

エージェントベース手法によるターン制戦略ゲームのマップ景観生成およびヒルクライム手法によるチーム戦力バランスの自動調整

内田直浩¹ 佐藤直之¹

概要: ターン制戦略ゲーム, と呼ばれる大規模な戦争を模したゲームにおいて, 1 タイトルごとに人間のデザイナーが数十個にわたるマップの作成を行っている. このマップ作成にあたっては, マップの地形配置の自然さを考慮するだけでなく, このマップ上で戦う軍勢同士の戦力バランスの調整も重要な課題となるため, これは一般に負担の大きい作業となる. そのため, 本研究では計算機によって同ゲームジャンルにおけるマップの自動生成を試みる. まず自然な地形配置を実現するために, 大規模な 3D ゲームにおける先行手法を参考にしたエージェントベースの生成法を流用する. 次に, 適切な戦力のバランス調整のために, 各ユニット (駒) の耐久値をパラメータとした初歩的なヒルクライム法による最適化を試みる. ターン制戦略ゲームのプラットフォームである『TUBSTAP』上で提案手法の実装を行い, マップの最適化が行われた事を確かめた. また生成されたマップの評価を別の観点から行うため, 最適化の過程に関与していないコンピュータプレイヤー同士を戦わせて戦績を観察した.

1. はじめに

ゲームやパズルのコンテンツの自動生成は広く研究されている. この種の研究は人間のデザイナーによるマップやパズルの生成の手間を省き, より少ない手間でゲームやパズルの楽しさを充実させることを目指している.

しかし, 迷路やパズルの自動生成は進んでいるがターン制戦略ゲームにおけるマップの自動生成はあまり着手されていない分野である. ターン制戦略ゲームとは将棋やチェスと同様に複数の駒を用い, 「マップ」と呼ばれる多種多様な初期盤面からターン毎に駒を動かしていきお互いに駒を倒していくことで勝敗を決めるゲームである.

ターン制戦略ゲームでは製作者がそのゲームタイトル一つにつき制作する必要があるマップの数は数十個程度になるが, プレイヤーを満足させるには様々な要素を考える必要があり手間がかかる. そこで本研究では自動生成のプラットフォームに TUBSTAP[1]を選び自動生成手法の実装を行う. このゲームでは合計で 6 種類のマップ地形(以後, 「属性」と呼ぶ)がある.

ターン制戦略ゲームのマップの自動生成の分野において田口[2]の研究がある. この研究はマップをランダムで生成, 評価してマップの生成方法に工夫を加えることでマップの評価やターン超過数に関してより詳しい分類を行うというものだった. しかし, 問題点として生成されたマップが不自然で美的景観に優れないという課題があった. 本研究では, マップの属性ごとに自動生成のアルゴリズムを作成し, 美的景観を志向したマップの自動生成を試みる.

更にターン制戦略ゲームでは, マップの地勢的な自然さだけでなく, その上で戦うユニット同士の戦力のバランスも重要であるため戦力バランスの最適化を試みる.

最適化にあたって今回は, 各ユニットの HP のみをパラメータとし, そのマップで強い AI プレイヤーと弱い AI プレイヤーが戦った際の勝率を評価値として, 単純なヒルクライ

ム法により最適解の探索を行う.

本研究で自動生成されたマップの景観に対しての評価はアンケート, 戦力バランスに対しての評価は勝率による分析で行う.

2. 関連研究

コンテンツの自動生成の分野では数独や迷路など様々な研究がされてきた. この様な自動生成は制作する人間の負担を緩和するために研究されており, 迷路では棒倒し法[3]や壁伸ばし法[3], 穴掘り法[3]などの様々な手法が考えられている.

ターン制戦略ゲームのマップの自動生成の分野においては田口[2]の研究があった. この研究はマップをランダムで生成, 評価し, マップの生成方法にいくつかの工夫を加えることでマップの評価に関してより詳しい分類を行うというものであった. しかし, 問題点として生成されたマップが不自然で美的景観に優れないという欠点があった.

