Step by step:

1. In the first step, we call the function that draws the initial pool in the number of initial_pop line-ups (the default for this function is 100 line-ups in the initial pool). The function initialising the initial pool of compositions creates 100 individual compositions in such a way that it calls 100-generating individual line-ups, which are created by selecting individual random players for a given position GK, LB, RB, CDM, CM, CAM, LW, RW, ST whose position in the team (club_position variable) corresponds to the given position, and similarly selecting 2 players for the CB position. Individual players in a given squad are drawn from a set of players with players only of a given corresponding position. For the second draw of a CB central defender for an individual line-up, we draw from the set of central defenders without the central defender drawn in the first iteration to avoid that we select the same player for an individual line-up twice. The individual line-up in the initial pool is therefore made up of individual players in the positions of GK, LB, RB, CDM, CM, CAM, LW, RW, ST and 2 different players for the CB position.
2. In the next step, the selection of the first parent pool takes place via a selection operator. The selection operator used in this case is roulette wheel selection. The parent pool is therefore selected in such a way that the selection of a random line-up from the line-up pool is done with a probability equal to the relative score for the individual line-up - the score for the individual line-up divided by the score for all line-ups in the pool. The score for an individual squad is calculated as the sum of the skills of all players (overall variable) for cases where the sum of the players' salaries does not exceed a given budget. For total salaries greater than the maximum budget, the score is equal to 0.
3. In the next step, we apply the crossover operator, i.e. the creation of offsprings (children) using the characteristics of the line-ups from the parent pool - the parents. The crossover operator used is one-point crossover. In the defined function, the crossover_point is selectable and the default is set to 5, i.e. the crossover proceeds in such a way that one offspring takes a line-up with 5 players from the first parent (positions: GK, LB, CB, CB, RB) and 6 more players from the second parent (remaining positions), and the second offspring inversely takes 5 from the second parent (positions: GK, LB, CB, CB, RB) and 6 from the first parent (remaining positions). In the implemented crossover, there is no problem with duplication of players in given positions - there is no situation that a line-up with, for example, two goalkeepers is formed.
4. In the next step, we apply the mutation operator. The implemented mutation operator changes with the given selectable mutation_rate (default set to 0.15) a single player from the given line-up from the pool of children - offspring - to another player of the same position from the set of all players of the given position. By changing to a player of the same position, there is no problem of duplication of players in given positions - there is no situation where a squad with, for example, two goalkeepers is created. After performing the initial step and going through all the operators of the GA algorithm, from the pool of line-ups created as a result of the first iteration of the algorithm we again run the algorithm of selection, crossover and mutation, obtaining another pool for which we again apply the above operators and repeat these steps given n number of iterations. For the GA() function, the number of iterations is selectable (default set to 99). The function finally returns the best composition from the pool after a given number of iterations.
5. In the final step, the algorithm returns the best line-up in the form of names, positions and overall of the players concerned.