CSE 4355 | Week 5 | Class 2 | Sunday · 9:30 AM – 10:50 AM | 80 min

Fuzzy logic and genetic algorithms

Two ideas borrowed from people and from nature: reason with 'quite hot' instead of 'hot or not', and search by breeding solutions instead of improving one. Then: what to expect in the midterm.

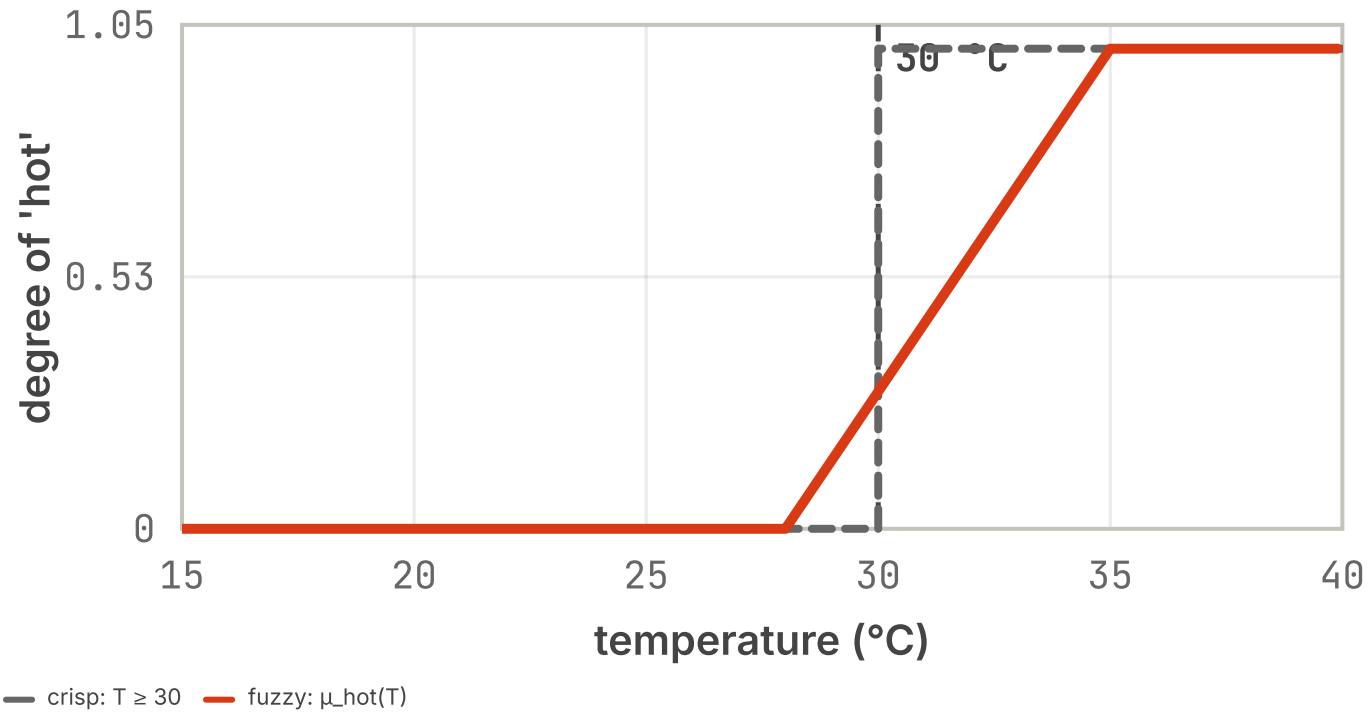
---

01 COLD OPEN 0–4 min

Is 30 °C hot?

Hands up: is **30 °C** hot? Is **29.9 °C**?

A crisp rule "hot if T ≥ 30" says 29.9 is not hot and 30.0 is. Nobody in Dhaka thinks that. People say "a bit hot", "quite hot", "very hot".



> On the fuzzy curve, **0.29 hot** (and, as we will see, 0.57 warm). Membership is a degree between 0 and 1, and a value can belong to two sets at once.