本研究は美的景観を志向する方法として Jonathon ら[4]の研究を参考にした. この研究は一面を海としたマップの上に, 陸を作るエージェントや山の高さを滑らかにするエージェント, 砂場を作るエージェント, 谷を作るエージェント, 川を作るエージェントを活動させる, エージェントベースの手法により自然な景観の地形をデザインにする手法を提案している. 地形をデザインする手法には他に Thomas ら[5]の研究がある. この研究は, 都市のモデリングを NetLogo というエージェントベース用環境を用いて行う, 人の移住や商業, 産業, 公園, 道路の分布をルールベースに基づいて開発していく手法を提案している.

また, 本研究では計算機により生成されるコンテンツの評価を自動で行うが, その評価の方法として, 麻雀牌を利用した上海というパズルのインスタンス自動生成の先行研究のやり方を参考にしている. この先行研究では 2 組の AI プレイヤーによる解答成功率を利用しており, 本研究のマッ

1 佐世保工業高等専門学校.

プの自動評価もこの研究の評価法を用いている。

3. 対象問題

3.1 ターン制戦略ゲーム(TBS)

ターン制戦略ゲーム (Turn-Based Strategy, 以下 TBS[6]と呼ぶ) は多数の駒を用いて複数のプレイヤーが争う形式のゲームである。このゲームは大まかには将棋やチェスのように盤上の駒同士を戦わせるルールだが以下の特徴がある。

- ・ 1 ターンに複数の駒を動かせる。
- ・ 盤上の駒は HP と呼ばれる数値を持ち、これがゼロになると盤から取り除かれる。
- ・ 駒は隣接する敵駒への攻撃で HP を減らす。
- ・ ゲームのマスには地形が設定され、進入不能などの効果を持つ場合がある。

TBS で対戦を行うための初期盤面をマップといい、様々な種類がある。マップのデータには、盤の大きさ、駒の数、駒の種類、駒の HP、マスの地形の種類が記されている。

図 1 に TBS マップの例を示す。

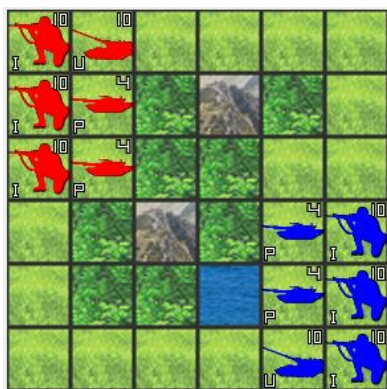


図 1 TBS マップの例

3.2 TUBSTAP

このゲームは 2 人のプレイヤーがそれぞれ操作可能な駒を持ち、駒を動かして倒し合いをするゲームである。そのようなゲームとしては将棋やチェスが古典的だが、このゲームはいくつか異なる点がある。以下異なる点について将棋を基準に記していく。

- 将棋では 1 回に動かせる駒は 1 つだが、このゲームでは全ての自軍の駒を動かせる。
- 将棋では王将を取れば勝ちだが、このゲームでは相手の全ての駒を倒すのが目的となる。
- 将棋では上に重なることで相手の駒を取るが、このゲームでは「近接して攻撃する」「移動せずに遠隔から攻撃する」ことで相手に損害を与える。
- 将棋では必ず 1 回で相手を倒せるが、このゲームでは、駒ごとに耐久度 (HP) が設定されており、それを減らしていく。HP が 0 になるとその駒を倒せる。
- 将棋は 9×9 の盤上で、決まった駒配置から開始するが、このゲームでは「マップ」と呼ばれる多くの盤サ

イズの初期配置があり、そのうち一つを選んでゲームを開始する。

- 将棋は (成りを考えなければ) どのマスもルール上、その性質に区別はなく、駒の移動に制約も加えないが、このゲームのマスには道路・平原・森・山・海などの属性があり、移動に制約が加わるなど、戦果に影響が出る。
- 将棋はどんな駒がどんな駒を倒すこともできるが、このゲームでは駒ごとに相性があり、体力 (HP) に与えられるダメージに影響がある。
- 将棋のような「成り」「持ち駒」はない。

