Core Question – Principles of Autonomy and Decision Making:

Please note there are **3** core questions. Please answer any **2 of 3**, but **NOT** all three.

Core Question 1: Propositional Logic Modelling.

Encode the following statement in propositional logic. Then reduce to conjunctive normal form:

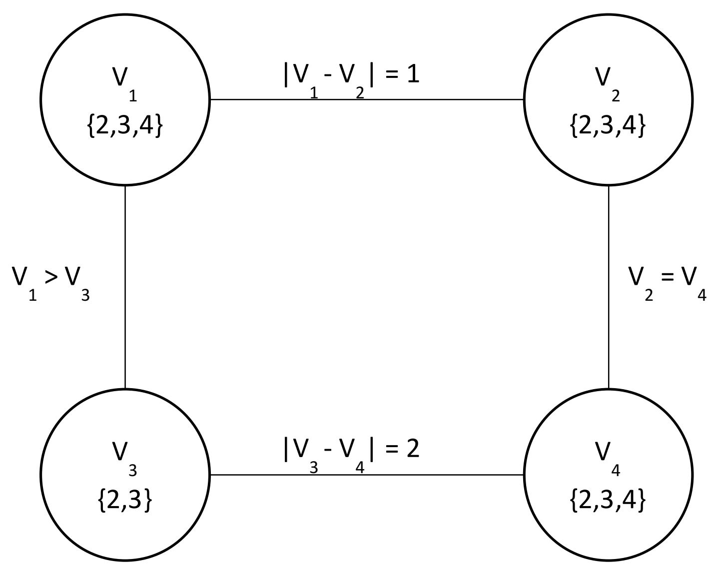
An autonomous car is shared between two users for on-demand use. On a given day, the car may be used by traveler A (denoted by proposition UA), and/or by traveler B (denoted by UB). If, and only if, both use the car (UA ∧ UB), they either make two individual uses (denoted by TI) or they carpool (denoted by CP), but not both. The car needs to be refueled (denoted by CR) after two uses that day, but should not be refueled otherwise. Please note there are exactly 5 propositions: UA, UB, TI, CP and CR.

Please note there are **3** core questions. Please answer any **2 of 3**, but **NOT** all three.

Core Question 2: Constraint Satisfaction

Satisfaction: Describe how **back-track search with forward checking** can be used to solve the following Constraint Satisfaction Problem, and show the search tree for the first step of the algorithm (and only the first step). The CSP is depicted graphically below. The circles represent variables with their initial domains enclosed, and lines between variables represent binary constraints, as labeled. Please show the progression of the algorithm using a search tree. Number the tree nodes in the order visited. Label each node with the variable domain elements that remain after pruning.

DO NOT use dynamic variable ordering. Instead, assign variable V_1, then V_2, then V_3, then V_4, and select domain elements in the order 2, then 3, then 4.



Please note there are **3** core questions. Please answer any **2 of 3**, but **NOT** all three.

Core Question 3: Markov Decision Processes

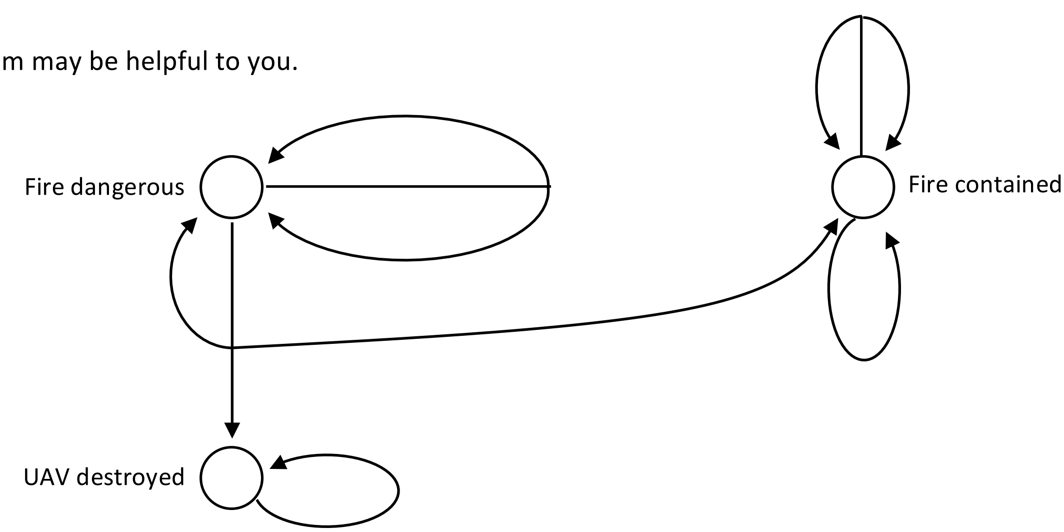
Given the following scenario description, model the system as an MDP and describe how the optimal policy can be computed. We do **not** want you to solve for the optimal policy.

