

**Due date:** Friday March. 24, 2017. 23:50 PM

**Late submission:** No late submission is accepted.

**Evaluation:** 5 % of the final mark

**Teams:** only for COMP474 can do the assignment individually or in teams of 2. Teams must submit only 1 copy of the assignment.

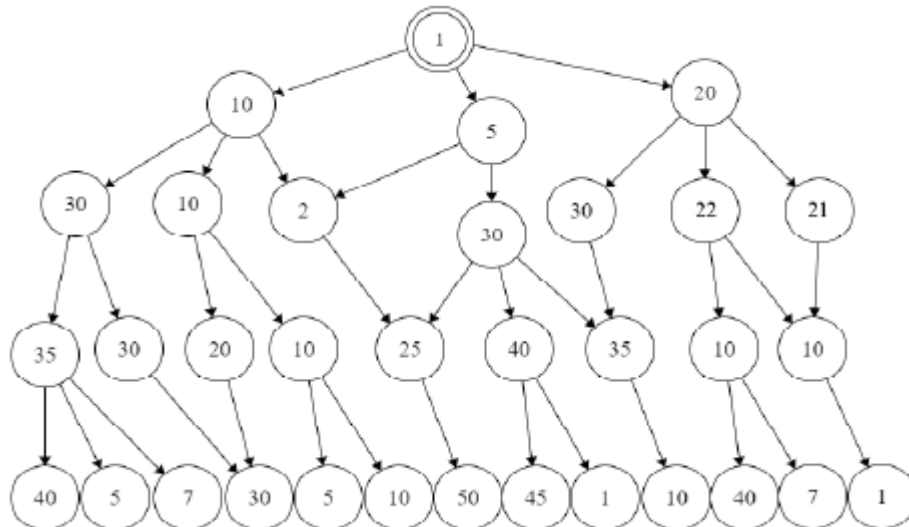
**Purpose:** The purpose of this assignment is to make you practice: local search, constraints satisfactions problems, and genetic algorithms

### Question 1 (20 pts)

In this question, we will be exploring two common local search algorithms: Hill-climbing and Simulated Annealing Search. Your task is to run each of these algorithm on the state graph below.

a) For each of the algorithms listed above, you should do the following tasks:

- Execute the algorithm on the graph.
- Record the path of execution by listing the nodes in order of traversal. In the case of Simulated Annealing Search, only record the nodes that you traverse, not the ones you consider traversing.
- Record the output of the algorithm.



The labels of each node is the objective value of that node. Note that we are trying to maximize in this problem, so high values are "good."

Note: For Simulated Annealing Search, there are two pieces of information you must know. These are the probability function,  $P(e, e', T)$  and the source of randomness. Define  $P$  as follows:

$$P(e, e', T) = \begin{cases} 1 & e' > e \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

Note that this function ignores the temperature parameter,  $T$ , for the sake of simplicity.

As for the source of randomness, use the following series of values,  $R$ , from the interval  $[0, 1]$  as "random" values. When you need to decide which of the  $n$  neighbors of a node to randomly choose, divide the range  $[0, 1]$  into  $n$

congruent **interval**, and then use the **next unused value from R** to see which interval and thus node to pick. When you are comparing values of  $P$ , note that you only need **consume a random value** when  $P(e, e', T) < 1$ . If  $P(e, e', T) > R_i$ , consider the random trial a "**success**," i.e., you should transition to that node.  
 $R = [.5; .25; .1; .75; .1; .9; .4; .8; .8; .2]$



### **Question 2 (8 pts)**

You are in charge of scheduling for computer science classes that meet **Tuesday, Thursday and Fridays**. There are **5** classes that meet on these days and **3 Instructors** who will be teaching these classes. You are constrained by the fact that each Instructor can only teach one class at a time.

The classes are:

1. Class 1 - COMP 335 Introduction to Theoretical Computer Science: meets from 8:00-9:00am
2. Class 2 - COMP346 Operating Systems: meets from 8:30-9:30am
3. Class 3 - COMP474 Intelligent Systems: meets from 9:00-10:00am
4. Class 4 - COMP442 Compiler Design: meets from 9:00-10:00am
5. Class 5 - COMP6321 Machine Learning: meets from 10:30-11:30am

The instructors are:

1. Instructor A, who is qualified to teach Classes 1, 2, and 5.
  2. Instructor B, who is qualified to teach Classes 3, 4, and 5.
  3. Instructor C, who is qualified to teach Classes 1, 3, and 4.
- a. Formulate this problem as a constraints satisfactions problem CSP problem in which there is one variable per class, stating the domains, and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.
  - b. Draw the constraint graph associated with your CSP.
  - c. Your CSP should look nearly tree-structured. Briefly explain (one sentence or less) why we might prefer to solve tree-structured CSPs.

### **Question 3 (12 pts)**

Four instructors, A, B, C and D have to give classes at the same time. 5 rooms are available: rooms 1, 2, 3, 4 and 5. Instructor A doesn't want to teach in room 1. Instructor B doesn't want to teach in room 2. Instructor D wants to teach in a room with number greater or equal to 3, yet strictly less than the number of the room B is teaching in. Instructor C doesn't want to teach in a room adjacent to that of B (rooms with successive numbers are adjacent), nor in room 5.

Obviously, we want to assign different rooms to different instructors.

