

多様でかつ格差の小さい ターン制RPGキャラクタ群のパラメータ調整法

今井 淳^{1,a)} 池田 心^{1,b)}

概要：近年の複雑かつ大型なゲームにおいては、ゲームバランス調整の困難さが課題となっている。ゲームバランスはプレイヤーの満足度に深く関わっているため、人間のデザイナーの代わりに AI の活用によって自動調整を行うことが求められていると考える。我々は、ターン制ロールプレイングゲーム (RPG) を対象として、多様でかつバランスが取れており、キャラクタの組み合わせ (パーティ) を考えることが楽しくなるようなキャラクタ群パラメータの自動生成を試みた。パラメータの生成手法として、進化的アルゴリズムによる最適化を用いた。その評価には AI によるシミュレーションを用いて、多様さや強さに関する 4 つの評価関数を組み合わせた。本手法により生成したパラメータを評価した結果、様々な個性を持ったうえで平均勝率が同程度であり、かつ比較的高い勝率となる組み合わせが数割程度あって、そして強すぎるパーティがほぼ存在しないようなキャラクタ群が生成されることを確認した。

キーワード：ロールプレイングゲーム、コンテンツ生成、ゲームバランス、進化的アルゴリズム

1. はじめに

近年、人工知能技術の進歩によりゲームにおける AI プレイヤーの性能の向上は著しく、人間よりも強い AI プレイヤーの生成に関して一定の成果が挙げられている。そのような人間の対戦相手や仲間としての AI の活用の一方で、ゲームの開発現場での省力化に AI を用いる研究もまた取り組まれている。

複雑かつ大型なゲームにおいては、不具合修正やゲームバランス調整といった品質保証 (Quality Assurance, QA) の困難さが課題となっている。さらに、近年ではゲームの発売後に更新を随時行い、新規キャラクタなどのコンテンツを追加する形式が主流であり、QA の自動化はコストの削減や多様なコンテンツの提供といった観点において急務である [1][2]。特に、それら品質のうちゲームバランスはプレイヤーの満足度に深く関わっているため、人間のゲームデザイナーの代わりに AI の活用によって自動調整を行うことが求められる。

ゲームバランスの自動調整に関する研究はいくつか行われており、ゲーム開発現場への導入事例も存在してい

る [3]。既存ゲームの QA に AI を用いる研究は盛んに行われているが、古典的かつ有名なターン制ロールプレイングゲーム (ターン制 RPG) に対しては、ゲームバランスの調整と多様なコンテンツの生成を同時に達成するような研究は、我々の知る限りでは行われていない。これらを踏まえ、本研究では QA の中でも RPG を対象として“キャラクタの能力を示すパラメータ”に注目し、ゲームバランスを自動調整することを試みる。

多くのターン制 RPG は、1 人または複数のプレイヤーで遊ばれるゲームであり、戦闘と探索を繰り返すことで「魔王を倒す」といった大目的の達成を目指す。RPG では戦闘の場面において、プレイヤーは攻撃力・防御力・素早さなどのステータスを持つキャラクタ群から数体選択してパーティを編成することも多く、各キャラクタを操作して攻撃・防御・スキルといった行動を駆使しながら敵パーティと戦闘する。

ゲーム開発においては、これらのステータス・行動へキャラクタごとに異なるパラメータを設定して個性を表現することで、多様なキャラクタを選択可能とする。そうすることで、「攻撃力の高いこのキャラクタとそれを高める能力を持つこのキャラクタを組み合わせよう」・「このキャラクタは防御力が低いから回復が得意なキャラクタを組み合わせよう」というような様々な戦略性をゲームへもたらし、面白さに繋げることができる。しかしながらキャ

¹ 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology, Nomi,
Ishikawa 923-1211, Japan

a) jun_imai@jaist.ac.jp

b) kokolo@jaist.ac.jp

ラクタ間の強さに偏りがある場合、特定のキャラクタのみ用いられる事態が生じる。その結果、「パーティ選択や戦略選択の自由度が減る」などといった面白さの低下につながり、プレイヤーは不満を覚えることが考えられる。そのため、ゲームバランスを調整するうえでは個性を持ち多様でありつつも、強さについて格差が小さいようなパラメータの作成・設定が必要であることが多いと考え、その方法論を提案することを本研究の目的とする。

通常、ゲーム開発現場においてゲームバランスの調整はテストプレイのデータに基づいて行うが、それには膨大な量が必要であるうえ、上級者など一部プレイヤーのデータしか反映されないといった問題がある。そのため本研究では『キャラクタのパラメータを生成および調整する機構』と『AI プレイヤーによるシミュレーション』から成るシステムを構築し、キャラクタ群パラメータの自動生成を目指す。

2. 関連研究

2.1 多様なゲームコンテンツの生成

任意のアルゴリズムを用いてダンジョンやキャラクタといったようなゲームコンテンツの自動生成を試みることを Procedural Content Generation (PCG) と呼ぶ。PCG では、コンテンツに応じて生成手法や用いられる AI 技術もまた様々に存在するが、これらについて頻繁に使われるものに Search-based PCG[4][5] というグループがある。Search-based PCG では、コンテンツの生成過程において、評価関数による選別と、何らかの手法に基づき評価を更新するようなコンテンツを生成することで、より良いものへの入れ替えを繰り返し行う。その評価には AI プレイヤーによるゲームのシミュレーションの結果を用いることが多い。

