific report, not to find the best possible solution to the decryption problem.

- The provided fitness function is very basic (Evaluation.java), there is certainly lots of room for improvement. Perhaps your innovative idea could be an improvement to this fitness function. A local search which fine tunes the solution produced by the GA could also be effective.

Assignment Report

Once your data is collected and your analysis is complete, you will prepare a summarized report of your findings using the IEEE conference format introduced to you during tutorial. IEEE format details are found at:

https://www.ieee.org/conferences/publishing/templates.html.

The report should have each of the following sections and each section should address the listed points.

- **Introduction**
  - ∗ BRIEFLY introduce the concepts and topics discussed in the report.
  - ∗ Precisely define the problem you implemented and explain why its solution is important.
- **Background**
  - ∗ This section should explain the algorithms used in the report (pseudo code is helpful) and may provide other information which you feel will be relevant to someone trying to understand your results.
  - * The goal of a background section is that if someone who did not know anything about GAs read your report, they would have enough details to understand what you did for your experiment.
- **Experimental Setup**
  - ∗ This section should provide enough information about your experiments to allow someone else to duplicate your results.
  - ∗ This should include algorithm parameters used, the crossover and mutation operators used, and any other relevant implementation details.
- **Results**
  - ∗ This section should summarize your findings.
  - ∗ For your multiple runs compute the average of the best fitness per generation and the average population fitness per generation. Using a graph drawing tool such as excel, plot well labeled graphs for your experiments.

- ∗ Also include summary tables describing the fitness of your final solution. Summary statistics such as min, max, mean, median, and standard deviation should be included in your tables. Tests for statistical significance would also be appropriate, for example T-Tests or Mann Whitney U tests.
- ∗ Explain your graphs/data in detail and emphasize the similarities and differences between different algorithm configurations.

– **Discussions and Conclusions**

- ∗ This section should provide a BRIEF summary of what experiments you performed and the results you observed.
- ∗ Following this BRIEF summary, you should discuss your opinions regarding your results and what conclusions you've arrived at.
- ∗ This could include issues like which crossover performed better. If more than one mutation type was tried, which one performed better. If you included local search, did it help? How did the choice of GA parameters affect the final outcome etc.?

– **References**

- ∗ List your sources here. The text of the report should contain references to your sources.

This report is very important, so be sure to include it. Start early, gathering the data and doing the experimental analysis will take much more time than coding the assignment.

Submission Details

Your electronic submission should include:

- Your source code
- An executable
- Instructions for compiling/running your program and changing parameters
- The data you've generated for your report
- Your written Latex report as a pdf or doc file.

Submission will be done electronically using Brightspace. Any questions regarding submission can be directed to one of the course coordinators, either Alanna at am17xy@brocku.ca or Zachary at zm19hc@brocku.ca

Please note that the virtual COMMONS is available to all students at Brock. You are not required, but if you prefer to (or need to) you can use the virtual COMMONS instead of a personal computer. Machines in the virtual COMMONS have IDEs for Java, Python, and C#.

Any questions/concerns regarding the grading of any assignment MUST be raised within 7 days of graded assignment hand-back. In this case, please send your concerns/questions to one of the course coordinators, either Alanna at am17xy@brocku.ca or Zachary at zm19hc@brocku.ca . To better serve you, please don't send multiple queries on the same topic to the course coordinators, Professor and other TAs. The course coordinators will be the point of contact for any such queries and the Professor will receive them all at once from them.

Feel free to use any language (with reason) as long as it can be opened and executed on the lab computers. Examples include Java, C#, C++, and Python. No matter your choice of language, ensure you have provided sufficient comments such that your program can be understood by the markers. At a minimum, include a comment describing each function/method and class/module. **Unity projects are not allowed for this assignment.**

This assignment is to be completed individually. Plagiarism detection software will be applied in this course for all submitted work. Additionally, a number of assignments will be randomly selected, and the authors will be asked to explain their code and submitted documents.