Matthew Ragoza
mtr22@pitt.edu

Assignment 3

ISSP 2160
September 29, 2021

Problem 1. Constraint satisfaction

Inference with forward checking

The values inferred by forward checking are $z = 5$ and $v = 0$. This can be seen as follows:

$x = 2$	1. Given
$z \bmod 3 = x \bmod 3$	2. Constraint
$z \bmod 3 = 2$	3. Substitute and simplify
$z \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$	4. Given
$z \in \{2, 5, 8\}$	5. Infer disequations
$y = 0$	6. Given
$z \bmod 5 = y \bmod 5$	7. Constraint
$z \bmod 5 = 0$	8. Substitute and simplify
$z \in \{5\}$	9. Infer disequations
$z = 5$	10. Infer by exhaustion of alternatives
$w \bmod 4 = z \bmod 4$	11. Constraint
$w \bmod 4 = 1$	12. Substitute and simplify
$w \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$	13. Given
$w \in \{1, 5, 9\}$	14. Infer disequations
$t = 0$	15. Given
$v \bmod 12 = t \bmod 12$	16. Constraint
$v \bmod 12 = 0$	17. Substitute and simplify
$v \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$	18. Given
$v \in \{0\}$	19. Infer disequations
$v = 0$	20. Infer by exhaustion of alternatives
$u \bmod 3 = v \bmod 3$	21. Constraint
$u \bmod 3 = 0$	22. Substitute and simplify
$u \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$	23. Given
$u \in \{0, 3, 6, 9\}$	24. Infer disequations

Inference with arc consistency

The values inferred by arc consistency are $z = 5$, $v = 0$, and $w = 9$. Proceeding from the same inferences made for forward checking, the following holds:

$w \in \{1, 5, 9\}$	25. From forward checking
$u \in \{0, 3, 6, 9\}$	26. From forward checking
$u \bmod 3 = w \bmod 3$	27. Constraint
$w = 1$	28. Assumption
$u \bmod 3 = 1$	29. Substitute and simplify
$u \in \emptyset$	30. Infer disequations
$w \neq 1$	31. Infer by contradiction
$w = 5$	32. Assumption
$u \bmod 3 = 2$	33. Substitute and simplify
$u \in \emptyset$	34. Infer disequations
$w \neq 5$	35. Infer by contradiction
$w = 9$	32. Infer by exhaustion of alternatives
$u \bmod 3 = 0$	33. Substitute and simplify
$u \in \{0, 3, 6, 9\}$	34. Infer disequations

Problem 2. Traveling salesman problem

a. I implemented simulated annealing with a linear cooling schedule and the given default parameters in `SA.py`. This baseline algorithm is labelled as SA_a in Figures 1, 2 and 3.

b. To experiment with simulated annealing parameters, I performed a grid search over ranges of values for the number of simulation steps and the initial temperature. I ran 100 trials with each parameter setting and selected the best parameters according to the mean performance across the trials. The performance metric I used was the lowest tour distance discovered in a given trial (Best distance). The best-performing algorithm from this experiment is labelled as SA_b in the results. As seen in Figure 2, I was able to discover optimized parameters that decrease the best tour distance from 88.49 to 63.67. This was accomplished by reducing the initial temperature from 100 to 10 and running 10 times more simulation steps, as shown in Figure 1. The best tours found using the initial (SA_a) and optimized (SA_b) parameters are listed in Figure 3.

c. To further improve upon my simulated annealing algorithm for the competition, I considered several different cooling schedules, depicted in Figure 4. I performed another grid search, this time over different cooling schedules, and over a refined range of initial temperatures based on insight from the previous grid search. However, this time I constrained the number of simulation steps to 20,000 per the competition rules. Again, I selected the best parameters according to the mean best-tour distance across 100 trials, and I refer to this algorithm as SA_c going forward. The best tour distance produced by SA_c was 66.32, which is just slightly worse than than SA_b, as seen in Figure 2. However, the new algorithm is far more efficient given that it found a comparable tour after trying only 20,000 tours compared to the 1 million tried by SA_b. Its performance is even more impressive given that it started from a worse initial tour than SA_b and managed to nearly close the gap. In terms of the difference in parameters of SA_c, the initial temperature was further reduce from 10 to 1, and the linear cooling schedule was replaced with a power cooling schedule, which is defined as follows:

$$T_{i+1} = T_i b^{\frac{i}{k}}$$

The variable b is a parameter specifying the rate of exponential decay in temperature, where a lower value indicates faster decay. For the optimized SA_c algorithm, $b = 0.1$. Taken in combination with its far lower initial temperature compared to the baseline of SA_a, the final algorithm had much lower temperature for the entire simulation, and its temperature decayed more quickly than with the linear cooling schedule. This translated into lower probability of accepting worse tours, leading to a greedier algorithm. This would tend to lead to local optima, and there is some evidence that this happened based on the fact that SA_b reached a slightly better tour distance. The SA_b algorithm had much higher temperature for much longer, which would facilitate escaping local optima. With a tight constraint on the simulation length, however, the optimzation of SA_c's parameters biased it to prefer reaching reasonably good optima very quickly rather then wasting time trying many bad configurations in hopes of finding the global optimum.

Algorithm	# simulation steps	Initial temperature	Cooling function
SA$_a$	100,000	100	Linear
SA$_b$	1,000,000	10	Linear
SA$_c$	20,000	1	Power

Figure 1: Simulated annealing parameters for the traveling salesman problem.

Algorithm	Initial distance	Best distance	# tours tried	# tours accepted
SA$_a$	283.73	88.49	100,000	92,195
SA$_b$	296.69	63.67	1,000,000	592,074
SA$_c$	315.15	66.32	20,000	1,506

Figure 2: Simulated annealing statistics for the traveling salesman problem. The statistics shown are for the best tour discovered in 100 trials for each parameter set.

Algorithm	Best tour
SA_a	0, 16, 39, 45, 46, 14, 9, 2, 55, 48, 40, 43, 56, 25, 29, 47, 7, 27, 1, 34, 18, 36, 3, 11, 28, 54, 51, 57, 23, 41, 6, 58, 44, 20, 4, 21, 10, 13, 31, 22, 19, 49, 8, 35, 59, 30, 38, 52, 12, 37, 26, 24, 42, 5, 53, 50, 15, 33, 17, 32
SA_b.	0, 16, 14, 30, 42, 24, 26, 38, 37, 12, 52, 46, 9, 2, 55, 20, 4, 40, 59, 48, 6, 10, 35, 21, 22, 19, 8, 49, 31, 13, 58, 41, 54, 44, 43, 28, 11, 57, 3, 56, 25, 23, 51, 47, 29, 7, 50, 1, 27, 15, 34, 33, 18, 36, 17, 45, 39, 32, 53, 5
SA_c	0, 42, 30, 16, 14, 24, 26, 38, 37, 12, 52, 46, 2, 9, 55, 20, 4, 40, 48, 41, 58, 13, 31, 49, 8, 19, 22, 21, 10, 6, 35, 59, 36, 18, 33, 34, 15, 29, 25, 23, 56, 3, 44, 54, 43, 28, 57, 11, 51, 47, 7, 50, 1, 27, 17, 45, 39, 32, 5, 53

Figure 3: Best tours for the traveling salesman problem discovered by simulated annealing with each parameter set.

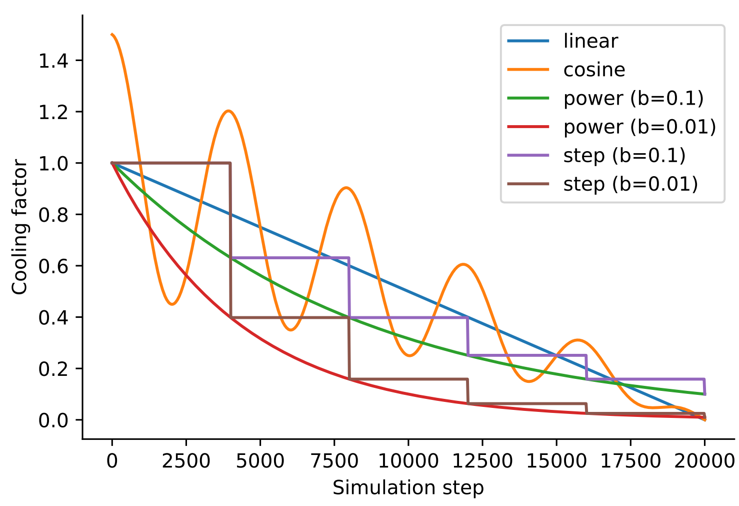

Figure 4: Visualization of the different cooling schedules considered for the competition. The value of the cooling factor is multiplied by the initial temperature to get the current temperature at a given simulation step, and each schedule is scaled horizontally based on the number of simulation steps parameter.