An unmanned aerial vehicle assists during forest fires by fulfilling two functions: a) ***search*** *for civilians in areas of risk, and b)* ***drop*** *retardant on the forest fire. The fire is classified as '****contained'*** *or '****dangerous,'*** *and changes as a result of the drop action. The UAV performs an action after every time step, with a fixed duration between time steps. The mission is long-term, and is* ***approximated as infinite*** *in duration. Future reward is discounted, with each successive action being only 0.8 times as valuable as an action at the preceding time step.*

The action of dropping retardant on a dangerous fire will cause the UAV to be ***destroyed*** *with 5% probability (in which case the mission ends and no reward is gained). There is no chance of failure when either the fire is contained or the UAV is searching. When the UAV is not destroyed, dropping retardant on a dangerous fire has a 10% probability of containing that fire. In addition, when the fire is dangerous, if dropping retardant succeeds at containing a dangerous fire, it nets a reward of 20; otherwise, the fire remains dangerous and the reward is 10. Dropping retardant on a contained fire will always result in a contained fire, with a reward of 5.*

The action of searching for civilians will find civilians 10% of the time and will fail to find civilians 90% of the time. No damage is inflicted to the AUV by searching. Finding civilians is worth a reward of 100 when the fire is dangerous, and a reward of 20 when the fire is contained. Failure to find civilians always nets 0 reward.

This diagram may be helpful to you.



Estimation

Let us assume a simplified Dubins car in 2D that does not know its pose $\mathbf{x} = (x, y, \theta)$. The vehicle dynamics model is given according to

$$\begin{aligned} x_{k+1} &= x_k + \cos(\theta_k) + w_{x,k+1} \\ y_{k+1} &= y_k + \sin(\theta_k) + w_{y,k+1} \\ \theta_{k+1} &= \theta_k + \tan(\phi_k) + w_{\theta,k+1} \end{aligned}$$

where ϕ_k is the steering angle. Notice that this is a simplified Dubins car that is travelling at a constant velocity of 1 m/s. The noise terms $w_{\cdot,k}$ are distributed according to a zero-mean Gaussian with known covariance W,

$$\begin{bmatrix} w_x \\ w_y \\ w_\theta \end{bmatrix}_k \sim \mathcal{N}(0, W).$$

The vehicle can estimate its position and orientation from the bearing to a set of active beacons distributed in the plane, each with known position. The vehicle has an initial position and orientation estimate given by a Gaussian with mean $\hat{\mathbf{x}}_0$ and covariance Q_0. At each time step k, the vehicle receives some number n_k of bearing measurements, where the bearing $b_{i,k}$ is to the i^{th} beacon at position $(l_{x,i}, l_{y,i})$ with Gaussian noise such that

$$\begin{aligned} b_{i,k} &= atan2(l_{y,i} - y_k, l_{x,i} - x_k) - \theta_k + v_k \\ v_k &\sim \mathcal{N}(0, r^2), \end{aligned}$$

where r^2 is known and the same for every beacon.

The vehicle knows which beacon generated each bearing, and the position of each beacon. You may assume that a signal is received from every beacon on every step.

- Please identify an algorithm for inferring the distribution $p(\mathbf{x}_t | b_{0,t})$. (Any derivatives you may need, you can leave unsolved.)
- Does your answer change if the vehicle trajectory keeps the vehicle very far from the beacons? Does your answer change if the vehicle trajectory causes the vehicle to travel very close to the beacons?
- Please identify an algorithm for inferring the distribution $p(\mathbf{x}_t | b_{0:T}), t \in \{0, T\}$, that is, the position of the vehicle at any point in the past.
- Imagine that you no longer know the beacon positions. How does your model change? Do your inference algorithms change?