02 CONCEPT 4–9 min

Fuzzy sets: membership is a degree

- A crisp set: x is in or out, μ(x) ∈ {0, 1}. A **fuzzy set** A: a membership function μ_A: X → [0, 1].
- A **linguistic variable** (temperature) has fuzzy values (cool, warm, hot) that **overlap**: 30 °C is 0.57 warm and 0.29 hot.
- Shapes are a design choice; triangles and trapezoids are standard because they are cheap and easy to tune.

$$\mu_{tri}(x;\, a, b, c) = \max\Big(0,\ \min\Big(\frac{x-a}{b-a},\ \frac{c-x}{c-b}\Big)\Big)$$

FUZZY ≠ PROBABILITY

"0.7 chance that the glass is poisoned" and "the glass is 0.7 full of poison" are different statements. Probability is about **uncertainty** whether something is true; fuzziness is about the **degree** to which it is true, with nothing left to find out.

POP CULTURE

Lotfi Zadeh, an electrical engineer at Berkeley, introduced fuzzy sets in 1965. Japanese engineers made them famous in the 1980s.

03 CONCEPT 9–13 min

Fuzzy AND, OR, NOT, and hedges

OPERATION	ZADEH	M_A = 0.7, M_B = 0.4
A AND B	min(μ_A, μ_B)	0.4
A OR B	max(μ_A, μ_B)	0.7
NOT A	1 − μ_A	0.3
very A	μ_A^2	0.49
somewhat A	$\sqrt{\mu_A}$	0.84

GOTCHA

A AND NOT A = min(0.7, 0.3) = **0.3**, not 0. A OR NOT A = 0.7, not 1. The laws of contradiction and excluded middle do **not** hold.

PRO TIP

With μ only 0 or 1, min/max/1 − μ reduce to ordinary Boolean AND/OR/NOT: fuzzy logic contains crisp logic.

04 WORKED EXAMPLE 13–19 min

Worked example: a fuzzy fan controller, by hand

GIVEN

Room at 30 °C, 70 % humidity. Temperature: cool (15, 15, 20, 26), warm (20, 27, 34), hot (28, 35, 40, 40). Humidity: dry (30, 30, 45, 65), humid (50, 75, 100, 100). Fan: low (0, 0, 15, 45), medium (25, 50, 75), high (55, 85, 100, 100). Rules: R1 cool → low; R2 warm ∧ dry → low; R3 warm ∧ humid → medium; R4 hot → high.

FIND

The fan speed by Mamdani inference with centroid defuzzification (sample every 10 %).

Method: Mamdani inference: fuzzify, AND = min, clip, OR = max, centroid

SOLUTION

1. **Fuzzify.** Read each membership function at the input:

$$\mu_{warm}(30) = \tfrac{34-30}{7} = 0.57, \quad \mu_{hot}(30) = \tfrac{30-28}{7} = 0.29, \quad \mu_{humid}(70) = \tfrac{70-50}{25} = 0.8$$

2. (cool, dry are 0.) **Rules**, AND = min:

$$R_3:\ \min(0.57,\ 0.8) = 0.57 \to \mathit{medium} \qquad R_4:\ 0.29 \to \mathit{high} \qquad R_1 = R_2 = 0$$

3. **Aggregate.** Clip "medium" at 0.57 and "high" at 0.29; the output fuzzy set is their max. Sampled every 10 %:

$$\mu(z):\ \underset{0\text{–}20}{0},\ \underset{30}{0.2},\ \underset{40,50,60}{0.57},\ \underset{70\text{–}100}{0.29}$$

4. **Defuzzify** (centre of gravity):

$$z^* = \frac{\sum \mu(z)\, z}{\sum \mu(z)} = \frac{0.2(30) + 0.57(40+50+60) + 0.29(70+80+90+100)}{0.2 + 3(0.57) + 4(0.29)} = \frac{188.9}{3.06} \approx 61.8\,\%$$

> With fine sampling the centroid is 60.2 %: the 11-point sum is the usual exam approximation.