次にゲームの詳細を記す。

- 盤面と地形の属性

図 2 にマップの属性を示す。左から順に、平原、海、森、山、道路、陣地を表す。



図 2 マップの属性

平原: 移動は用意で防御に向かない。

海: 航空ユニットのみ移動できる。

森: 車両は移動困難だが、防御に適している。

山: 地上ユニットは歩兵のみ移動でき防御に適している。

道路: 移動は容易だが、防御には向かない。

陣地: 移動が容易で防御にも適している。

- ユニット (駒)

このゲームには F, A, P, U, R, I と略称される 6 種類のユニット (駒) が存在する。

それぞれに相性があり、移動できる範囲も異なる。

航空ユニット:

F 戦闘機 : 移動範囲の広い対空ユニット。A に強く、R に弱い。

A 攻撃機 : 対地ユニット。P・U に強く、F・R に弱い。

地上ユニット:

P 戦車 : 対地ユニット。R, U に強く、A に弱い。

U 自走砲 : 唯一の遠距離攻撃ユニット。範囲内の地上ユニットを一方向的に攻撃できる。

R 対空戦車: F, A に強い対空ユニットだが、対地攻撃も可能。

I 歩兵 : 歩兵以外への攻撃力は期待できない。

図3に相性の例を示す。矢印が向いているユニットへの攻撃は相性が良く、大きなダメージを与えられる。

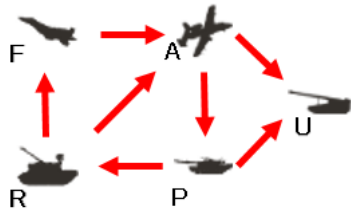


図3 ユニットの相性

- HP
1～10の間で設定される。
- 勝利条件
相手のユニットの全滅または、ターン超過による判定で決まる

4. 手法

4.1 マップ生成手法概要

本研究では、Jonathonら[4]の研究でのエージェント手法を参考に、平坦な地形に様々な侵食等の作用を与えるエージェントを動作させることで、地形の自然なニュアンスを保持したまま美的景観に優れたマップの生成を試みる。以下に自動生成の手法を記す。

- ① マップ初期化(海化)
- ② 海岸線作成(陸地作成)
- ③ 山作成
- ④ 川作成
- ⑤ 森作成
- ⑥ 道路作成
- ⑦ 道路作成補助
- ⑧ 陣地作成

この手法では、川や森や道路を毎回決まった手順で配置するため景観が画一的になるリスクはあるが、それぞれの手順のパラメータの調節やオン・オフによって景観にそれなりの多様性は確保できる。4.1.1～4.1.8節では各手法の内容について述べる。

4.1.1 マップ初期化(海化)

マップのサイズを指定し、その全てを海にする。図4はマップ初期化を行った縦10×横15のマップである。

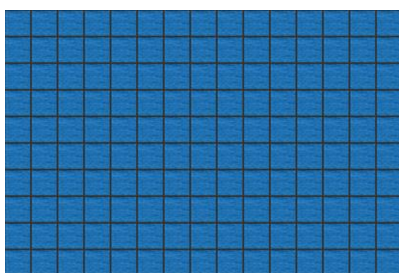


図4 マップ初期化

4.1.2 海岸線作成(陸地作成)

ランダムな海を平原にし、そこから上下左右で隣接したランダムなマスで平原にする。進んだ先から上下左右ランダムなマスで平原にする。これを指定回数まで繰り返させることで海の上に陸地を作成する。

