A formal description of the components of the genetic algorithm follows:

(1) Recall that the three decision parameters in our model control the maximum stocking rate, z_{max}, the level of shrub biomass above which destocking starts, q_z, and the degree to which fire is suppressed, q_d. Thus in our case, the relevant candidate solution is the vector (z_{max}, q_z, q_d). These candidate solutions will be represented by a binary bit string of fixed length L, which is denoted as α, and $\alpha \in \{0,1\}^L$. The bit string can be separated into *n* segments of equal length l_x (thus $L=n \cdot l_x$). Each segment is interpreted as the binary code of the object variable $x_i \in [x^l_i, x^u_i]$, where x^l_i is the lower bound of the objective variable x_i and x^u_i the upper bound, which can be decoded by applying the following rule

$$\Gamma_i(a_{i1}..a_{il_z}) = x_i^1 + \frac{x_i^u - x_i^1}{2^{lx}-1} \cdot \left(\sum_{j=0}^{lx-1} a_{i,lx-j} \cdot 2^j \right) \qquad (1)$$

where ($\alpha_{i1,..,}\alpha_{ilx}$) denotes the i-th segment of an individual $\alpha \in (0,1)^L$. For example, if we wish to represent a vector of three real numbers, each with a 10-digit binary string, then our “candidate solution” a is a 30-digit long string of zeros and ones, i.e. $a \in \{0,1\}^{30}$, with n=3m and l_x =10. The first ten element of the string correspond to the binary representation for z_{max}, i.e. $z_{max}=\Gamma_1 (a_{1,1},\ldots,a_{1,10})$, the second ten to q_z, and the last ten to q_d. Then $\Gamma_1 x \ldots x \Gamma_n$ yields a vector of real values, each in the desired range [x^l_i ,x^u_i]. For example, suppose we wish to use a 10-digit binary bit string to represent a number between 0 and 1, i.e. x^l=0 and x^u=1. Using equation (1) the string [1001100001], can be translated into a real value:

$$\Gamma = 0 + \left(\frac{1-0}{2^{10}-1} \right) (1 \cdot 2^9 + 0 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0) = 0.595$$

(2) Mutation is a bit reversal event that occurs with small probability, p_m, per bit. This mutation can explore new genetic information and is a powerful operator in creating new candidate solutions.

Example: Consider the following bit string; 1111111111. At random, roughly one in every 1000 symbols flips from 0 to 1 or vice versa; e.g. 1111111111 -> 1101111111.

(3) The algorithm uses a cross over operator that exchanges substrings arbitrarily between two individuals with probability p_c. Length and position of these substrings are chosen at random, but are identical for both individuals.

Example: Consider the following bit strings: 1111111111 and 0000000000. A point along the strings is selected at random, for example between the fifth and the sixth bit, and the offspring contains a mixture of the parents: 1111100000 and 0000011111.

(4) The probabilistic selection operator forms the next generation by copying individuals on the basis of fitness-proportionate probabilities p_i. In other words the new generation of candidate solutions are generated by using a kind of roulette wheel with slots sized according to the value of the objective function:

$$p_i = \frac{F(\alpha_i)}{\sum_{j=1}^{N} F(\alpha_j)}$$

where $F:\{0,1\}^L \rightarrow \Re$ is the fitness function. In an optimization problem the fitness function is equal to the objective function of a maximization problem. In case of a minimization problem, the objective function is multiplied with “-1” to derive a maximization problem. Candidate solutions that lead to lower values of the objective function are less likely to reproduce their “genetic” information.