ANSWER

R3 fires at 0.57 (medium), R4 at 0.29 (high). Fan speed ≈ 61.8 % (11-point centroid; 60.2 % exact).

05 LIVE LAB 19–26 min

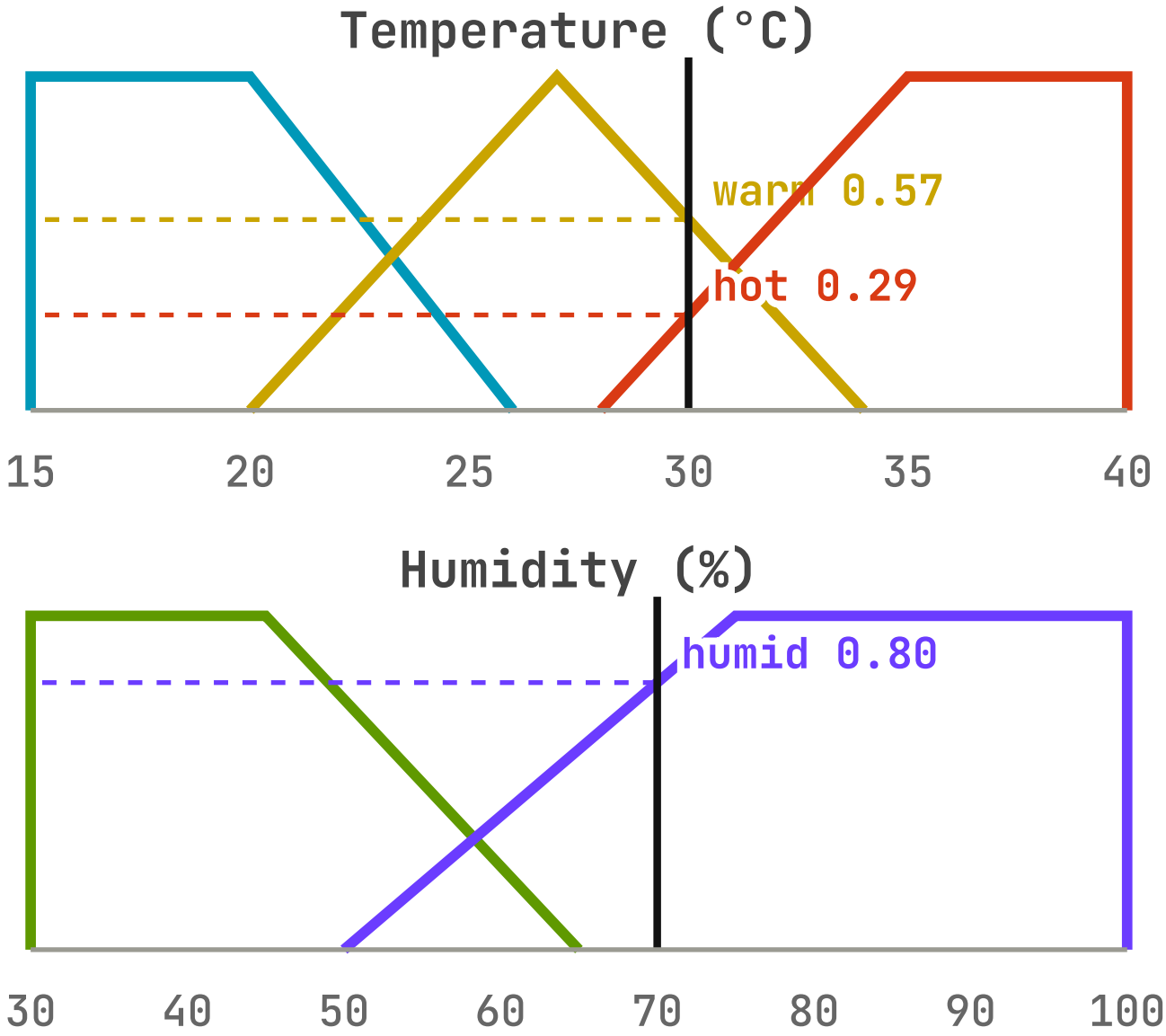
Live lab: the fan controller

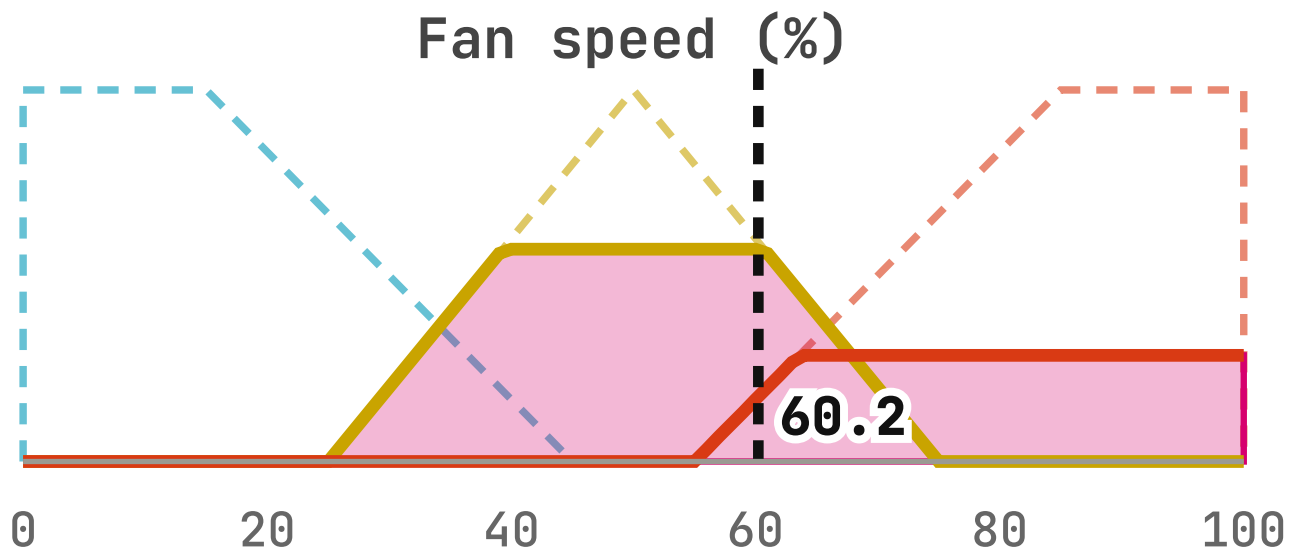
A fuzzy fan controller Move the temperature and humidity. Each rule fires as strongly as its weakest condition (AND = min); its output set is clipped at that height; the clipped sets are merged (max) and the centroid is the fan speed.

Temperature **30 °C**

Humidity **70 %**

FAN SPEED
60.2 %





Rule	IF	THEN fan	strength
R1	temp is **cool**	**low**	0.00
R2	temp is **warm** AND humidity is **dry**	**low**	0.00
R3	temp is **warm** AND humidity is **humid**	**medium**	0.57
R4	temp is **hot**	**high**	0.29

CENTROID

60.2 %

11-POINT SUM

61.8 %

SUGENO (20/50/90)

63.3 %

06 YOUR TURN 26–30 min

Your turn · 1: a Sugeno shortcut

1 EXAM CQ

Same sets and rules as the worked example, but each rule now outputs a single number (a **Sugeno** system): low = 20 %, medium = 50 %, high = 90 %.

Find the fan speed at **33 °C and 55 %** humidity. Check with the lab's Sugeno readout.

