システム最適化 学籍番号：50616016 氏名：毛島由裕

「バトルロイヤルゲームで勝つためには⁇」

1. はじめに

最近バトルロイヤルゲームというものが流行っている。これは100人程度のプレイヤーが同時に同じステージに降り立ち武器等を拾って他のプレイヤーを倒すなどして最後の1人になるまで戦うゲームである。このゲームではプレイヤーの性格がよく出るゲームである。腕に自信のあるプレイヤーなら他のプレイヤーを見つけしだいすぐに戦闘を挑み相手から武器等を奪うようなことをする。逆に腕に自信のないプレイヤーは他のプレイヤーに見つからないようにひたすら隠れ続けるというようなこともできる。本論文では自分自身がリスクを背負って積極的に他のプレイヤーから武器等を奪った方がよいのか、それともひたすら隠れるようなことをした方がよいのかシミュレーションによって検証する。

1. アルゴリズムの説明

まず、今回のアルゴリズムで使用する変数を示す。

i = 0 ~ 99：各プレイヤーを表すラベル、自分自身は0とする。

t：時間

：時間tにおいてプレイヤーiが生きているか死んでいるかを表す

：時間tにおいてプレイヤーiが資源を探す確率

：時間tにおいてプレイヤーiが他のプレイヤーと出会う確率

：時間tにおけるプレイヤーiの資源所持量

= 3 or 4 or 7：宝箱(バトルロイヤルが行われるフィールドでは宝箱が落ちている)、敵から得る物資の量(戦闘力だと考えてもらって差し支えない)

また、3 or 4 or 7となる確率はそれぞれ45%とし、10%の確率で7となる。

バトルロイヤルは100人で行われるものとする。時間ステップごとに以下の手順を行う。

Step1：t=0、、をセット

Step2：t = t + 1

Step3：をで回し資源を自分から探しに行くかどうかを各プレイヤーで判定しそれぞれのプレイヤーが資源を獲得

Step4：をで回し各プレイヤーが他のプレイヤーと出会い戦闘するかどうかを判定

・Step4においてStep3で自分から資源を探しに行ったプレイヤーはフィールドを駆け回るため、その分他のプレイヤーと出会う確率にバイアスを加える。

・Step4においてλの値によって他のプレイヤーと戦闘を行うわけであるがその戦闘相手の選び方は以下ようにする。

⇒他のプレイヤーと戦闘するプレイヤーの中でランダムに2組ずつ抽出しがより大きい方のプレイヤーが生き残り、より小さい方のプレイヤーが脱落する。また、両者のが等しい場合は両者ともに脱落するものとする(共倒れと考える)。さらに、最後に1人で余った場合(上記で判定された人数が奇数の場合)はそのプレイヤーは無条件で生き残るものとする。

・Step3,Step4で得られる資源(戦闘力)はGによって決まる。

Step4：生き残っているプレイヤーが1人以下なら終了、そうでなければStep2に戻る。

上記アルゴリズムを自分自身がプレイヤー0としてプレイヤー0が扱うを様々に変更し、どのような立ち回りをすればより生き残れるのかをシミュレーションする。つまり、積極的に他のプレイヤーに攻撃をしかけるのかもしくはひたすら他のプレイヤーに出くわさないように消極的な行動をするのかどのような行動が最も生き残りやすいのかをシミュレーションによって求める。

1. 結果

今回は、プレイヤーの積極性を表すを3つのパターンでそれぞれ1000回ゲームを行ったときの勝率を下に示す。

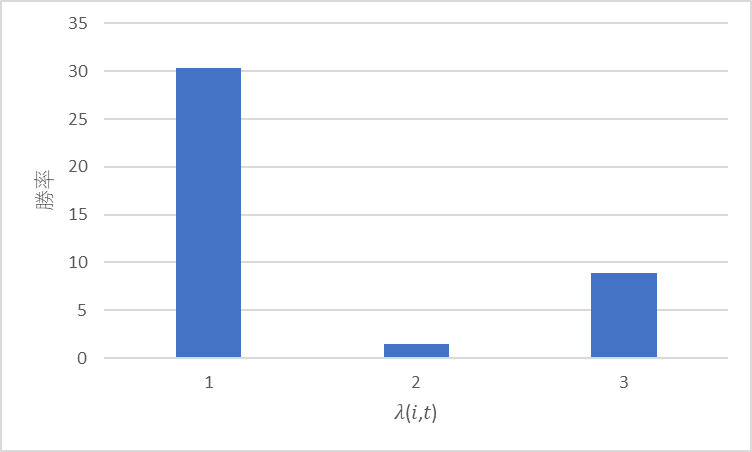


図1 各𝜆(𝑖,𝑡)におけるプレイヤー0の勝率

ここで図1横軸1は、つまり非常に積極的である。横軸2は、つまり消極的である。横軸3は、つまり今回のシミュレーションでいうところの標準である。

1. 考察

今回のシミュレーションでは結果から分かる通り積極的に他のプレイヤーから資源を奪ったりフィールドに落ちている資源を拾うなどしたほうがより最後の1人まで生き残る確率が高いことが分かった。この結果となった要因として考えられるのは今回のシミュレーションでは他のプレイヤーと出会った際、より資源を持っている方が生き残ると設定したからであると考えられる。また、プレイヤー同士が戦闘をする際、どちらが生き残るかで考えられる他の要因としてはプレイヤー自身のゲームの上手さがある。実際にこのゲームをプレイする際、いくら強い武器を持っていても相手の方がプレイヤースキルが上の場合には負けることが多々あるため、そういった細かい要因も考慮すればさらに精度の高いシミュレーションを実現できると考えられる。

プログラム実行結果

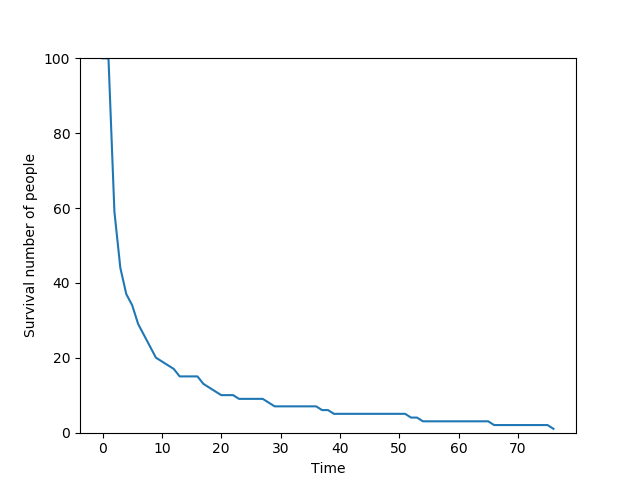


図2 時間ステップごとの生存者数の推移

端末上での実行画面

(base) C:\Users\user\Desktop\システム最適化>python Fortnite.py

Winner is player 66

Winner is none or I lose

Winner is player 0

Winner is player 87

Winner is none or I lose

Winner is player 72

Winner is none or I lose

Winner is player 12

Winner is none or I lose

Winner is none or I lose

Winner is none or I lose

・・・

・・・

・・・

Winner is none or I lose

Winner is none or I lose

victory times is 6