1. **Population**: In genetic algorithms, a population is a set of potential solutions to the problem at hand. Each solution, often referred to as an individual or a chromosome, is typically encoded as a string of characters or numbers.

2. **Chromosome**: A chromosome is a single solution represented within the population. It is typically structured as a string of genes, which collectively represent a candidate solution to the problem being solved.

3. **Gene**: A gene is a part of a chromosome and represents one aspect of the solution. For example, in a routing problem, a gene might represent a waypoint or destination.

4. **Fitness Function**: This is a function that evaluates how good a solution is in solving the problem. The fitness function assigns a fitness score to each individual in the population, and this score determines the likelihood that an individual will be selected to reproduce.

5. **Selection**: This process involves selecting individuals from the current population to create the next generation. Selection is typically based on fitness, with higher fitness individuals being more likely to be chosen. Common methods include roulette wheel selection, tournament selection, and rank selection.

6. **Crossover (Recombination)**: Crossover is a genetic operator used to vary the programming of chromosomes from one generation to the next. It is analogous to reproduction and biological crossover. During crossover, parts of two parent chromosomes are combined to produce one or more offspring. Common types of crossover include single-point, two-point, and uniform crossover.

7. **Mutation**: Mutation is another genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm solutions to the next. It is analogous to biological mutation. The mutation operator makes small random changes in the individuals to explore new regions of the solution space, which helps prevent the algorithm from being trapped in local optima.

8. **Elitism**: This is a method of ensuring that the best solutions from the current generation are carried over to the next generation. By preserving a few of the top-ranked individuals, elitism prevents loss of the best-found solutions due to the stochastic nature of selection and genetic operators.

9. **Termination Condition**: This defines when the genetic algorithm should stop. Common termination conditions include a maximum number of generations, a threshold fitness level that has been reached, or if the population has converged to a similar state where further generations do not produce significantly different results.
10. **Generational Model**: In this model, a new generation is formed by completely replacing the old generation with the new one after applying genetic operators. This is the traditional approach in genetic algorithms.

11. **Steady-State Model**: Unlike the generational model, the steady-state model introduces new individuals into the population while only a few of the worst individuals are replaced. This allows for a more gradual change in the population over time.