SOLUTION

1. Fuzzify:

$$\mu_{warm}(33) = \tfrac{1}{7} = 0.14, \quad \mu_{hot}(33) = \tfrac{5}{7} = 0.71, \quad \mu_{dry}(55) = \tfrac{10}{20} = 0.5, \quad \mu_{humid}(55) = \tfrac{5}{25} = 0.2$$

2. Rules: R1 = 0; R2 = min(0.14, 0.5) = 0.14 → 20; R3 = min(0.14, 0.2) = 0.14 → 50; R4 = 0.71 → 90.

3. Weighted average:

$$z = \frac{0.14(20) + 0.14(50) + 0.71(90)}{0.14 + 0.14 + 0.71} = \frac{74.3}{1.0} \approx 74.3\,\%$$

ANSWER

Rule strengths 0, 0.14, 0.14, 0.71; fan speed ≈ 74.3 %.

07 THINK DEEPER 30–33 min

Critical thinking · why not just use an equation?

THINK DEEPER

A fan speed of 61.8 % could also come from a formula, e.g. speed = 6(T − 20). What does fuzzy logic give you that the formula does not?

Readable knowledge. A fuzzy controller is an **expert system** (Week 2) whose rules can fire partly. You can explain every decision ("hot fired at 0.29"), tune it by moving a corner of a triangle, and add humidity without starting over. The cost: tuning by hand, no guarantees, and a rule explosion with many inputs.

A STRONG ANSWER MENTIONS

- The rules are in words an expert (or a customer) can read and check: "warm and humid → medium".
- Adding a factor (humidity) means adding rules, not re-deriving an equation.
- It interpolates smoothly between rules, with no thresholds that make the fan jump.
- But: membership shapes are guesses; there is no proof of optimality, and many inputs mean very many rules.

08 CONCEPT 33–38 min

Genetic algorithms: search by breeding

1. **Encode** a candidate solution as a string (a chromosome): bits, digits, a permutation.
2. Start with a random **population** of N individuals. Score each with a **fitness** function.
3. **Select** parents in proportion to fitness (roulette wheel) or by tournament.
4. **Crossover**: cut two parents at a random point and swap tails.
5. **Mutate**: change each gene with a small probability.
6. The children form the next generation (optionally keep the best: **elitism**). Repeat until good enough or out of time.

✦ WHERE IT FITS

A GA is a **local search** (Week 3) that keeps k states, like beam search, but makes new ones by *combining* two good states instead of only nudging one.

$$p_i = \frac{f_i}{\sum_j f_j}$$

Roulette selection: fitness must be positive; bigger slice, more children.

09 WORKED EXAMPLE 38–44 min

Worked example: one generation by hand

GIVEN

Maximise f(x) = x², x an integer 0–31 encoded in 5 bits. Population: 01101, 11000, 01000, 10011 (Goldberg's classic).

FIND

Selection probabilities, expected copies, the children of crossover, and the new total fitness.

Method: simple GA (Goldberg, 1989)

SOLUTION

1. Decode and score. Fitness f = x²:

$$01101 = 13 \to 169, \quad 11000 = 24 \to 576, \quad 01000 = 8 \to 64, \quad 10011 = 19 \to 361; \quad \sum f = 117($$

2. **Roulette selection**: $p_i = f_i / \Sigma f$. Expected copies = $N \cdot p_i$:

$$p = 0.14,\ 0.49,\ 0.05,\ 0.31 \qquad Np = 0.58,\ 1.97,\ 0.22,\ 1.23$$

3. Suppose the wheel picks 13, 24, 24, 19 (8 is lost). **Crossover** 0110|1 × 1100|0 and 11|000 × 10|011:

$$01100 = 12, \quad 11001 = 25, \quad 11011 = 27, \quad 10000 = 16$$

4. New fitness: 144 + 625 + 729 + 256 = **1754** (was 1170); best 729 (was 576). No mutation was needed this time.

> Selection copies good building blocks ("11…"); crossover combines them. The population improves on average, not every individual.