Search-based PCG の枠組みの中で、品質が高く、かつ多様であるようなコンテンツを生成する研究はいくつか行われている。Gravina らは PCG の新たなコンセプトとして PCG Quality-Diversity を提示しており [1]、この中で Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [6] や Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [7] といったアルゴリズムによる品質多様性を維持したゲームコンテンツ生成の研究をいくつか挙げている。例えば、Alvarez らは Map-Elites を用いて、多様なインタラクティブデザインが可能であるようなローグライクゲームのダンジョン生成を行っている [8]。

2.2 ゲームバランスの調整

ゲームバランスの調整に関してはいくつかのアプローチが存在する。眞鍋らは、『グリムノーツ』[9] を対象ゲームとして、不具合やゲームバランスが崩壊している部分を検知するために、遺伝的アルゴリズムによって QA 作業の効率化を図っている [2]。また、de Mesentier Silva らは、『Hearthstone』[10] を対象ゲームとして、デッキ間の何す

くみかの循環関係(メタゲーム)をより改善するために、バランス変更の対象となるカードを見つけるような仕組みを提案している [11]。

3. アプローチ

RPG の戦闘シーンにおいて、パーティ内の各キャラクタはそれぞれ得意な行動をとり、敵に大打撃を与えたり回復を付与するなどして、補完的な関係を取りながら戦闘を進行させる。これらの行動の能力を示すパラメータを様々な設定することで、キャラクタ群へ多様な性質を与えることが出来る。そのため、キャラクタ群のパラメータはゲームバランスの調整や、多様なコンテンツの生成の対象となるといえる。そこで、次節以降では、戦闘シーケンスの基本的な流れと、その中でキャラクタのパラメータと役割についての考察を行い、アプローチについて検討する。

3.1 RPG の戦闘におけるキャラクタについての考察

RPG では、各キャラクタが「攻撃力」・「防御力」・「素早さ」・「回復力」といったパラメータを持ち、「敵を攻撃する」・「味方を回復する」などの行動をとる。これらのパラメータがキャラクタごとに異なることで、パーティ内に役割分担や相性が発生し、その編成を楽しむことができる。

キャラクタのパラメータと、そのパラメータに最もふさわしいような行動に注目した際、おおむね次のような 5 種類の役割に分類することができる。

アタッカー (Attacker)

攻撃の中心となる役割。主に攻撃力が高いキャラクタがこの役割を担う。どのキャラクタも攻撃は可能であるが、短い間に高ダメージを稼ぎたい戦闘シチュエーションにおいて有効的である。また、攻撃に関するスキルによってこの役割を担う場合、**スキルアタッカー (Skill Attacker)** と呼称されることもある。

ディフェンダー (Defender)

パーティの防御の中心となる役割。行動またはスキルによって味方キャラクタを相手の攻撃から守る。体力や防御力が高いキャラクタがこの役割を担う。

バッファー (Buffer)

任意の対象を一時的に強化する「バフ」、その反対に弱体化させる「デバフ」といったスキルを駆使し、味方をサポートする役割。例えばアタッカーに攻撃力バフをかけて大ダメージを狙う、といった戦略が可能。

ヒーラー (Healer)

「回復」のスキルにより味方をサポートする役割。パーティ全体の行動を安定的に持続させるためには重要な役割である。

クラウドコントローラー (Crowd Controller)

「眠り」・「麻痺」といった状態異常のスキルを駆使し、

相手を無力化することで味方をサポートする役割。

以上のような役割が機能することで戦闘は戦略性を帯びつつ進行する。なお、1キャラクターにつき上記の役割のうち1つを担うというわけでもなく、複数を担うことが多い。例えば自身に攻撃力バフをかけられるアタッカー、回復も状態異常も行うキャラクター、などといったようにである。キャラクターが持ちうる役割が多様であるほど、プレイヤーが考えられるパーティの組み合わせもまた多様さを帯び、ゲームに面白さを与える。従って、このような役割を持ちうるキャラクターが生成されるべきである。

3.2 キャラクター群パラメータ生成手法・パラメータ評価手法についての検討

前節では、RPGの各キャラクターがもつ役割が多様であるべきであると述べた。本節では、キャラクター“群”が持つべき性質について考察する。

望ましいキャラクター群とは、プレイヤーに楽しさを与え、ひいてはゲーム制作者に利益を与えるようなものであると考える。楽しさには様々なものがありえ、例えばキャラクターの外見や設定、仕草などが重視されるゲームもある。本研究では、いわゆる思考ゲームとしての側面のみを対象に、「複数のキャラクター群から数体キャラクターを選択し、パーティを編成して特定の敵パーティと戦闘する」状況に限定して検討を行う。このとき、考慮すべき項目は以下の通りとなる。