図5は海岸線作成を行った縦10×横15のマップである。

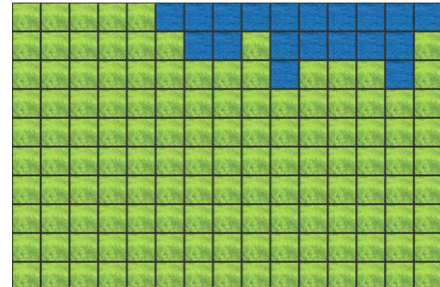


図5 海岸線作成

4.1.3 山作成

ランダムな平原を山にして海を除外した隣接したランダムな方向を山にする。次に山にする方向は $+45^\circ \sim -45^\circ$ にする。この処理を一定回数まで繰り返させることで山脈を作成する。もしマップの外に出た場合や、ランダムな位置から海に隣接した場合、同じ手順で山を作成する。

図6は山の作成を行った縦10×横15のマップである。図6の赤印で囲まれた箇所が山作成の始点である。紫印で囲まれた箇所は山の終点となっている。図右下の紫印で囲まれた箇所からマップの外に出ると判断され、図左上の赤印で囲まれた箇所から新しい山を作成している。



図6 山作成

4.1.4 川作成

現実で川は山から海に向かって流れる事が多いので、海と山をつなぐ川をマップ上に作成する。ランダムな山と海のマスを選びその2点の位置関係を取得する。選ばれた海のマスから見て八方位どの方向に選ばれた山があるかを調べる。分かった方向から $+45^\circ \sim -45^\circ$ の3方向のいずれかに隣接したランダムなマスで海にする。次からは先程海にしたマスから見て八方位どの方向に選ばれた山があるかを調べる。分かった方向から $+45^\circ \sim -45^\circ$ の3方向から隣接したランダムなマスで海にする。これを繰り返す。選ばれた山にたどり着いた場合や他の山に接触した場合にループを終了し川の作成を終える。

図7は川作成を行った縦10×横15のマップである。図

7の赤印で囲まれた箇所がランダムで選ばれた海のマス(川の始点)である。また、図7の紫印で囲まれた箇所がランダムで選ばれた山のマス(川の終点)である。この2点をつなぐように平原が海に上書きされている。

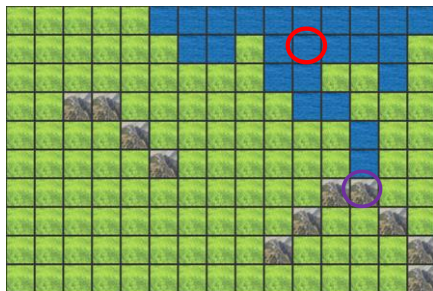


図7 川作成

4.1.5 森作成

現実では山の周辺は森に覆われる事が多いため、山の周囲に森を描画する。マップの各マスからみて周辺8方向に何個山が存在するかを調べる。周辺に2個以上山が存在し、そのマスが平原だった場合、そのマスを森にする。これを全てのマスで行うことで森を作成する。この手法によって山の周辺にある程度の森を作成することができる。

図8は森作成を行った縦10×横15のマップである。図8の赤印で囲まれた箇所は周辺8箇所のうち2箇所に山があるため森となっているが、紫印で囲まれた箇所は周辺8箇所のうち2箇所に山があるもののその箇所が海のため森に上書きされていない。

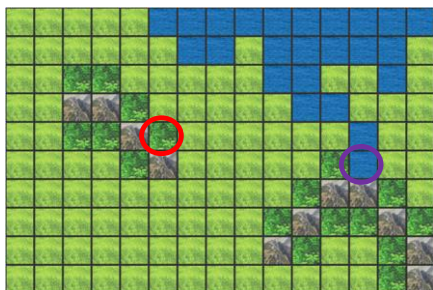


図8 森作成

4.1.6 道路作成

現実の道路は平野部に優先して敷かれる事が多いため、平野の2点を道路で結ぶ。マップ全てのマスでそのマスの周辺8箇所の平原の数を取得する。はじめに道路にするマスを最も周辺の平原の数が多かったマスにする。複数あった場合は複数の中からランダムなマスをはじめに道路にする。次に先程道路にしたマスの周辺8箇所の各マスにおける周辺8箇所の平原が最も多いマスを道路にする。複数あった場合はその中からランダムなマスを道路にする。次からは8箇所ではなく方向を+45° ~ -45° に絞り同じ処理を行う。決められた数を道路にするか、できなくなるまでループさせることで道路を作成する。

