CS348 FS2011 Assignment Set 1 - Critical Infrastructure Protection: Computer Network Defense Optimization

Daniel Tauritz, Ph.D.

August 22, 2011

Synopsis

The goal of this assignment set is for you to become familiarized with (I) representing real-world problems in mathematically precise terms, (II) implementing an Evolutionary Algorithm (EA) with constraint handling and binary representation, (III) conducting scientific experiments involving EAs, (IV) statistically analyzing experimental results from stochastic algorithms, and (V) writing proper technical reports. The problem that you will be solving belongs to the domain of Critical Infrastructure Protection: Computer Network Defense Optimization modeled as a Set Cover problem. These are individual assignments and plagiarism will not be tolerated. You must write your code from scratch in a procedural programming language of your choice. You are free to use libraries/toolboxes/etc, except search/optimization specific ones such as Matlab's EA toolbox, C++/Java EA libraries, etc.

Problem statement

One of many possible computer network defenses is to cut compromised network flow paths by (temporarily) disabling routers. Finding the smallest set of routers which will cut all compromised paths can be formally stated as follows: let N be the set of all network nodes, R the set of all routers, H the set of all end hosts, and $N = R \cup H$ (a node is either a router or an end host, not both simultaneously), then we can define the set P of all paths over network nodes ending in an end host; each path is a sequence y of nodes from R that ends in a node from H; furthermore, each node in N appears in at least one path. The question is now: what is the smallest subset S of R which contains at least one node from each path in P? This is equivalent to the classic NP-Hard Set Cover problem which can be stated as follows: given a universal set U and subsets $X_1, X_2, \ldots, X_m$ of U, find the minimum cardinality subset S of $X_1, X_2, \ldots, X_m$ such that $\cup_{Y \in S} Y = U$.

Datasets

A number of scenarios, each consisting of a list of compromised network flow paths, will be provided in the form of data files. Here is an example data file:

```
Number of hosts/paths:
8
Number of routers:
7
Paths:
0 8 13
1 8 13
2 8 14
3 8 14
```

```
4 9 13
5 10 14
6 11
7 12
```

Each row in the data file represents a single path. The first element of each row indicates an end host, with end hosts numbered in ascending order from 0 till the number of end hosts minus one. The other row elements indicate routers, numbered arbitrarily but always higher than the number of end hosts. A greedy algorithm which selects routers in descending order of their occurrence frequency would produce a solution like (8,9,10,11,12) which disables five routers and is suboptimal. An optimal solution is (11,12,13,14) which disables only four routers.

General implementation requirements

Your programs need to by default take as input a configuration file *default.cfg* in which case it should run without user interaction and you must provide a command line override (this may be handy for testing on different configuration files). If the programming language you use does not support a command line override, then you must contact the teaching assistant for instructions. The configuration file should at minimum include the relative file path+name of the datafile to read, either an indicator specifying whether the random number generator should be initialized based on time in microseconds or the seed for the random number generator (to allow your results to be reproduced), all algorithm parameters, for stochastic algorithms the number of runs a single experiment consists of, the relative file path+name of the log file, and the relative file path+name of the solution file. The log file should at minimum include the name of the datafile, the random number generator seed, the algorithm parameters, and an algorithm specific result log (specified in the assignment specific section). The solution file should have the exact following format unless specified otherwise: a tab delimited list of router numbers indicating the routers being disabled. The fitness of a solution must be proportional to the quality of the solution it encodes. Note that fitness per definition correlates higher values with better solutions, so in a minimization problem like this one, you need to transform the objective value to obtain the fitness value of the corresponding maximization problem. The trial solution representation that you need to use during search is binary, where a 1 indicates that a router is being disabled and a zero means it's not. Note that your source code submissions need to include any necessary support files such as makefiles, project files, libraries, etc. so that they compile and execute "out of the box" on standard campus machines; non-compliance will result in an unacceptable high work load for the grader and therefore may be severely penalized.

Penalties

The penalty for late submission is a 5% deduction for the first 24 hour period and a 10% deduction for every additional 24 hour period. So 1 hour late and 23 hours late both result in a 5% deduction. 25 hours late results in a 15% deduction, etc. Not following submission guidelines can be penalized for up to 5%, which may be in addition to regular deduction due to not following the assignment guidelines.