- (i) パラメータの格差によって、あるキャラクターが強すぎる、あるいは弱すぎるという状況が生じることは望ましくない。これらのような状況により、「ある特定のキャラクターのみをパーティに編成する」といった事態が発生し、パーティの組み合わせを考慮する自由度や、採りうる戦略の幅が減少する可能性がある。
- (ii) どのキャラクターも似たようなパラメータだとつまらない。例えば3.1節で述べた役割が1~2種類程度しかない、取りうるパーティの組み合わせが少なく面白さを感じられない。
- (iii) 各キャラクターが同程度の強さで、異なるパラメータを持っていたとしても、どのキャラクターを組み合わせても同じくらいにそこそこ勝てる、という状況は面白くない。パーティ内の役割の組み合わせを考慮することで勝利しやすくなるといったような、優位に立てる状況がある程度存在することが望ましい。

本研究においては、以上のような項目を満たすことが出来る生成手法、および評価手法を用いるべきである。特に、(i)と(iii)のようにトレードオフの関係にある項目が存在することを考えると、それらについて定式化し、重みづけを行うといった操作が可能であるような手法が望ましい。本研究では、実装の容易さなども含めて考慮したうえで、Search-based PCGの枠組みに依ったキャラクター群パ

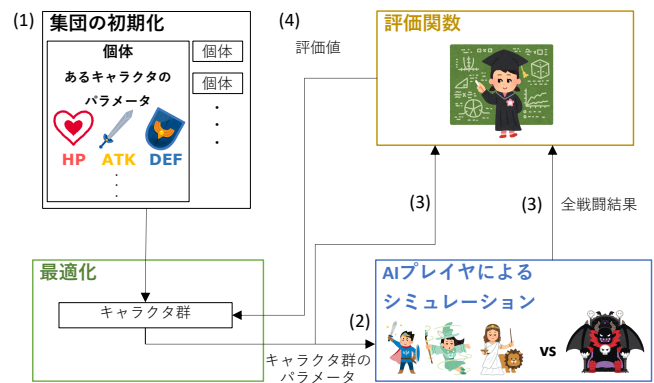


図 1: キャラクター群パラメータ生成・調整システムの概観

ラメータの生成および評価を行うこととした。そして、上記の項目を満たすような評価関数を設計し、これを生成したコンテンツの評価へ用いることにした。次節では、本研究における生成手法の概観について説明し、具体的な評価関数については5.3節において詳述する。

3.3 本研究における生成手法の概観

本研究における Search-based PCG に依ったキャラクター群パラメータ生成・調整システムの概観を図1に示す。図中の要素番号を用いて、本システムは以下のようなフローを持つ。

- (1) 任意の調整範囲内でキャラクター群のパラメータをランダムに生成する
- (2) 生成したパラメータをキャラクター群へ入力し、これらキャラクター群を用いてAIプレイヤーによるRPGの戦闘シミュレーションを一定回数行う
- (3) プレイ結果を評価関数に代入する。なお、直接パラメータから導出される指標についてもここで評価関数に代入する
- (4) 評価関数から出力される評価値に基づき最適化を行い、次世代のキャラクター群パラメータを生成する
- (5) この(1)~(5)が一定回数行われていけば終了する。そうでない場合は(2)へ戻る

なお、図中のシミュレーションのベースとなるRPGプラットフォームについては、続く4節で説明し、最適化手法は5.1節、AIプレイヤーは5.2節、そして評価関数は5.3節において詳細な実装内容を述べる。

4. RPGプラットフォームの提案

既存のターン制RPGでは、多種多様なシナリオやゲームモード、自由に行動できる広大なマップなどが盛り込まれている。しかし、本研究では分析を容易に行うために、これらについて取り扱わず、ターン制RPGから戦闘シーケンスのみを抽出したゲーム環境を構築し、これを研究用のプラットフォームとする。

3.2節では、キャラクター群の望ましい特徴について考察

した。これらの特徴を満たすため、コンセプトとしては、キャラクターのパラメータや行動に関して、プレイヤーやゲームデザイナーが様々な悩み工夫する余地があるようなゲームシステムを有し、かつその他の部分で複雑すぎないようなプラットフォームを構築することとした。

以下に、本研究で用いたゲーム環境における諸要素およびシステムを列記する。

戦闘と勝利条件

本プラットフォームでは、冒険者である味方パーティが、複数のモンスターで構成される敵パーティと1対1で戦い、全てのモンスターを倒すことが勝利であると定義した。また、個々の戦闘は独立しているものとし、経験値などによるキャラクターの育成システムは取り扱わないこととした。

行動決定

本プラットフォームでは各キャラクターの行動順は素早さに基づき定まり、そのたびに行動選択・対象選択を行い、決定後に即時に行うとした。また、素早さ順にすべてのキャラクターがそれぞれ最大1回だけ行動するものとし、これを1ターンと定義した。

スキルシステム

一般的なRPGの戦闘では、通常の攻撃や防御といった行動とは趣の異なる特殊な行動が存在する。これらは総じて「スキル」などと呼称される。本プラットフォームにおいては、スキルポイント (SP) というパラメータを導入し、これをいくらか消費することでスキルが行えるものとした。本プラットフォームにおけるスキルシステムの主な特徴は次の通りである。

