CS 182, PROBLEM SET 2

Due: October 13, 2021 11:59pm

This problem set covers Lectures 5, 6, 7. The topics include Constraint Satisfaction Problems, Convex Optimization, and Integer Programming.

1. *Comprehension.* (17 points)

(1) *CSPs.* You are in charge of scheduling classes There are 5 courses that meet on these days and 3 instructors who will be teaching these classes. Each instructor is only able to teach one class at a time. The class times are as follows:

- Class A runs from 9:00am-10:00am.
- Class B runs from 9:30am-10:30am.
- Class C runs from 10:00am-11:00am.
- Class D runs from 10:00am-11:00am.
- Class E runs from 10:30am-11:30am.

Each instructor is capable of teaching some subset of the course:

- Instructor X is able to teach classes C and D.
- Instructor Y is able to teach classes B, C, D, E
- Instructor Z is able to teach classes A, B, C, D, and E.

Here are the questions:

(a) (2 points) In order to assign each class an instructor, formulate the problem as a CSP. What are the variables, domains, and constraints?

(b) (1 points) Draw the constraint graph of the CSP that you just formulated.

(c) (2 points) What is the new CSP after enforcing arc consistency? Why is this new CSP easier to solve? Provide a solution to the new arc-consistent CSP, and verify that it is also a solution to the CSP in (a)

(2) *Convex Functions.* Are the following functions are convex? Provide proof.

(a) (3 points) $f : \mathbb{R}^n \to \mathbb{R}$, $f(\mathbf{x}) = \max_i x_i$. (For example, in $\mathbb{R}^2$, $f((x, y)) = \max(x, y)$.)

(b) (3 points) $f : \mathbb{R}^2_+ \to \mathbb{R}$, $f((x, y)) = x/y$, where $\mathbb{R}^2_+$ denotes the set $\{(x, y) \in \mathbb{R}^2 | y > 0\}$.

Date: September 29, 2021.

(c) (3 points) $f : \mathbb{R} \to \mathbb{R}$. Suppose $f_1, \ldots, f_n$ are all convex functions $\mathbb{R} \to \mathbb{R}$, and let $f(x) = \max\{f_1(x), \ldots, f_n(x)\}$.

(3) *Integer Programming.* (3 points)

(a) The famous Knapsack Problem in computer science is as follows. Given n items numbered $1, \ldots, n$ with weights $w_1, \ldots, w_n$ and values $v_1, \ldots, v_n$, respectively, how do we maximize the value of the items we take given a maximum weight capacity C? Formulate the Knapsack Problem as an Integer Program.

2. *Convex Sets.* (12 points)

(a) (4 points) Let $C \subseteq \mathbb{R}^n$ be a convex set, with $x_1, \dots, x_k \in C$, and let $\theta_1, \dots, \theta_k \in \mathbb{R}$ satisfy $\theta_i \geq 0$, $\theta_1 + \cdots + \theta_k = 1$. Show that $\theta_1 x_1 + \cdots + \theta_k x_k \in C$ for all $k \geq 1$. [1]

(b) Define the convex hull of a set of points S in $\mathbb{R}^n$ as the set of all convex combinations of points in S:

$$CH(S) = \{\theta_1 x_1 + \cdots + \theta_k x_k | x_i \in S, \theta_i \geq 0, i = 1, \dots, k, \sum_{i=1}^{k} \theta_i = 1\}$$

(a) (4 points) Show that $CH(S)$ is indeed a convex set.

(b) (4 points) Show that the convex hull of a set S is the intersection of all convex sets that contain S.

[1] Hint: When is this definition equivalent to the definition of convexity? Try induction.

3. *Integer Programming and CNFs.* (12 points) Define a *conjunctive normal form* (CNF) to be an expression that involves the AND ($\wedge$) of ORs ($\vee$) of variables $x_1, \ldots, x_n \in \{0,1\}$ and their negations (the negation of x is denoted by $\neg x$), collectively known as literals. A 3-CNF formula is an AND of ORs where every OR consists of exactly 3 literals, whereas a general CNF formula may have any number of literals within each OR. Each OR is known as a clause. Here is an example of a 3-CNF formula φ:

$$\varphi_1 = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_4 \vee \neg x_1) \wedge (x_5 \vee x_4 \vee x_3)$$

We say that a 3-CNF (or general CNF) formula φ is satisfiable if there is an assignment $(x_1, \ldots, x_n) \in \{0,1\}$ such that φ evaluates to true if we assign its variables the values of $(x_1, \ldots, x_n)$. For example, check that φ_1 above is satisfiable with the assignment $(x_1, x_2, x_3, x_4, x_5) = (1, 0, 0, 1, 1)$. The problem of determining whether or not a 3-CNF formula is satisfiable is known as the *3-SAT* problem.

(1) (4 points) Formulate the 3-SAT problem as an Integer Programming problem.

(2) (2 points) Now suppose that you took your integer programming problem from (1) and turned it into a linear programming problem (i.e. removed the constraint that all variables must take on integers while keeping all the other constraints). Given a 3-CNF formula, what can you say about whether or not it is satisfiable in this new situation?

(3) (6 points) Suppose now that we relax the condition that $x_i \in \{0,1\}$ to a new condition that $x_i \in [0,\epsilon] \cup [1-\epsilon, 1]$, where $\epsilon = 0.0001$. Is there an example of a 3-CNF formula and corresponding assignment where we obtain a solution for this relaxed problem that wasn't present in the original problem? What about for a general CNF? For both cases, either give an example or prove that there does not exist one.

4. *Programming.* (15 points) The n-queens problem is the problem of placing n queens on a chessboard so that no two queens attack each other. (For those of you unfamiliar with chess, a queen attacks a square if it is on the same row, column, or diagonal as a square). Formulate this problem as an Integer Program, then follow the instructions and template in the starter code to use the `cvxpy` Python package to solve this problem via Integer Programming. An autograder on Gradescope will verify the correctness of your formulation.