Thomas Jansen
University College Cork

Sample Test

The time for the test is 50 minutes. Answer all questions. The maximum number of marks is 20 (translating to 10 marks in the final grade).

Name: **Student #:**

Topic I) Task Environments (2 marks)

Give a PEAS description for an adaptive traffic light control at an intersection where two streets intersect. Classify the corresponding task environment using the list of seven dimensions from the lecture.

Performance measure should be related to the throughput taking into account traffic laws. The **environment** consists of the streets and the people using it (cars, pedestrians, ...). **Actuators** are the actual traffic lights. **Sensors** could be induction loops (to be aware of moving traffic), cameras (to have a more precise idea), possibly a signal from pedestrians.

- (depending on the sensors) partially observable
- single agent
- stochastic
- sequential
- dynamic
- continuous
- known

Topic II) Rational Agents (2 marks)

Compare a table-driven agent to a model-based agent.

The table-driven agent stores the sequence of all percepts. It has a table that specifies for each possible sequence of percepts the appropriate action. It is structurally simple and unlimited in what it can do. However, the table usually needs to be huge (in many cases infinite) making it more a theoretical model than a real option.

The model-based agent has a model of the current status of the world that it updates based on the percepts and the action it takes. It has a set of rules that lets it chose an action based on the current status of its world model. It is a flexible and powerful kind of agent. It's usefulness is largely determined by the quality of the model of the world.

Topic III) Uninformed Search (2 marks)

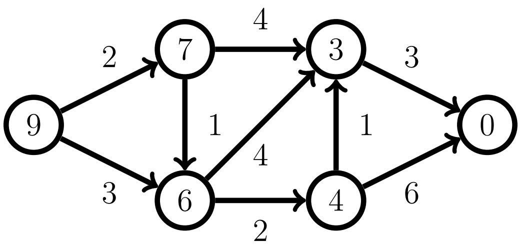
Remember the 8 puzzle. Consider using BFS or DFS for solving this puzzle and compare the two algorithms.

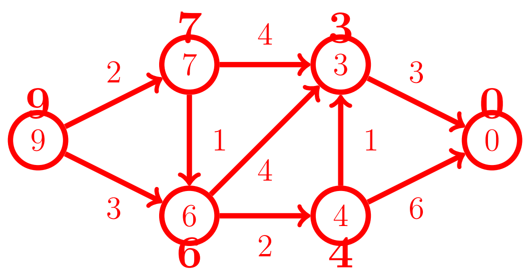
The 8 puzzle has a finite number of states. Therefore, both BFS and DFS are guaranteed to find a solution. BFS is guaranteed to find a solution with a minimal number of moves since it is optimal in the uniform cost model. For DFS there is no guarantee which solution is found. It may be the case that DFS is faster then BFS but this depends on the order in which moves are considered and would therefore be just luck. DFS is more memory efficient as BFS but for such a small game this is not really an issue.

Topic IV) Informed Search (2 marks)

The graph below specifies a search problem. States are nodes, the start state is 9, the only goal state is 0. Edges describe actions and their outcome, the number on an edge gives its cost. We use the number in the nodes as heuristic.

For each node write next to it the minimal cost to reach the goal state from there. Decide if this heuristic is admissible. Decide if this heuristic is consistent.