- (1) SPはターンの経過ごとに1だけ加算される。例えば3対3の戦闘だったとして、6体のキャラクターが全員行動し終わったあとにSPが加算される。なお、SPの初期値は0である。
- (2) SPはパーティで共有する。例えばあるターンで、自身がスキルを使用してSPを使い切った場合、以降の行動順の味方はそのターン以内ではスキルを使うことができなくなる。

パラメータ

キャラクターのパラメータは、大別して、体力・攻撃力・防御力・素早さといったキャラクターの身体的な要素を示す基本パラメータと、スキルの消費SPや威力のもととなるスキルパラメータから成るものとした。なお、本論文の実験で用いるパラメータについては以下の11種とする。

体力: 0になると行動不能となる。攻撃か攻撃スキルを受けた場合、後述のダメージ計算に基づいて減少する。

攻撃力: 攻撃または攻撃スキルの威力となる。

防御力: ダメージを低減させる度合いとなる。

素早さ: 行動順を定める値。

攻撃倍率: 攻撃スキルの能力値。攻撃力に乗算できる補正値を示す。

回復率: 回復スキルの能力値。最大体力のうち、何割回復させられるかを示す。

バフ倍率: バフスキルの能力値。攻撃力に乗算できる補正値を示す。

バフ期間: バフスキルの能力値。バフの持続するターン数を示す。

消費SP: 各スキルの使用時に必要となるSP。

行動

本プラットフォームでは、既定の職業はなく、どのキャラクターでも攻撃・防御・攻撃スキル・回復スキル・バフスキルの5つの行動を行えるものとした。ただし、個々のパラメータによっては、事実上回復やバフといったスキルを用いる意味がないキャラクターも生成されることになる。以下に、本論文の実験に用いた各行動の詳細を示す。

攻撃: 敵1体にダメージを与える。ダメージ計算は(攻撃力) * (バフ付与時のバフ倍率) - (防御力) * (防御力の補正值)となる。

防御: 攻撃を受けるか、自身の行動が再度回ってくるまで、自身の防御力に補正值1.5が乗算される。

攻撃スキル: 敵1体にダメージを与える。ダメージ計算は(攻撃力) * (攻撃倍率) * (バフ付与時のバフ倍率) - (防御力) * (防御力の補正值)となる。

回復スキル: 自身を含む味方1体の体力を最大体力のうち回復率だけ回復させる。

バフスキル: 自身を含む味方1体の攻撃力にバフ期間だけバフ倍率を乗算する。なお、使用したターンを含むターン数の間だけ効果は持続する。また、重ねて付与した場合、新たなバフ倍率・バフ期間に上書きされる。

5. アプローチの各部詳細

本節では、3.2節において述べた、望ましいキャラクター群パラメータの生成手法として、3.3節で提案したキャラクター群パラメータ生成・調整システムの各部詳細について述べる。

5.1 進化的アルゴリズム

進化的アルゴリズム (Evolutionary Algorithm, EA) とは、生物の進化を参考にして、確率的に探索を行う最適化アルゴリズムの総称である。本研究では、個体を1つのキャラクターのパラメータとして、個体群全体が「多様で強さのバランスが取れている」ようになるように最適化を行う。

一般的なEAでは、交叉によって子供となる個体を作成し、個体群を評価したあとに優れた個体のみを選択する、という操作を行う。しかし、ある個体が高個体とのパー

ティを組むことを想定している場合、交叉を行うと、親となる個体とは似たようなパラメータになることが懸念される。その結果、多数の割には最終的に似通ったキャラクターパラメータが生成されることになり、これは本研究の目的とは相反する。

そこで、本研究では、交叉を行わずに、解の多様性を保つことが可能であるようなEAを実装し、これを用いることにした。以下に、本研究で用いた実験条件のパラメータとともに、その手順を示す。

- (1) n_{chara} の個体をランダムに生成し、これを個体群とする。このとき、パラメータは事前に定めた調整範囲内で定めるものとする。
- (2) 個体群からランダムに1個体 p を選択する。
- (3) 個体群から p 以外の2つの個体 p_1, p_2 を選択する。この3体1組をパーティとして、RPGの戦闘のシミュレーションを行う。これを試合回数 m 回繰り返す。
- (4) (3) を p のパーティの総組合せ ${}_{n_{chara}-1}C_2$ だけ行い、この全結果に基づいて、適度な強さを持っているかに関する評価値を導出する。
- (5) p の他個体とのパラメータを比較し、他のキャラクターと似すぎていないかに関する評価値を導出する。
- (6) (4), (5) を線形和して評価値 $f(p)$ とする。
- (7) p のパラメータを突然変異させて p' を作成する。これは、 p の各パラメータに対してランダムに0.9~1.1倍することで行う。
- (8) p' について(3)~(6)を行い、 $f(p')$ を求める。
- (9) $f(p) < f(p')$ ならば、 p を p' に置き換える。
- (10) 探索回数 n に達していたら終了する。そうでなければ(2)に戻る。