ANSWER

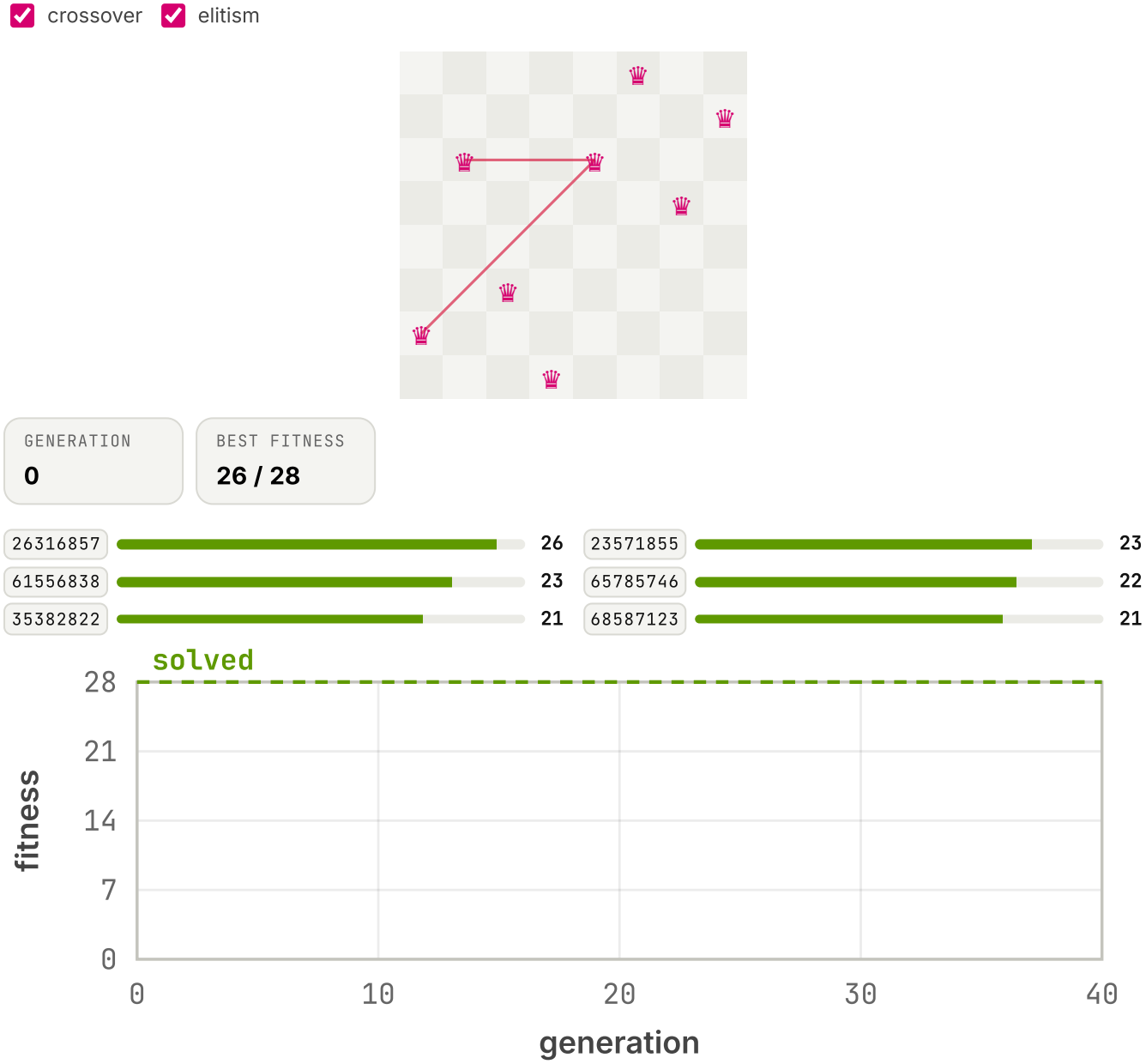
p = 0.14, 0.49, 0.05, 0.31. Children 12, 25, 27, 16; total fitness 1170 → 1754, best 576 → 729.

10 LIVE LAB 44–51 min

Live lab: evolve an 8-queens solution

A genetic algorithm solves 8-queens

Each individual is 8 digits: the row of the queen in each column. Fitness = pairs of queens that do not attack each other (28 = solved). Select in proportion to fitness, cross over, mutate, repeat.