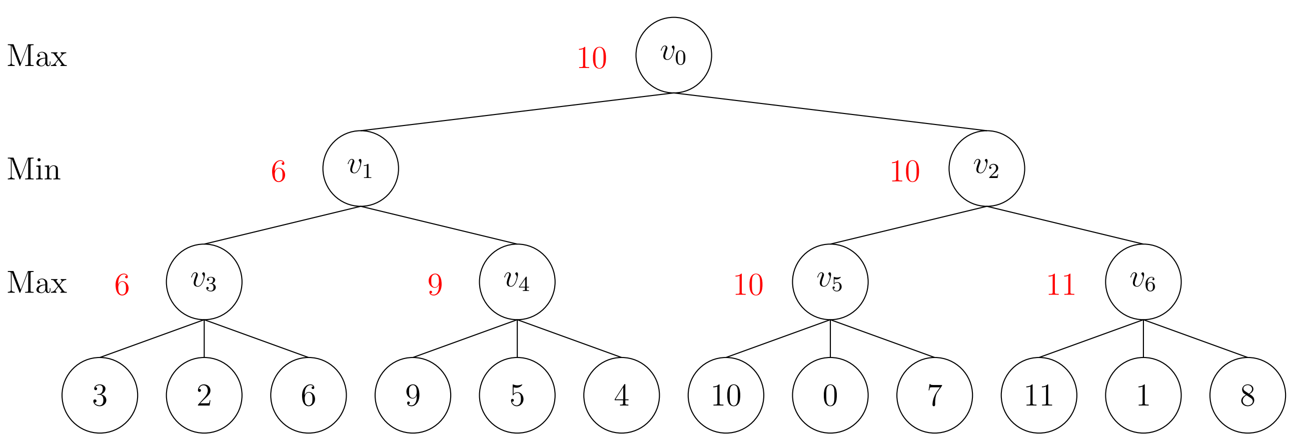


The heuristic is admissible because it does not overestimate.

The heuristic is consistent because the triangle inequality holds everywhere.

Topic V) Alpha-Beta Pruning (2 marks)

Evaluate the game tree given below using alpha-beta pruning. Consider the moves from left to right. Write down a list of the states that are not considered. Write next to each state that is evaluated the value that is computed by alpha-beta pruning.



5, 4, 1, and 8 are not explored.

Topic VI) Adversarial Search (2 marks)

How are cut-off tests used to speed up the evaluation of games trees? Name consequences of introducing cut-off tests.

A cut-off test is used to decide if the evaluation should stop early at a non-terminal node. This way parts of the game tree are not considered which reduced the time needed to evaluate the game tree. Since values are needed for each node there needs to be a value for a non-terminal where the evaluation is stopped. This is done by evaluation functions that estimate the value of a non-terminal based on the current state of the game in that node.

Topic VII) Random Processes (2 marks)

Consider the coupon collector's scenario with n different types of coupons. Assume that you want to collect at least m copies of each type of coupon. Let T be the number of coupons collected until your collection is complete. Prove $\mathrm{E}(T) = \Theta(nm + n \log n)$.

We begin with the lower bound. We observe that a complete collection contains at least $m \cdot n$ coupons and thus $T \geq nm$ holds. On the other hand, we have $\mathrm{E}(T) = \Omega(n \log n)$ since this bounds already holds for $m = 1$. Together this implies $\mathrm{E}(T) = \Omega(nm + n \log n)$.

For the upper bound consider the situation after $4nm+4n \ln n$ coupons have been collected. The expected number of coupons of each type equals $4m + 4 \ln n$. We apply Chernoff bounds to derive an upper bound on the probability that we have collected the first type of coupons less than m times. Let X denote the number of coupons collected of the first type. We introduce indicator variables $X_1, X_2, \ldots, X_{4nm+4n \ln n}$. We have $X_t = 1$ if the t^{th} coupon we get is of type 1 and 0 otherwise. We have $X = X_1 + X_2 + \cdots + X_{4nm+4n \ln n}$ and $\mathrm{E}(X) = 4m+4 \ln n$. Since the X_i are completely independent we can apply Chernoff bounds: $\mathrm{Prob}(X < m) < \mathrm{Prob}(X < m + \ln n) = \mathrm{Prob}\left(X < \left(1 - \frac{3}{4}\right) \mathrm{E}(X)\right) < e^{-(3/4)^2(4m+4\ln n)/2} = e^{-(9/8)(m+\ln n)} < e^{-(9/8)\ln n} = \left(\frac{1}{n}\right)^{9/8}$ We apply the union bound and see that the probability that there is at least one type of coupons where we have less than m coupons is bounded by $n \cdot (1/n)^{9/8} = (1/n)^{1/8} < 1/2$. Thus, we are done with probability at least $1/2$. If we are not done we repeat the process, i. e., collect another $4nm + 4n \ln n$ coupons. The expected number of times we have to repeat the process is bounded above by $(1/2)^{-1} = 2$. This proves $\mathrm{E}(T) \leq 8nm + 8n \ln = O(nm + n \log n)$.

Together we have $\mathrm{E}(T) = \Theta(nm + n \log n)$ as claimed.