5.2 シミュレーション用 AI プレイヤ

前節で述べた手順(3)では、AIプレイヤによる戦闘シミュレーションが必要となる。ここでは、AIプレイヤはキャラクターのパラメータに対して合理的な行動をとるようなものであるべきである。AIプレイヤの実装方法は複数考えられるが、木探索アルゴリズムや強化学習などを用いると、EAによる最適化のループと相まって計算コストは膨大なものとなる懸念が生じる。

そこで本研究では、とりうる行動(および対象)それぞれに状況を踏まえて優先度を付与し、この優先度に基づいて行動・対象選択を行うAIプレイヤを用いることとした。このようなAIプレイヤは、特定の複雑すぎない問題に関して、対象への事前知識が十分にある人間が作成した場合であれば、“賢い”とまではいかずとも、ある程度の合理性を持つものになると考える。

以下に、本研究で用いた優先度計算方法の一例として、攻撃行動に関する優先度の計算法を示す。なお、全ルールの詳細については第1著者の修士論文を参照されたい。

- (I) 今日のスキルも使えず、攻撃が生きている味方の平均より高ければ、優先度を1000とする。そうでなければ0とする。
- (II) (I)に関わらず、攻撃によって、敵1人を倒すことができるなら、優先度に500足す。
- (III) (II)の場合ただし、与えるダメージがその敵の残り体力の3倍以上なら、優先度を100減らす。
- (IV) (II)の場合ただし、その敵の行動前に味方の行動が2つあるなら、優先度を200減らす。
- (V) (I)~(IV)に関わらず、ランダムに-200~200の範囲内で優先度を加算する。

5.3 評価関数

3.2節では、望ましいキャラクターパラメータの在り方について考察を行い、その満たすべき項目とは以下の通りであった。

- (a) キャラクター間の強さについての格差が小さいこと
- (b) キャラクターの持ちうる役割が豊富であること
- (c) 比較的優位なパーティの組み合わせがある程度は存在すること

これらのうち、(a)については、個体の全平均勝率が0.5を超えないようにする評価項目を導入した。

(b)については、各個体のパラメータをベクトルとみなし、これに距離関数を用いて、“パラメータが他個体とどれだけ距離が離れているか”を評価することで、各キャラクターのパラメータに多様性が生じるようにして、それぞれ異なった役割を持てるようにした。

(c)については、ある個体 p の「比較的優位なパーティの組み合わせ」を、その平均勝率 wr_i が0.5~0.7の範囲内であるものとし、このときの組み合わせをシナジーを持つパーティであると定義した。そして、「そのようなシナジーを持つパーティが、個体の採りうるパーティの総組み合わせに対してどれだけ存在するか」「シナジーを越えて強すぎる組み合わせがどれだけ存在するか」を評価関数として取り扱うこととした。

以上の内容を踏まえ、ある個体 p の評価関数を式1のように定めた。 f_s はシナジーに関する評価値、 f_d はパラメータの距離に関する評価値、 f_{pen_i} は強すぎるパーティへのペナルティ項、 f_{pen_p} は強すぎる個体へのペナルティ項、 $\beta, \gamma, \delta (> 0)$ はそれぞれ線形和パラメータである。

$$f(p) = f_s(p) + \beta f_d(p) - \gamma f_{pen_i}(p) - \delta f_{pen_p}(p) \quad (1)$$

シナジーに関する評価値 f_s は式2で求める。ここで s_p はシナジー点数、 s_p^* は任意に定める最適なシナジー点数である。シナジー点数 s_p とは、個体 p のあるパーティの平均勝率 wr_i を式3に示すような関数で得点付けし、これを全パーティに対して行い、式4のように合計したものである。式3は、あるパーティの平均勝率 wr_i が0.5~0.7の範囲内

であるほど高得点となる。すなわち、式2は、シナジーを持つパーティの数が設定した s_p^* に近いほど高評価であることを表している。

なお、 σ_1, σ_2 は f_s のチューニングパラメータである。今回の実験では、「シナジーを持つパーティは総組み合わせに対して1/3程度存在することが望ましい」と定義して、それに基づき $s_p^*=12, \sigma_1=3.5, \sigma_2=7.5$ とした。そのときの式(2)は図2のように図示される。

$$f_s(p) = \begin{cases} \exp\left\{-\frac{(s_p-s_p^*)^2}{2\sigma_1^2}\right\} & (0 \leq s_p \leq s_p^*) \\ \exp\left\{-\frac{(s_p-s_p^*)^2}{2\sigma_2^2}\right\} & (s_p^* < s_p) \end{cases} \quad (2)$$

$$s_{p_i}(wr_i) = \begin{cases} 0 & (0 \leq wr_i \leq 0.45 \vee wr_i > 0.75) \\ 10wr_i - 4.5 & (0.45 < wr_i \leq 0.55) \\ 1 & (0.55 < wr_i \leq 0.65) \\ -10wr_i + 7.5 & (0.65 < wr_i \leq 0.75) \end{cases} \quad (3)$$