11 YOUR TURN 51–55 min

Your turn · 2: Russell &

Norvig's 8-queens population

2 EXAM CQ

Four 8-queens individuals with fitness (non-attacking pairs): **24748552** (24), **32752411** (23), **24415124** (20), **32543213** (11).

(a) Selection probability of each? (b) Cross 32752411 with 24748552 after the 3rd digit: write both children.

SOLUTION

1. Total fitness 24 + 23 + 20 + 11 = 78:

$$p = \frac{24}{78}, \frac{23}{78}, \frac{20}{78}, \frac{11}{78} = 31\%,\ 29\%,\ 26\%,\ 14\%$$

2. Crossover after digit 3:

$$327|52411 \times 247|48552 \ \rightarrow\ 327\,48552 \text{ and } 247\,52411$$

3. Their fitness: **23** and **22** (count the attacking pairs). Neither child beats the best parent here, and that is normal: improvement is statistical.

ANSWER

p = 31 %, 29 %, 26 %, 14 %. Children 32748552 (fitness 23) and 24752411 (fitness 22).

---

12 THINK DEEPER 55–58 min

Critical thinking · when does crossover help?

THINK DEEPER

Crossover helped in x^2 and 8-queens. Can you design an encoding where it is useless, or harmful?

Crossover pays off when the encoding keeps **useful building blocks together**, so that halves of good parents are still good. For a TSP tour as a list of cities, a naive cut-and-swap makes invalid tours. Choosing the encoding is most of the work in a GA.

A STRONG ANSWER MENTIONS

- Crossover helps when good solutions share good blocks in fixed positions (schemas): "11…" in x^2; a good left half of a board.
- For a permutation (a TSP tour), one-point crossover produces children that visit some cities twice: special operators are needed (order crossover).
- If neighbouring genes do not interact meaningfully, crossover is just a big random mutation.

13 REAL WORLD 58–62 min

Real world: rice cookers and space antennas

FUZZY AT HOME

Fuzzy-logic rice cookers, washing machines (load and dirt sensors → water and spin time) and inverter air conditioners sold in Dhaka markets all run small fuzzy controllers. The Sendai subway in Japan (1987) used fuzzy control for smoother, more energy-efficient braking.

BME CORNER

Fuzzy controllers are studied for anaesthesia depth and insulin dosing, where expert rules ("glucose high and rising → increase a little") are natural, and smooth action is safer than thresholds.

AN EVOLVED ANTENNA

NASA's ST5 satellites (2006) flew an X-band antenna whose odd, bent-wire shape was designed by an evolutionary algorithm; it met the specification better than the hand design.

THINK DEEPER

Your university class routine: rooms, teachers, sections, no clashes. Why is a GA a reasonable tool?

Because the space is enormous and discrete, but **scoring** a timetable is easy (count clashes). A GA, or simulated annealing from Week 3, finds a clash-free routine without anyone proving it is the best one.

A STRONG ANSWER MENTIONS

- Huge discrete search space, no gradient.
- Fitness is easy to score: count clashes and penalise bad gaps.
- Any good-enough timetable is acceptable: no need for the proven optimum.

14 POP QUIZ 62–66 min

Pop quiz

1. **μ_{tall}(Rafi) = 0.7. Then $\mu_{not\ tall}$(Rafi) and $\mu_{tall\ AND\ not\ tall}$(Rafi) are…**

   A. 0.3 and 0

   B. 0.3 and 0.3 ✓

   C. 0.7 and 0.7

   D. 0.3 and 0.21

   NOT = 1 − 0.7 = 0.3; AND = min(0.7, 0.3) = 0.3. In fuzzy logic "A and not A" need not be 0: the law of contradiction fails.

