CSP Problem Set
CSE422 (IBU)
Deadline: 4th April, 2024

Submission Link: https://forms.gle/rkLHcrPkz8wiHvt26

Conceptual Questions

1. Explain what a Constraint Satisfaction Problem (CSP) is. List and describe the three main components of a CSP.
2. Describe how the backtracking algorithm works for solving CSPs. Include in your answer how backtracking deals with dead ends (i.e., situations where no solution is found).
3. Explain the concept of forward checking in the context of CSPs. How does it enhance the basic backtracking algorithm? Provide an example where domain reduction significantly impacts the problem-solving process.
4. Discuss the role of heuristics in solving CSPs. Describe two common heuristics used in the selection of variables and values in backtracking algorithms for CSPs.
5. What are global constraints in CSPs, and how do they differ from binary constraints? Provide an example of a global constraint and explain its utility.
6. Discuss the role of constraints in defining the structure and complexity of a CSP. How can the nature of constraints affect the difficulty of solving a CSP?
7. Explain how heuristics impact the search process in solving CSPs. Are there any potential downsides to relying on heuristics?
8. Compare the efficiency of the backtracking algorithm with and without forward checking. Under what circumstances does forward checking significantly improve backtracking performance?

EXAMPLE: Simulation of Backtracking

Four jobs (J1 through J4) need processing on three machines (M1, M2, M3), with each job requiring a unique sequence of operations across these machines.

Job Details:

- J1: Operates on M1 for 2 hours, then M2 for 1 hour, and finally M3 for 1 hour.
- J2: Operates on M3 for 2 hours, then M1 for 1 hour.
- J3: Needs processing on M2 for 1 hour, then M3 for 1 hour, and finally M1 for 2 hours.
- J4: Requires M2 for 2 hours, then M1 for 1 hour, and ends with M3 for 1 hour.

Variables:

- Start_Time(Job, Machine): Represents the start time of each job on each machine.

Domains:

- Each operation can start at any hour within a 9-hour workday (0 to 8, where 0 represents the start of the workday).

Constraints:

Machine Availability: Each machine can only perform one operation at a time.
Operation Sequence: Each job's operations must be processed in the specified sequence.
Working Hours Limitation: All operations must be scheduled within the workday.

Backtracking Algorithm Simulation without Forward Looking:

Start J1 on M1 at 0: J1 begins on M1 since it's the first operation in its sequence.
Attempt to place J1 on M2: We see J1 can go on M2 at time 2, after completing its operation on M1.
Place J1 on M3: After M2, J1 moves to M3, starting at time 3.
Move to J2, starting on M3: J2 starts on M3 at the next available slot, which is time 4 (conflict with J1's schedule). Backtrack to adjust J1's start time on M3 or reprioritize J2's start.

- Adjustment: To accommodate, we may start J2 on M3 at time 0 and adjust J1's operation on M3 to start later.

Continue with J3 and J4, adjusting their start times on each machine, respecting the machine availability and job sequence. Conflicts might require backtracking to earlier job schedules.

Backtracking Algorithm Simulation without Forward Looking (Domain Reduction):

After scheduling an operation: Immediately reduce the availability domain for that machine for the operation's duration. **(This is the same process as what we did in class)**
Before scheduling each job's operation: Check the reduced domains to ensure no conflicts with machine availability.

For J1: After placing it on M1, reduce M1's domain to reflect its unavailability for the next 2 hours.
For subsequent jobs (J2, J3, J4): Schedule considering reduced domains. For instance, knowing M3 is taken at the start by J2, place J3 and J4 on their first operations in available slots, ensuring forward compatibility.

Forward Looking/Domain reduction helps identify: Where jobs cannot be scheduled without violating constraints, allowing for adjustment before committing to a scheduling decision.

Problem Formulation and Simulation

For each of the situations given below, define the variables, domain and values, constraints. Make sure that the formulation is node consistent. Simulate using backtracking with forward looking.

1. **Schedule 10 CSE courses** with unique timeslots over 5 days, each day having 4 time slots. There are 5 instructors, each teaching 2 courses; no instructor can teach 2 courses at the same time. There are 2 lecture halls. Aim to avoid having more than 2 CSE courses on the same day.
2. **Job Scheduling:** A bakery needs to prepare 3 different products (bread, cake, cookies) using 3 stages of preparation (mixing, baking, packaging) within a single day. Each stage starts on the hour from 4 AM to 8 PM, with each stage lasting 2 hours. Each product must go through the stages in order: mixing → baking → packaging. Only one product can be baked at a time. Products must be ready for sale by 8 AM for bread and cookies, and by 12 PM for cake.
3. **Schedule 3 software development tasks** (coding, testing, deployment) for 2 projects over 2 days, considering resource availability. Tasks can be scheduled in 4-hour blocks starting from 9 AM to 5 PM each day. Tasks for each project must be completed in order: coding → testing → deployment. Only one project can be in the testing stage at any given time due to limited testing resources. No project can have more than one task scheduled per day to ensure team bandwidth.
4. **Consider a simplified map with 5 regions** (A, B, C, D, E) where some regions share borders. The adjacency constraints are as follows: A shares borders with B, C, and D. B shares borders with A, C, and E. C shares borders with A, B, and D. D shares borders with A and C. E shares borders with B. The goal is to color this map using the fewest colors possible, ensuring no two adjacent regions share the same color.
5. **Schedule lab sessions for 5 CSE courses**, each with 1 lab section, across 5 days. You have 10 available time slots (2 per day), across 2 lab rooms. Each course has 1 TA; TAs cannot oversee two labs simultaneously. 2 courses require a specific software setup available in only 1 lab room. Each lab session lasts 2 hours, requiring scheduling at the start of available time slots.
6. **Schedule classes for three subjects** (Math, Science, English) across three time slots in a single day, ensuring no teacher or student group has overlapping classes.
   Subjects and Teachers: Math (taught by Mr. A), Science (taught by Ms. B), English (taught by Mr. C). **Student Groups:** Group 1: Needs to attend all three subjects. Group 2:

Needs to attend Math and Science. Group 3: Needs to attend Science and English. **Time Slots:** 9:00 AM, 10:00 AM, and 11:00 AM.

7. **Schedule four workshops** (W1, W2, W3, W4) into three time slots (T1, T2, T3) throughout a single day, ensuring that attendees can participate in as many distinct workshops as possible without any scheduling conflicts. **Workshops and Leaders:** W1 (led by Leader A), W2 (led by Leader B), W3 (led by Leader C), W4 (led by Leader D). **Attendee Preferences:** Attendee Group 1 wants to attend W1 and W2. Attendee Group 2 is interested in W2 and W3. Attendee Group 3 prefers W3 and W4. **Time Slots:** T1: 9:00 AM - 10:30 AM. T2: 11:00 AM - 12:30 PM. T3: 2:00 PM - 3:30 PM.
8. Arrange 6 books (A, B, C, D, E, F) such that Book A next to Book B, Book C not next to Book D, Book E at one end.

ALSO SOLVE:

Problem 6.1, 6.2, 6.3, 6.4, 6.7, 6.8, 6.16 from Text book **Artificial Intelligence, a modern approach by Russell and Norvig (third edition)**