$$s_p = \sum_i s_{p_i} \quad (4)$$

また、強すぎるパーティ、強すぎる個体へのペナルティ項 f_{pen_i}, f_{pen_p} は、それぞれ式5、式6で求める。式5はパーティの平均勝率 wr_i が0.7を超えるほど、またそのようなパーティが多いほど、式6は個体の全平均勝率 wr_p が0.5を超えるほど $f(p)$ が低くなることを示している。

$$f_{pen_i}(p) = \sum_i \max(wr_i - 0.7, 0) \quad (5)$$

$$f_{pen_p}(p) = \max(wr_p - 0.5, 0) \quad (6)$$

一方、パラメータの距離に関する評価値 f_d は式7で求める。ここで $\text{dist}(p, q)$ は距離関数であり、個体 p と他個体 q とのパラメータのベクトルの距離を計算する。本論文の実験では、距離関数にはユークリッド距離を用いた。なお、個体のパラメータについては、調整範囲となる最小値と最大値をあらかじめ定め、それぞれ0, 1とする正規化を各パラメータに対して行い、これを距離関数へ代入するものとした。

$$f_d(p) = \frac{1}{\sum_{q \neq p} \frac{1}{\text{dist}(p, q)^2}} \quad (7)$$

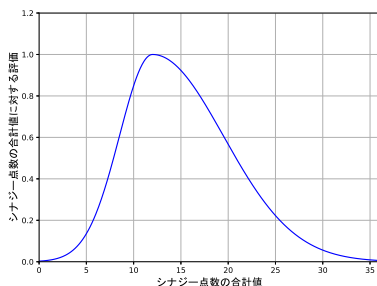


図2: あるキャラクターのシナジー点数の合計値に対する評価

6. キャラクタ群パラメータ生成実験・評価

6.1 目的

本節における実験の目的は、(A) 3.2節で説明した望ましい特徴を持つキャラクタ群パラメータの生成・評価、(B) 線形和パラメータの変化による生成結果の変化の確認、の2つである。次節以降では、これらの目的に基づいた条件と結果、そして考察について述べる。

6.2 条件

本実験は、以下のような条件で行った。

進化的アルゴリズム

実験目的(B)のために、多様性に関する評価値 f_d の重みである線形和パラメータ β を1, 10, 100と変化させて比較実験を行った。それ以外については、個体数 $n_{chara} = 10$ 、探索回数 $n = 5000$ 、1パーティごとの試合回数 $m = 10$ 、線形和パラメータ $\gamma = 0.5, \delta = 10$ として行った。

キャラクターのパラメータと調整対象

キャラクターのパラメータは、4節にて述べたパラメータ11種とした。これらのパラメータのうち、本実験では調整対象を体力・攻撃力・防御力・素早さ・攻撃スキル倍率・回復率・バフ倍率の7つに限定した。表1にこれらのパラメータの調整範囲となる最小値と最大値を示す。また、それ以外のバフ期間および各スキルの消費SPについては2に固定した。

シミュレーション条件

1パーティを3体とし、同じく3体から成る敵パーティとの戦闘を行う。なお、敵パーティの構成は、キャラクターとは相対的に強いボス1体と、相対的に弱いミニオン2体とした。表2にボスとミニオンのステータスを示す。本実験では、ボスは攻撃や攻撃スキルが得意であり、ミニオンは回復スキルやバフスキルが得意であるような設定とした。

表1: 調整対象のパラメータとその範囲

	体力	攻撃力	防御力	素早さ	攻撃倍率	回復率	バフ倍率
最小値	200	150	50	1.0	1.5	0.1	1.1
最大値	400	400	100	20.0	3.0	0.5	2.0

表2: ボスとミニオンのステータス

	体力	攻撃力	防御力	素早さ	攻撃倍率	回復率	バフ倍率
ボス	1300	200	100	5.0	2.5	0.1	1.1
ミニオン	200	50	100	15.0	1.5	0.4	3.0

	バフ持続期間	攻撃スキル SP	回復スキル SP	バフスキル SP
ボス		1	2	1
ミニオン		2	1	2

6.3 結果

実験は CPU: AMD Ryzen 5 5600X, RAM: 32GB の PC 環境下で行い、平均探索時間は 4 時間程度であった。線形和パラメータ $\beta = 10$ の場合を一例として、図 3 に全個体平均・各個体の線形和評価値の推移を示す。図 3 より、評価値の推移としてはやや振れはあるものの、良好に探索が行われているといえる。

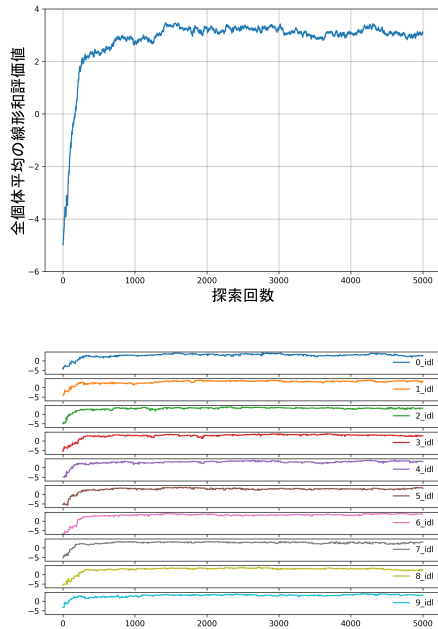


図 3: 線形和評価値の推移 ($\beta = 10, \gamma = 0.5, \delta = 10$)