Topic VIII) Black Box Complexity (2 marks)

We call a function $f\colon \{0,1\}^n \to \mathbb{R}$ a function of unitation if its function value for x only depends on the number of 1-bits in x and not x itself. Let S denote the set of all functions of unitation where the function value is strictly increasing with the number of 1-bits. Prove an upper bound on the black-box complexity of S.

Since the function value strictly increases with the number of 1-bits it is maximal for the all ones bit string. Thus, a black-box algorithm for S needs only to sample this one bit string. This proves $\mathcal{B}_S = 1$.

Topic IX) Applying Randomised Search Heuristics (2 marks)

Consider attempting to solve a difficult optimisation problem with some evolutionary algorithm. Discuss advantages and disadvantages of using crossover.

Crossover is a variation operator with very much different properties than mutation. While mutation is restricted to generate search points that are close the parent crossover may create offspring with a large Hamming distance from both parents. This allows for the exploration of parts of the search space that may be very difficult to reach with only mutation. Depending on the problem structure this may be very useful or even essential for success. On the other hand, crossover requires a population which increases the computational effort. It can only be effective if the members of the population are not too similar which may require additional algorithmic steps.

Topic X) Analysing Evolutionary Algorithms (2 marks)

Consider the (1+1) EA with mutation probability $p_m = 1/n$ on the function $f\colon \{0,1\}^n \to \mathbb{R}$ below. Prove an upper bound on the expected optimisation time $\mathrm{E}\left(T_{(1+1)\ \mathrm{EA},f}\right)$.

$$f(x) = \begin{cases} n+i & \text{if } x = 1^i 0^{n-i} \text{ with } i \in \{0,1,2,\dots,n\} \\ n - \textsc{OneMax}(x) & \text{otherwise} \end{cases}$$

We use f-based partitions and define one set for each function value. We have function values $v \in \{0,1,2,\dots,2n\}$. For $v < n$ the situation is similar to OneMax: It suffices to flip one of the $n-v$ 0-bits. The probability for this is $((n-v)/n)\cdot(1-1/n)^{n-1} \geq (n-v)/(en)$.

For $v \geq n$ it suffices to flip the left-most 0-bit to 1. This happens with probability $(1/n)(1-1/n)^{n-1} \geq 1/(en)$. Together we get the following bound.

$$\mathrm{E}\left(T_{(1+1)\ \mathrm{EA},f}\right) \leq \left(\sum_{v=0}^{n-1} \frac{en}{n-v}\right) + \left(\sum_{v=n}^{2n-1} en\right) \leq \left(en \sum_{v=1}^{n} \frac{1}{v}\right) + \left(en^2\right)$$
$$= O(n \log n) + O\left(n^2\right) = O\left(n^2\right)$$

'Cheat Sheet' (Part 1)

Markov inequality for $X \geq 0$ random variable; $s \in \mathbb{R}^+$: $\mathrm{Prob}\left(X \geq s \cdot \mathrm{E}\left(X\right)\right) \leq 1/s$

Chernoff bounds for $X_1, X_2, \dots, X_n \in \{0,1\}$ independent random variables; $X = \sum\limits_{i=1}^{n} X_i$:

$\forall \delta > 0$: $\mathrm{Prob}\left(X > (1+\delta)\cdot \mathrm{E}\left(X\right)\right) < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\mathrm{E}(X)}$

$\forall 0 < \delta < 1$: $\mathrm{Prob}\left(X < (1-\delta)\cdot \mathrm{E}\left(X\right)\right) < e^{-\mathrm{E}(X)\delta^2/2}$

Martingale $Y_0, Y_1, Y_2, \dots \in \mathbb{R}$ is a martingale with respect to a random process $X_0, X_1, X_2, \dots$ if the following hold for all $n \in \mathbb{N}_0$.

1. Y_n is a function of $X_0, X_1, \dots, X_n$
2. $\mathrm{E}\left(|Y_n|\right) < \infty$ or $Y_n \geq 0$
3. $\mathrm{E}\left(Y_{n+1} \mid X_0, X_1, \dots, X_n\right) = Y_n$

'Cheat Sheet' (continued)

Optional Stopping Theorem Let $Y_0, Y_1, Y_2, \ldots$ be a martingale with respect to $X_0, X_1, X_2, \ldots$. Let T be a stopping time of $X_0, X_1, X_2, \ldots$. If $\exists k \in \mathbb{N}_0 \colon T \leq k$ almost surely or $T < \infty$ and $\exists k \colon |Y_t| \leq k$ for all $t < T$ almost surely then $\mathrm{E}(Y_T) = \mathrm{E}(Y_0)$.

