1 Fitness studied

In our previous works, we evaluated a single bot of the population versus always the same bot (our reference-bot). For fitness function, the bot was evaluated several times (in different maps). The fitness function was defined depending of the result of the battle (if the bot win all his battles or lose in someone) and the numbers of turns needed for end the game. For two bots of the population (A and B) the fitness is defined like:

$A, B \in Population$
if A WINs always **then**
    if B LOSEs some battle **then**
        A is better than B
    else if A take less turns than B **then**
        A is better than B
    else
        B is better than A
    end if
else
    if B WINs always **then**
        B is better than A
    else if A take less turns than B **then**
        B is better than A
    else
        A is better than B
    end if
end if

Fig. 1. Fitness used in battles of 2 bots

This fitness works well for battles between two bots, and our first fitness proposed is an natural evolution of this fitness for 4 bots battles.

1.1 Fitness based in Position-turns

This fitness it's an natural evolution of the previous fitness, applied to 4 bots battles. Again, the evaluations are in several maps. In this case, we studie the position (1th,2th,..) of the bot in the 4-battle and the number of turns. For a bot that wins all the battles (it's 1th in all the battles) we call the sum of the numbers of turns $ferocity$, and in previous works we found that a bot that wins in less turns it's best that other that takes more turns (even if the both wins). In other case, the sum of turns is call *sturdy* and oposite to the $ferocity$, it's desirable a bot that take more turns in be defeated.

The principal problem in the previous fitness, it's that there we are using two independents variables. In this case, we try to found a fitness that resume the goodness of our bot in a single number.

In this fitness, we are only interesting in the final result: position and turn. We aren't student how the bot reach it. In the nexts fitness, we use another

$A, B \in Population$
if A WINs always **then**
  if B LOSEs some battle **then**
    A is better than B
  else if A take less turns than B **then**
    A is better than B
  else
    B is better than A
  end if
else
  if B WINs always **then**
    B is better than A
  else if A take less turns than B **then**
    B is better than A
  else
    A is better than B
  end if
end if

Fig. 2. Fitness used in battles of 2 bots

metric to define the goodness of the bots. The percentage of the ships that in each turn belong to each player.

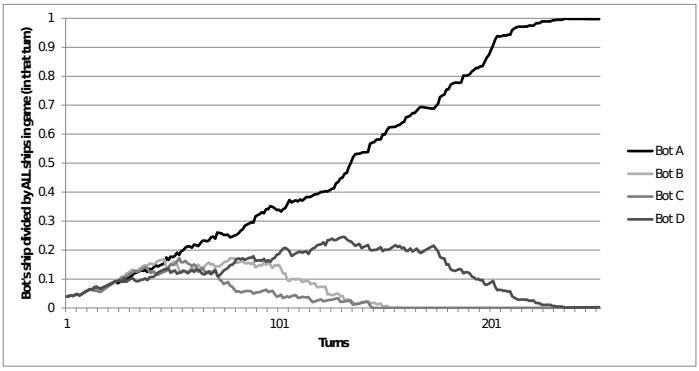

Fig. 3. Representation of the number if ships of each bot in each turn

1.2 Fitness based in Slope

For this fitness, we use the leats squares regression analysis for resume the cloud of points to a simple rect. The rect is represented as $y = \alpha \times x + \beta$ where, α and β are calculated as 4:

For our fitness, we take the slope of the rect: α. We can see if the bot win, $\alpha > 0$ and if loses, $\alpha < 0$. However,

How we are using several evaluations in different maps,

$$\alpha = \frac{\sum_{i=1}^{n}(X_i - \bar{X}_i)(Y_i - \bar{Y}_i)}{\sum_{i=1}^{n}(X_i - \bar{X}_i)^2} \tag{1}$$

$$\beta = \bar{Y} - \alpha\bar{X} \tag{2}$$

Fig. 4. Leats squares regression

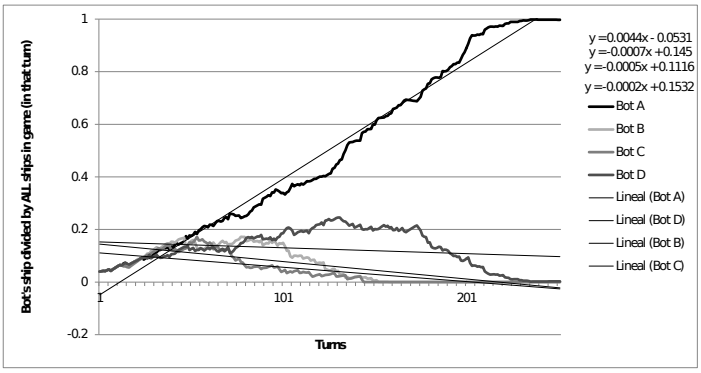

Fig. 5. Representation of the number if ships of each bot in each turn

1.3 Fitness bases in Integral-Area