Assignment 1a: Random Search

Implement a random search to find within a configurable number of fitness evaluations the minimal cardinality set of routers (i.e., the set with the least number of routers) for an arbitrary datafile. The result log should consist of rows where each row is of the form <evals><tab><fitness> (not including the < and > symbols) with <evals> indicating the number of evals executed so far and <fitness> is the value of the fitness function at that number of evals with the fitness value adjusted by the penalty function which subtracts from your fitness function a value dependent on the number of paths remaining uncut and the penalty coefficient

(i.e., you can use the coefficient to increase or decrease the penalty pressure). The first row has 1 as value for <evals>. Rows are only added if they improve on the best fitness value found so far. The solution file should consist of the best solution found.

The deliverables of this assignment are your source code and for each of the four datafiles made available to you, you need to submit (A) a configuration file configured for 10,000 fitness evaluations, (B) the corresponding log file, (C) the corresponding solution file, and (D) the evals versus fitness plot corresponding to the log in the log file. If it's not completely obvious how to compile your code, then include a readme file to explain how to do this. Submit all files in a .ZIP or gzipped tar ball format. The due date for this assignment is Sunday 4 September 2011.

Grading
The maximum number of points you can get is 50. The point distribution is as follows:

Algorithmic	30
Configuration files and parsing	5
Logging and output/solution files	5
Good programming practices including code reliability and commenting	5
Evals versus fitness plots	5

Assignment 1b: Evolutionary Algorithm Search

Implement a $(\mu + \lambda)$-EA with penalty function to find within a configurable number of fitness evaluations the minimal cardinality set of routers (i.e., the set with the least number of routers) for an arbitrary datafile. At minimum the following EA operators should be supported via your configuration file:

Representation Binary

Initialization Uniform Random

Parent Selection Fitness Proportional Selection, k-Tournament Selection with replacement

Recombination Uniform Crossover, n-point crossover

Mutation Bit-flip

Survival Selection Truncation, k-Tournament Selection without replacement

Termination Number of evals, no change in fitness for t evals

Your configuration file should allow you to select which of these configurations to use as well as how many runs a single experiment consists of. Your configurable EA strategy parameters should include all those necessary to support your operators, such as:

- μ
- λ
- k for parent selection
- n for n-point crossover
- Mutation bit-flip rate
- k for survival selection
- Number of evals till termination

- t for termination convergence criterion
- penalty coefficient

The penalty function should subtract from your fitness function a value dependent on the number of paths remaining uncut and the penalty coefficient (i.e., you can use the coefficient to increase or decrease the penalty pressure).

The result log should be headed by the label "Result Log" and consist of empty-line separated blocks of rows where each row is headed by a run-label of the format "Run i" where i indicates the run of the experiment and where each row is tab delimited in the form <evals><tab><average fitness><tab><best fitness> (not including the < and > symbols) with <evals> indicating the number of evals executed so far, <average fitness> is the average population fitness at that number of evals, and <best fitness> is the fitness of the best individual in the population at that number of evals (so local best, not global best!). The first row has $< \mu >$ as value for <evals>. Rows are added after each generation, so after each λ evaluations. The solution file should consist of the best solution found in any run.

The deliverables of this assignment are your source code and for each of the four datafiles made available to you, you need to submit (A) a configuration file configured for 10,000 fitness evaluations, 3 runs, and the best EA configuration you can find, (B) the corresponding log file, (C) the corresponding solution file, and (D) the evals versus fitness plot corresponding to the log in the log file. If it's not completely obvious how to compile your code, then include a readme file to explain how to do this. Submit all files in a .ZIP or gzipped tar ball format. The due date for this assignment is Sunday 18 September 2011.

Grading
The maximum number of points you can get is 100. The point distribution is as follows:

Algorithmic	80
Configuration files and parsing	5
Logging and output/solution files	5
Good programming practices including code reliability and commenting	5
Evals versus fitness plots	5

Assignment 1c: Multi-Objective EA

Implement the Pareto-front based MOEA called NSGA or or other instructor approved MOEA to optimize within a configurable number of fitness evaluations the Pareto set of router sets which balance minimizing the cardinality of the router sets and maximizing the number of cut network flow paths for an arbitrary datafile. You need to implement support for the EA configurations appropriate for your chosen MOEA, with at minimum two operators having more than one user selectable option. Your configuration file should allow you to select which of these configurations to use as well as how many runs a single experiment consists of. Your configurable EA strategy parameters should include all those necessary to support your operators.