Gambler's ruin Consider two gamblers, one with initial funds s_A, winning in each round 1 with probability p_A, the other with initial found s_B, winning in each round 1 with probability $p_B = 1 - p_A$. They keep playing until one player runs out of money, i. e., is ruined. Let $p_A \neq p_B$, $q := p_B / p_A$.
$\text{Prob}(\text{A is ruined}) = (q^{s_A} - q^{s_A + s_B}) / (1 - q^{s_A + s_B})$
$\mathrm{E}(\text{rounds they play}) = ((1 - \text{Prob}(A \text{ ruined}))(s_A + s_B) - s_A) / (p_A - p_B)$

Collecting coupons Let T denote the number of coupons one collects until one has at least one coupons of in total n different types. The coupons are obtained one by one, each type with equal probability.
$\mathrm{E}(T) = n \ln(n) + O(n)$
$\forall \beta > 1 \colon \text{Prob}(T > \beta n \ln n) \leq n^{-(\beta - 1)}$
$\forall c \in \mathbb{R} \colon \text{Prob}(T > n \ln n + cn) = 1 - e^{-e^{-c}}$

number of non-empty classes of function $\mathcal{F} \subseteq R^S$ that are closed under permutation of the search space equals $\left(2^{\binom{|S|+|R|-1}{|S|}} - 1\right) / \left(2^{|R|^{|S|}} - 1\right)$

ANFL theorem Let $S = \{0,1\}^n$, $R = \{0, 1, \ldots, N-1\}$, $f \colon S \to R$, A a randomised BBA. The number of functions $f' \colon S \to R \cup \{N\}$ sucht that A does not find an optimum of f' within $2^{n/3}$ f-evaluations with probability at least $1 - 2^{-n/3}$ is bounded below by $N^{2^{n/3} - 1}$.
Of these exponentially many have the additional property that their complexity (measured by evaluation time, circuit size, or Kolmogoroff complexity) is by $O(n)$ larger than that of f.

f-based partitions For $f \colon \{0,1\}^n \to \mathbb{R}$ and $k \in \mathbb{N}$ the sets $L_0, L_1, \ldots, L_k$ are an f-based partition iff the following hold.

1. $\bigcup\limits_{i=0}^{k} L_i = \{0,1\}^n$, $\forall i \neq j \colon L_i \cap L_j = \emptyset$
2. $\forall i \neq j \colon \forall x \in L_i, y \in L_j \colon (i < j) \Rightarrow (f(x) < f(y))$
3. $L_k = \{x \in \{0,1\}^n \mid f(x) = \max\{f(y) \mid y \in \{0,1\}^n\}\}$

Metropolis algorithm

1. $t := 1$; Choose $x_t \in S$ uniformly at random.
2. Select $y \in N(x_t)$ uniformly at random.
3. $t := t + 1$; With prob. $\min\{1, e^{(f(y) - f(x_{t-1}))/T}\}$ set $x_t := y$, else $x_t := x_{t-1}$.
4. If 'not stop' continue at line 2.
5. Output x_t.

Steady state GA

1. Choose $x_1, x_2, \ldots, x_\mu \in \{0,1\}^n$ uniformly at random.
2. With probability p_c
   Select $z_1, z_2 \in \{x_1, x_2, \ldots, x_\mu\}$ u. a. r., $z := \text{crossover}(z_1, z_2)$
3. Else select $z \in \{x_1, x_2, \ldots, x_\mu\}$ uniformly at random.
4. $y := \text{standard bit mutation}(z)$
5. Select new $x_1, x_2, \ldots, x_\mu$ out of best $x_1, x_2, \ldots, x_\mu, y$.
6. If 'not stop' continue at line 2.
7. Output x_i with $f(x_i) = \max\{f(x_j) \mid 1 \leq j \leq \mu\}$.