また、 $\beta = 10$ の場合に得られた 10 体の個体について、パラメータおよび、パラメータから想定される役割を表 3 に示す*1。さらに同表中に、集団内の他個体とのパーティを組み ${}_{10-1}C_2 = 36$ 通りの試合をそれぞれ 200 回行った結果 (以下、個体評価結果) について、勝ち数をカウントし $36 * 200$ で割った全平均勝率、平均個体間距離 $\frac{\sum_{q \neq p} \text{dist}(p,q)}{n_{\text{chara}} - 1}$ 、シナジー点数を示す。

表 3 より、各パラメータが調整範囲内ではばらついており、また想定される役割についても様々な存在することが確認できる。

6.4 考察

初めに実験目的 (A) についての考察を行う。図 4 に、個体 1、個体 8、個体 3 の集団内の他個体とのパーティにおける平均勝率のヒートマップを示す。これは、シナジー点数が設定した最適値 s_p^* に対して、それぞれ最大、理想値、最小の個体をピックアップしている。図 4 より、個体差はあるものの、各個体に優位な組み合わせがある程度存在していて、かつ適度な強さであることがわかる。

*1 3.1 節で述べた役割と表 3 の略記の対応は次の通りである。
 アタッカー: ATK, スキルアタッカー: S-ATK, バッファー: BUF, ヒーラー: HEA

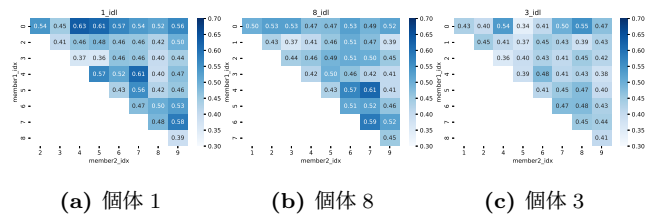


図 4: 各個体の集団内の他個体とのパーティにおける平均勝率

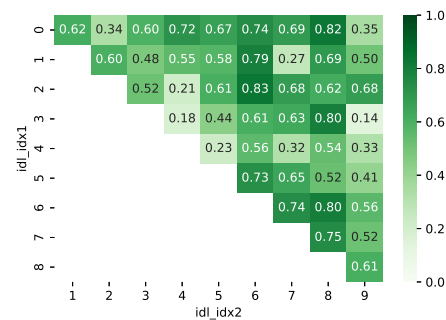


図 5: 個体間の正規化パラメータのコサイン類似度

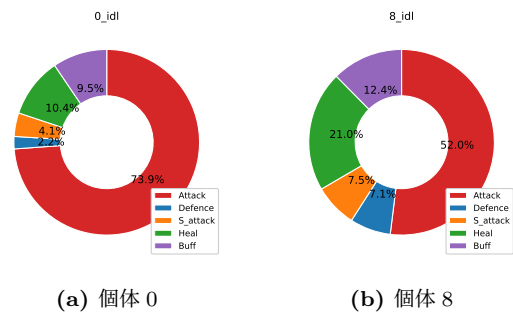


図 6: 個体評価結果における行動割合

また、図 5 に個体間のパラメータのコサイン類似度を示す。これは、各個体の調整対象パラメータに対して正規化を行い、それぞれの個体間のコサイン類似度を計算してヒートマップで図示したものである。これより、例えば個体 0 と個体 8 はパラメータについて比較的似た個体だといえる。しかしながら、これらの個体についてはパラメータは似ているものの、図 6 に示す個体評価結果における行動割合を参照すると、個体 0 は攻撃使用を中心としているのに対し、個体 8 は回復スキルやバフスキルも多く行っている。これより、個体 0 は回復スキルとバフスキルを扱えるアタッカー、個体 8 は回復スキルを比較的多く行うヒーラー&バッファー、と捉えられる。このように、パラメータが似ていたとしても、その役割としては異なるキャラクターが存在することが確認できる。

次に実験目的 (B) についての考察を行う。表 4 に線形和パラメータ β ごとの生成結果を示す。ここで強すぎるパーティ数とは各個体について平均勝率 $wr_i > 0.7$ であるパーティを数え上げたものである。強さ評価を重視した場合、平均個体間距離が最小であることから、似たようなキャラクターが多いといえる。また、多様性評価を重視した場合、