- a. Formulate the above constraint problem in terms of constraints over the **variables A, B, C and D** and the **finite domain {1,2,3,4,5}**. Use **Forward Checking** to generate a **solution** to the problem by **drawing the search tree** (depth-first with forward checking as **arc-consistency method** between successive **assignments** of the back-tracking algorithm). Clearly indicate at each step which elements are **removed** from the domains.
- b. Finally, do forward checking again, but this time provide any heuristic search that optimize the solution.

### **Question 4 (15pts)**

In this question we tackle a (sort-of) cryptarithmic problem.

$$\begin{array}{r} AB \\ + CBA \\ \hline DBC \end{array}$$

The goal is to find numeric assignments to letters such that the above addition is correct. **Read this part carefully:** In this problem, we will require that A, B, and C are all different from each other. D is allowed to have the same value

as another letter. No leading zeros are allowed (i.e. A, C, and D cannot equal 0). Finally, the arithmetic for the above problem is done in base 5.

- What are the variables, domain, and constraints? Hint: You may find characters from the following set useful in writing your constraints:  $\{+, \equiv 5 \text{ (equivalence mod five)}, =, \neq, \Rightarrow, \leq, \geq\}$
- Now solve the problem using backtracking with forward checking. As heuristics, use *Minimum remaining values* MRV for selecting variables (break ties alphabetically) and assign values in numeric order when multiple options are available. Show the sequence of variable assignments, and state how many times the algorithm backtracks.

For example:

A: 2

A: 2 B: 2

A: 3

A: 3 B: 2

A: 3 B: 2 D: 1

A: 3 B: 2 D: 1 C: 4

We backtracked 1 time.

### **Question 5 (15 pts)**

Consider a genetic algorithm (GA) with chromosomes consisting of six genes  $x_i = abcdef$ , and each gene is a number between 0 and 9. Suppose we have the following population of four chromosomes:

$$x_1 = 4\ 3\ 5\ 2\ 1\ 6$$

$$x_2 = 1\ 7\ 3\ 9\ 6\ 5$$

$$x_3 = 2\ 4\ 8\ 0\ 1\ 2$$

$$x_4 = 9\ 0\ 8\ 1\ 2\ 3$$

and let the fitness function be  $f(x) = (a + c + e) - (b + d + f)$ .

1. Sort the chromosomes by their fitness.
2. Do one-point crossover in the middle between the 1st and 2nd fittest, and two-points crossover (points 2, 4) for the 2nd and 3rd.
3. Calculate the fitness of all the offspring

### **Question 6 (15 pts)**

Suppose a genetic algorithm uses chromosomes of the form  $x = abcdefgh$  with a fixed length of eight genes. Each gene can be any digit between 0 and 9. Let the fitness of individual  $x$  be calculated as:

$$f(x) = (a + b) - (c + d) + (e + f) - (g + h),$$

and let the initial population consist of four individuals with the following chromosomes:

$$x_1 = 6\ 5\ 4\ 1\ 3\ 5\ 3\ 2$$

$$x_2 = 8\ 7\ 1\ 2\ 6\ 6\ 0\ 1$$

$$x_3 = 2\ 3\ 9\ 2\ 1\ 2\ 8\ 5$$

$$x_4 = 4\ 1\ 8\ 5\ 2\ 0\ 9\ 4$$

- a) Evaluate the fitness of each individual, showing all your workings, and arrange them in order with the fittest first and the least fit last.
- b) Perform the following crossover operations:
  - i. Cross the fittest two individuals using one-point crossover at the middle point.

- ii. Cross the second and third fittest individuals using a two-point crossover (points b and f).
  - iii. Cross the first and third fittest individuals (ranked 1st and 3rd) using a uniform crossover.
- c) Suppose the new population consists of the six offspring individuals received by the crossover operations in the above question. Evaluate the fitness of the new population, showing all your workings. Has the overall fitness improved?
  - d) By looking at the fitness function and considering that genes can only be digits between 0 and 9 find the chromosome representing the optimal solution (i.e. with the maximum fitness). Find the value of the maximum fitness.
  - e) By looking at the initial population of the algorithm can you say whether it will be able to reach the optimal solution without the mutation operator?

### **Question 7 (15 pts)**

A budget airline company operates 3 plains and employs 5 cabin crews. Only one crew can operate on any plain on a single day, and each crew cannot work for more than two days in a row. The company uses all planes every day. A Genetic Algorithm is used to work out the best combination of crews on any particular day.

- a) Suggest what chromosome could represent an individual in this algorithm?
- b) Suggest what could be the alphabet of this algorithm? What is its size?
- c) Suggest a fitness function for this problem.
- d) How many solutions are in this problem? Is it necessary to use Genetic Algorithms for solving it? What if the company operated more plains and employed more crews?

### **Submission:**

The assignment must be handed-in electronically by midnight on the due date.

Before you handle in your assignment electronically:

- Create one zip file, containing all files for your assignment.
- Name your zip file this way:
  - For individual work: name the zip file: *a1\_studentID*, where *studentID* is your ID number.
  - For group work: name the zip file: *a1\_studentID1\_studentID2*, where *studentID1* and *studentID2* are the ID numbers of each student.
  -
- Upload your zip file on Moodle, under **Assignment 2 Submission** Box.