The result log should be headed by the label "Result Log" and consist of empty-line separated blocks of rows where each block is headed by a run-label of the format "Run i" where i indicates the run of the experiment and where each row is tab delimited in the form <evals><tab><average fitness><tab><best fitness> (not including the < and > symbols) with <evals> indicating the number of evals executed so far, <average fitness> is the average population fitness at that number of evals, and <best fitness> is the fitness of the best individual in the population at that number of evals (so local best, not global best!). The first row has $< \mu >$ as value for <evals>. Rows are added after each generation, so after each λ evaluations. Note that while your MOEA will internally be differentiating between the fitness of each of the two objectives, for logging and comparison purposes you will use the same fitness function as you used in Assignment 1b. However, the solution file should consist of the final Pareto front found by the run with the highest final average fitness. It should be formatted as follows: each solution in the Pareto front should

have its own tab delimited row in the form <cardinality of router set><tab><number of cut network flow paths><tab><default solution format> (not including the < and > symbols).

All experiments in this assignment are on Network 3 with termination after at least 10,000 fitness evaluations (note: all experiments should use the same number of fitness evaluations). Informally experiment with the sensitivity of the final global best to the EA strategy parameters to determine which two parameters seem to make the most difference. Then formally experiment with the sensitivity of the final global best to the two parameters identified previously by at minimum trying three different values for each of the two parameters (so nine combinations) and collecting statistics for 30 runs. Use an appropriate statistical test (e.g., t-test or anova) to determine with $\alpha = 0.05$ which combinations are statistically better in terms of final global best. Make nine plots, one for each combination, with each plot showing evals vs. population mean fitness averaged over the 30 runs (fitness on the left vertical axis).

The deliverables of this assignment are your source code, the nine configuration files and associated nine log files, nine solution files, and nine plots, and a table containing your statistical comparison of the nine combinations. If it's not completely obvious how to compile your code and reproduce your results, then include a readme file to explain how to do this. Submit all files in a .ZIP or gzipped tar ball format. The due date for this assignment is Sunday 2 October 2011.

Grading
The maximum number of points you can get is 125. The point distribution is as follows:

Algorithmic	100
Configuration files and parsing	5
Logging and output/solution files	5
Good programming practices including code reliability and commenting	5
Choice of parameters	5
Nine plots	10
Statistical table	20

Bonus
You can earn up to 25 bonus points by adding a diversity maintenance technique to your NSGA implementation or by implementing a more advanced MOEA such as NSGA-II, SPEA-II, SNDL-MOEA, or *epsilon*-MOEA or other instructor approved MOEA. Note that a fully correct yet only moderately advanced techniques such as fitness sharing will not garner the full 25 bonus points which is reserved for a fully correct advanced technique such as NSGA-II.

Assignment 1d: EA Technical Report

For the algorithms you implemented and experimented on in assignments 1a-1c, you need to write a technical report. Your report needs to consist of the following sections:

Methodology Describe your EA/MOEA design in sufficient detail to allow someone else to implement a functionally equivalent EA/MOEA and explain your EA/MOEA design decisions.

Experimental Setup Provide your experimental setup in sufficient detail to allow exact duplication of your experiments (i.e., producing the exact same results within statistical margins) and justify your choice of EA/MOEA strategy parameters.

Results List your experimental results in both tabular and graphical formats along with your statistical results. Make sure to include a figure with the two objectives on the axises which plots the solutions found by your MOEA.

Discussion Discuss your experimental and statistical results, providing valuable insights such as conjectures you induce from your results. Your choice of what to report on and how you go about rationalizing it is your subjective interpretation.

Conclusion Conclude your report by stating your most important findings and insights in the conclusion section.

Bibliography This is where you provide your citation details, if you cited anything. Only list references here that you actually cite in your report.

Appendices If you have more data you want to show than what you could reasonably fit in the body of your report, this is the place to put it along with a short description.

This report needs to compare your nine MOEA results with each other, and also compare the best solutions found by your Assignment 1a random search, your Assignment 1b EA, and your Assignment 1c MOEA. You are encouraged to reuse your assignment results, including plots and statistical comparison table, except where you received feedback to improve/correct your assignment submission materials. The deliverable of this assignment is your technical report in PDF or Microsoft Word format. The due date for this assignment is Sunday 9 October 2011.

Grading
The maximum number of points you can get is 100. The point distribution is as follows:

Spelling	2
Grammar	3
Clarity	10
Typesetting	5
Methodology	20
Experimental setup	10
Experimental results	15
Statistical results	10
Discussion	20
Conclusion	5