These features favor the GA in cases where the traditional optimization approaches fail. The algorithm for a two player Nash equilibrium game (D'Amato et al. 2012a, b) is here described for simplicity.

Let U, V be players' strategy sets (both are metric spaces). Let f_1, f_2 be two real valued functions defined on $U \times V$ representing the players' objective functions. The used algorithm is based on the Nash adjustment process (Fudenberg and Tirole 1991), where players take turns setting their outputs, and the chosen output of each player (in U) is his best response to the output previously chosen by his opponents (in V). The converged steady state value of this process is a Nash equilibrium of the game. Let $s = u, v$ be the pair representing the potential solution for a 2 person Nash problem. Then u denotes the subset of variables handled by the player 1, belonging to U, and optimized under the objective function f_1. Similarly, v indicates the subset of variables handled by the player 2, belonging to V, and optimized under a different objective function f_2. As stated in the Nash equilibrium definition (Nash 1951), the player 1 optimizes pair s with respect to f_1 by modifying u while v is fixed by the player 2; symmetrically, the player 2 optimizes pair s w.r.t. the f_2 by modifying v, while u is fixed by the player 1. This procedure can be implemented numerically considering u^{k-1} and v^{k-1} be the best values found by players 1 and 2, respectively, at step (or generation) $k-1$. At next step, k, the player 1 optimizes u^k using v^{k-1} to evaluate the pair $\mathbf{s} = u^k, v^{k-1}$. At the same time, the player 2 optimizes v^k using u^{k-1} to evaluate the pair $\mathbf{s} = u^{k-1}, v^k$. The algorithm is structured in several phases, see also Fig. 1:

1. Generation of two different random populations, one for each player, at the first step. Player 1's optimization task is performed by acting on the first population and *vice versa*.
2. The sorting of the individuals among their respective population, is based on the evaluation of a fitness function typical of GAs. The results of the matches between each individual of population 1 with all individuals of population 2 (scoring 1 or −1, respectively, for a win or loss, and 0 for a draw) are stored, see Eq. (2).

$$\begin{cases} \text{if } f_1\left(u_i^k, v^{k-1}\right) > f_1\left(u^{k-1}, v_i^k\right), \text{fitness}_1 = 1 \\ \text{if } f_1\left(u_i^k, v^{k-1}\right) < f_1\left(u^{k-1}, v_i^k\right), \text{fitness}_1 = -1 \\ \text{if } f_1\left(u_i^k, v^{k-1}\right) = f_1\left(u^{k-1}, v_i^k\right), \text{fitness}_1 = 0 \end{cases} \tag{2}$$

A similar procedure is need for the player 2, as expressed in Eq. (3).

$$\begin{cases} \text{if } f_2\left(u_i^k, v^{k-1}\right) > f_2\left(u^{k-1}, v_i^k\right), \text{fitness}_2 = 1 \\ \text{if } f_2\left(u_i^k, v^{k-1}\right) < f_2\left(u^{k-1}, v_i^k\right), \text{fitness}_2 = -1 \\ \text{if } f_2\left(u_i^k, v^{k-1}\right) = f_2\left(u^{k-1}, v_i^k\right), \text{fitness}_2 = 0 \end{cases} \tag{3}$$

The individuals having an equal fitness value are sorted by f_1 for player 1 and f_2 for player 2.