2. **In Mamdani inference, a rule "IF x is A AND y is B THEN z is C" with $\mu_A = 0.6$, $\mu_B = 0.9$ ...**

   A. adds 1.5 to C

   B. clips C at 0.6 ✓

   C. clips C at 0.9

   D. multiplies C by 0.54

   AND = min(0.6, 0.9) = 0.6; the output set C is cut at that height. (Product inference would scale it by 0.54 instead.)

3. **A Sugeno system has rule strengths 0.2 → 30 and 0.6 → 80. Its output is...**

   A. 55

   B. 67.5 ✓

   C. 110

   D. 62

   Weighted average: (0.2·30 + 0.6·80)/(0.2 + 0.6) = 54/0.8 = 67.5.

4. **In roulette-wheel selection, an individual with fitness 30 in a population whose fitness sums to 120 is chosen with probability...**

   A. 0.30

   B. 0.25 ✓

   C. 0.12

   D. 4

   30/120 = 0.25 per spin. With N = 4 spins, it expects one copy.

5. **Mutation in a GA is mainly there to...**

   A. speed up convergence

   B. keep diversity and reach values no parent has ✓

   C. replace crossover

   D. compute fitness

   Crossover can only recombine what is already in the population. If every individual has a 0 in position 3, only mutation can bring back a 1.

---

15 RECAP 66–69 min

Three things to remember

01

Fuzzy sets: $\mu \in [0, 1]$, overlapping linguistic values. AND = min, OR = max, NOT = 1 − μ; very = μ^2.

02

Mamdani: fuzzify → rule strength (min) → clip → aggregate (max) → centroid $\Sigma\mu z/\Sigma\mu$. Sugeno: $\Sigma wz/\Sigma w$.

03

GA: population, fitness, roulette selection f/Σf, crossover, mutation, elitism. A k-state local search that recombines.

---

16 RECAP 69–80 min

Midterm: what to expect

WEEK	BE ABLE TO…
1	define AI (four approaches), write PEAS, classify environments
2	formulate a problem (states, actions, goal, cost); propositional logic, forward/backward chaining, expert-system architecture
3	trace BFS, DFS, UCS, DLS, IDS; compare them; hill climbing, annealing acceptance $e^{\Delta E/T}$, beam search
4	greedy and A* traces; admissible vs consistent; dominance, b*, relaxed problems; minimax
5	alpha–beta cuts, move ordering, expectiminimax; fuzzy inference by hand; one GA generation

Creative questions (CQ) only, no MCQ. 30 % of the course. Each CQ has a short scenario and 3–4 parts that go from recall to analysis.

CQ EXAM CQ

A practice CQ

A ward robot moves on: S–A (2), S–B (5), A–B (2), A–G (6), B–G (2). Heuristic: h(S) = 5, h(A) = 4, h(B) = 2, h(G) = 0.

(a) Is h admissible? (b) Consistent? (c) Trace A∗: expansion order, path, cost. (d) What would greedy best-first return?

SOLUTION

1. (a) True costs: h*(B) = 2, h*(A) = min(6, 2 + 2) = 4, h*(S) = min(2 + 4, 5 + 2) = 6. h ≤ h* everywhere (5 ≤ 6, 4 ≤ 4, 2 ≤ 2): **admissible**.
2. (b) Every edge: 5 ≤ 2 + 4, 5 ≤ 5 + 2, 4 ≤ 2 + 2, 4 ≤ 6 + 0, 2 ≤ 2 + 0: **consistent**.
3. (c) S (f 5) → A (f 6), B (f 7); expand A → B improves to g 4, f 6; G g 8. Expand B → G improves to g 6. Expand G (f 6). Order **S, A, B, G**; path S–A–B–G, cost **6**.
4. (d) Greedy picks the smaller h: from S it goes to B (h 2), then G: path S–B–G, cost **7**, not optimal.

ANSWER

(a) Admissible, (b) consistent, (c) S, A, B, G; path S–A–B–G, cost 6. (d) Greedy: S–B–G, cost 7.

S. M. Sakeef Sani

smsakeefsani.hbitcorp.com

Adjunct Lecturer, CSE · Northern University Bangladesh