表 3: 各個体のパラメータと想定役割・全平均勝率・平均個体間距離・シナジー点数

個体	体力	攻撃力	防御力	素早さ	攻撃倍率	回復率	バフ倍率	想定役割	全平均勝率	平均個体間距離	シナジー点数
個体 0	220	286	65	15.7	1.6	0.4	1.8	ATK&BUF	0.462	1.32	11.4
個体 1	349	189	53	16.9	2.9	0.4	1.1	S-ATK&HEA	0.486	1.50	14.5
個体 2	395	154	98	2.3	2.9	0.1	1.5	S-ATK	0.448	1.49	6.0
個体 3	200	213	100	19.7	1.9	0.1	1.1	ATK	0.431	1.49	3.6
個体 4	233	259	57	1.1	1.5	0.5	1.2	ATK&HEA	0.453	1.52	8.7
個体 5	237	250	54	9.8	2.7	0.1	2.0	S-ATK&BUF	0.448	1.42	7.4
個体 6	374	153	99	14.2	3.0	0.5	2.0	HEA&BUF	0.474	1.59	12.0
個体 7	272	233	98	6.3	1.5	0.1	2.0	BUF	0.482	1.38	14.1
個体 8	344	257	89	19.9	1.5	0.3	1.4	ATK&HEA	0.477	1.27	12.5
個体 9	397	215	55	1.5	1.6	0.1	1.3	ATK	0.456	1.45	9.0

各個体の全平均勝率の標準偏差・最大個体差が増加し、かつ強すぎるパーティの存在する個体が増加していることから、強さに格差があり、かつ強すぎ・弱すぎのキャラクタがいることがわかる。これらの結果から、5.3 節にて設計した評価関数は想定通りに機能しており、線形パラメータを変更することで設計者が「格差の小ささ」と「多様さ」を制御できることが分かった。

表 4: 線形パラメータ β ごとの生成結果

	強さ評価重視	バランス評価	多様さ評価重視
線形パラメータ β	1	10	100
全平均勝率	0.46±0.01	0.46±0.02	0.48±0.09
全平均勝率の最大個体差	0.03	0.05	0.25
強すぎるパーティ数の平均	0	0	4.6
平均個体間距離	1.07	1.44	1.84

7. おわりに

本研究では、ターン制 RPG において多様でかつ強さについての格差が小さいようなキャラクタ群パラメータの生成手法の提案を行った。望ましいキャラクタ群のパラメータを考察し、そのような特徴を持つキャラクタ群を生成するために、AI プレイヤによるシミュレーションと進化的アルゴリズムによる最適化からなる自動生成システムを構築した。生成実験の結果、統計量などに基づき評価して、望ましい特徴を持つキャラクタ群パラメータの生成を確認できた。

今後の展望としては、さらにキャラクタ数やパラメータ、行動の種類を増加させたうえでキャラクタ群パラメータの生成を試みることがある。また、AI プレイヤに木探索アルゴリズムや強化学習を用いることで、よりシミュレーションを洗練させることも考えられる。いずれについても、実装した RPG プラットフォームや最適化手法の高速化が必要となるため、これについても改善を施していきたい。さらに、被験者実験などを通じて、本手法の有効性も検証していきたい。

参考文献

- [1] Gravina, D., Khalifa, A., Liapis, A., Togelius, J. and Yannakakis, G. N.: Procedural content generation through quality diversity, *2019 IEEE Conference on Games (CoG)*, IEEE, pp. 1–8 (2019).
- [2] 眞鍋和子, 三宅陽一郎ほか: 遺伝的アルゴリズムを用いたプレイヤーエージェントによるデジタルゲームのバランス調整, *SIG-SAI*, Vol. 30, No. 2, pp. 1–6 (2017).
- [3] Gudmundsson, S. F., Eisen, P., Poromaa, E., Nodet, A., Purmonen, S., Kozakowski, B., Meurling, R. and Cao, L.: Human-like playtesting with deep learning, *2018 IEEE Conference on Computational Intelligence and Games (CIIG)*, IEEE, pp. 1–8 (2018).
- [4] Togelius, J., Yannakakis, G. N., Stanley, K. O. and Browne, C.: Search-based procedural content generation: A taxonomy and survey, *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 3, No. 3, pp. 172–186 (2011).
- [5] Togelius, J., Yannakakis, G. N., Stanley, K. O. and Browne, C.: Search-based procedural content generation, *European Conference on the Applications of Evolutionary Computation*, Springer, pp. 141–150 (2010).
- [6] Mouret, J.-B. and Clune, J.: Illuminating search spaces by mapping elites, *arXiv preprint arXiv:1504.04909* (2015).
- [7] Hansen, N. and Ostermeier, A.: Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, *Proceedings of IEEE international conference on evolutionary computation*, IEEE, pp. 312–317 (1996).
- [8] Alvarez, A., Dahlskog, S., Font, J. and Togelius, J.: Empowering quality diversity in dungeon design with interactive constrained map-elites, *2019 IEEE Conference on Games (CoG)*, IEEE, pp. 1–8 (2019).
- [9] スクウェア・エニックス: グリムノート (2016).
- [10] Blizzard Entertainment: ハースストーン (2014).
- [11] de Mesentier Silva, F., Canaan, R., Lee, S., Fontaine, M. C., Togelius, J. and Hoover, A. K.: Evolving the hearthstone meta, *2019 IEEE Conference on Games (CoG)*, IEEE, pp. 1–8 (2019).