Metaheuristic Optimization: Binary Representation

AE 6310: Optimization for the Design of Engineered Systems

Spring 2017

Dr. Glenn Lightsey

Lecture Notes Developed By Dr. Brian German

Binary Representation

Some metaheuristic methods, e.g. genetic algorithms, require discrete representation of design variables.

Often, this discretization is accomplished by discretizing continuous design variables into binary strings of 0s and 1s.

Binary Representation

To define the discretization, we must specify:

- A "range" defined by upper and lower bounds on the design variables
- A "resolution"—the desired size for each discretized step

The number of binary bits, N, required to discretize a continuous variable is:

$$N \geq \frac{\ln\left(\frac{Range}{Resolution} + 1\right)}{\ln(2)}$$

Binary Representation

For example, representing a variable $10 \leq x_1 \leq 20$ with a resolution of 1 implies,

$$Range = 20 - 10 = 10$$

$$N \geq \frac{\ln\left(\frac{Range}{Resolution} + 1\right)}{\ln(2)} = \frac{\ln\left(\frac{10}{1} + 1\right)}{\ln(2)} = 3.457$$

which means we need at least 4 binary numbers ("bits"). The resulting binary mapping is:

'0000'=10, '0001'=11, '0010'=12, '0011'=13, '0100'=14, '0101'=15, '0110'=16, '0111'=17, '1000'=18, '1001'=19, '1010'=20

Binary Representation

To convert a base-10 number to a base-2 (binary) number, one approach is the “repeated division by two” method:

1. Divide the number by 2 and keep the quotient in dividend/remainder form
2. Divide the resulting quotient by 2 and keep the result in dividend/remainder form
3. Repeat step 2 until the last dividend is 0

Keep track of the remainders from each step in this process and arrange them in reverse order to form the binary number.

Binary Representation

Example: Convert 126 to binary.

Dividend	Remainder
126	
63	0
31	1
15	1
7	1
3	1
1	1
0	1

The final binary string is therefore 1111110.

Binary Representation

To convert a binary number with n bits to a base-10 number, apply the following formula,

$$\sum_{i=1}^{n} b_i 2^{i-1}$$

where b_i is the value of the binary digit i (0 or 1).

Concatenating Binary Numbers

We can represent a design as a concatenation of the binary numbers for each design variable into a single binary string.

For example, consider two design variables, x_1 & x_2, each ranging from 0 to 10 with resolution 1 (requiring 4 bits each).

A particular occurrence is $x_1 = 1 \rightarrow 0001$ and $x_2 = 4 \rightarrow 0100$.

The vector of design variables can be represented as a single string in the form,

$$\overbrace{0001}^{x_1}\overbrace{0100}^{x_2}$$

Concatenating Binary Numbers

The approach of concatenating the variables into a single string makes it easy to slightly modify a design by randomly changing a single bit.

For our example from the previous slide:

If '00010100' is modified to '00010101'…

…the design variables change from [1,4] to [1,5].

Hamming Distance

In the previous example, flipping one bit changed the design to an immediately adjacent design (in terms of the base 10 design variables).

However, flipping another bit may result in a move to a non-adjacent design.

For our example, if '00010100' is modified to '00010110' the design changes from [1,4] to [1,6].

The Hamming distance is a measure of how many bits must be flipped to move between two different designs.

Hamming Distance & Hamming Cliff

For a binary representation of 0 to 10 with resolution 1:

- The Hamming distance between 0 (‘0000’) and 1 (‘0001’) is 1
- The Hamming distance between 1 (‘0001’) and 2 (‘0010’) is 2
- The Hamming distance between 7 (‘0111’) and 8 (‘1000’) is 4

The **“Hamming cliff”** refers to the situation in which more than one bit flip is required to move from one number to its immediate “neighbor.”

In the above examples, the moves from 1 to 2 and from 7 to 8 experience the Hamming cliff.

Hamming Distance & Hamming Cliff

Consider the process of randomly changing each bit to modify a design in our 4-bit example. If the probability of any one bit flipping is 10%, then the probability of moving from:

0 to 1 is .9*.9*.9*.1 = 7.3%

1 to 2 is .9*.9*.1*.1 = 0.81%

7 to 8 is .1*.1*.1*.1 = 0.01%

Ideally, the probability of moving between any 2 adjacent designs should be the same. Also, we want the greatest probability of the design moving to its immediately adjacent neighbor.

Gray Code

Gray code is a binary representation in which the Hamming distance between two neighboring designs is 1.

Gray coding avoids the Hamming cliff problem for neighboring points.

Gray Code

To convert from binary code to Gray code:

Retain the first (left-most) bit as the first bit in the Gray code

For $i = 1$ to (Number of bits – 1)

Perform an "exclusive OR" (XOR) operation on the i and $(i + 1)$ bits of the binary string. If the bits are the same, the $(i + 1)$ bit should be set to 0. If they are different, the $(i + 1)$ bit should be set to 1.

End For

Gray Code

Example: Convert the number ‘1001’ from binary to gray code:

- Retain first bit of ‘1001’: ‘1_ _ _’
- Compare 1st and 2nd bits in ‘1001’: ‘11_ _’
- Compare the 2nd and 3rd bits in ‘1001’: ‘110_’
- Compare the 3rd and 4th bits in ‘1001’: ‘1101’

Gray Code

Base 10 Number	Gray Code	Binary Code
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

Gray Code

To convert from Gray code to binary code:

Retain the first (left-most) bit as the first bit in the binary string

For $i = 1$ to (Number of bits – 1)

Perform an XOR operation on the i^{th} bit of the binary string that we just set and the $(i+1)^{th}$ bit of the Gray string. If the bits are the same, the $(i+1)$ bit should be set to 0. If they are different, the $(i+1)$ bit should be set to 1.

End For

Gray Code

Example: Convert the number '1101' from gray code to binary code:

- Retain first bit of '1101': '1_ _ _'
- Compare 1st bit of the binary string '1_ _ _' with the 2nd bit in the Gray string '1101': '10_ _'
- Compare the 2nd bit of the binary string '10_ _' and 3rd bit in the Gray string '1101': '100_'
- Compare the 3rd bit of the binary string '100 _' and 4th bit in the Gray string '1101': '1001'