

In the first part you've evolved small logic circuits: adders and logic gates. However even for small circuits the problem can get quite complex and GA may take a lot of generations to find a configuration fulfilling the specifications.

In this second part you will explore two possible ways to speed up evolution. The first one consists in evolving a smaller PAL circuit for each of the output bit, such as to have a building block for each bit. The full circuit is the combination of those building blocks. The second method consists in performing incremental evolution using "don't care" bits. It is bits which are not used in the fitness computation. You will evolve the PAL such as to conform to a subset of the truth table, then progressively the subset of the truth table will be expanded (by removing don't care bits) until the circuit conforms to the whole truth table.

Exercise 1. Incremental evolution of a 2x2 bit adder with "building blocks"

Try to evolve the same circuit incrementally: evolve three different circuits (4 inputs, 1 output), one for each of the output bit of the adder (i.e. one circuit for bit 0, another one for bit 1,). You can load the truth table from the files `add2_2_i?.tbl`. Is the evolution faster? Are some bits more difficult to evolve than other? Why? Compare with the direct evolution (4 inputs, 3 outputs)

Exercise 2. Incremental evolution of a 2x2 bit adder with "don't care" bits

Try to evolve the 2x2 bit adder (4 inputs, 3 outputs) by using don't care bits. Don't care bits are bits that are not used to compute the fitness. To do so, place "X" in the first two column of the truth table, then evolve the circuit and stop as soon as the maximum fitness is reached. Modify the truth table to have don't care bits only in the first column now, and continue evolution, and eventually remove the last column of don't care bits to evolve the complete circuit. What kind of results do you get? Is there a similarity with the incremental evolution of exercise 1? If yes which one and why?

Exercise 3. Evolve a 3x3 bit adder

Try to evolve the 3x3 bit adder (6 inputs, 4 outputs). You can load the file `add3_3.tbl`. Do you succeed at evolving this circuit?... Compare the direct evolution of the 4-outputs circuit to incremental evolution with building blocks and don't care bits. Discuss the results.