図8は道路作成を行った縦10×横15のマップである。

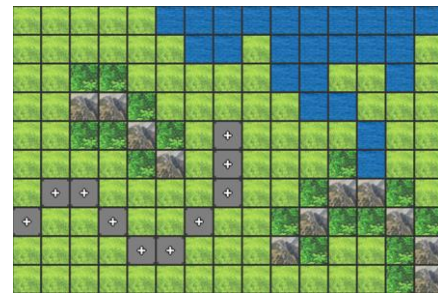


図8 道路作成

4.1.7 道路作成補助

道路の断続的な斜めの隣接を補正する機能を作成した。図9は道路作成補助を行った縦10×横15のマップである。マップ全てのマスでそのマスが道路だった場合斜め4箇所のうちいずれかに道路が存在する場合、そのマスと斜めのマスを直角でつなげるようなマスを道路にする。

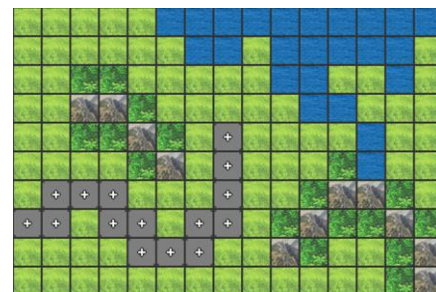


図9 道路作成補助

4.1.8 陣地作成

道路の始点と終点それぞれの周辺一箇所で平原のランダムなマスを陣地にする。図10は道路作成補助を行った縦10×横15のマップである。

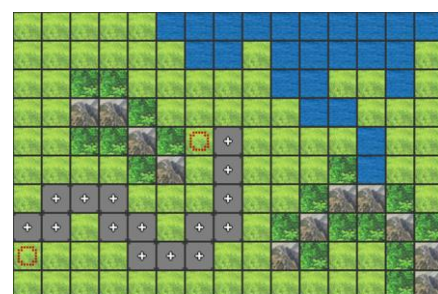


図10 陣地作成

4.2 マップ戦力の最適化

上述の方法で生成されたマップは、ある程度自然な景観になるかもしれないがユニットを配置したとき戦力のバランスが適切でない可能性がある。そこで以降は兵力を配置してからその戦力差を調整する最適化を行う。

4.2.1 最適化するパラメータ

戦力バランスを最適にするために調整できるパラメータは各ユニットの兵種や位置など多数あるが、本研究ではひとまずユニットのHPのみを調整可能なパラメータとする。

4.2.2 評価指標

最適化中のマップの評価は、そのマップ上で強い AI と

弱い AI が戦った場合の勝率差によって行う。戦力のバランスが取れたマップであれば、強い AI はどちらのチームの側で戦うかに関わらず、弱い AI に勝つはずである。これを踏まえ、強い AI と弱い AI で赤・青の陣営を交代しながら対戦させた場合の、先攻陣営における強い AI の勝利数を $w_{S,F}$ 、後攻陣営における強い AI の勝利数を $w_{S,S}$ 、先攻陣営における弱い AI の勝利数を $w_{W,F}$ 、後攻陣営における強い AI の勝利数を $w_{W,S}$ としたときに、 $(w_{S,F} + w_{S,S} - w_{W,F} - w_{W,S})$ をそのマップの基本的な評価値とする。

4.2.3 戦力バランスの調整手法

最適化の流れは、少し丁寧に探索を広げるものの、単純なヒルクライム法をベースとしている。以下に具体的な手順を示す。

- ① 4.1.節の方法で生成されたマップに、適当に種類と位置を決めたユニットを各陣営 HP10 で配置する
- ② どれか1つのユニットの HP を1に上書きする
- ③ 4.2.2 節で示した方法で評価値を取得する。
- ④ そのユニットの HP を1ずつ増やし、HP10 までの各場合における評価値を記録する
- ⑤ HP1~10 の場合のうち評価値が最も高くなった HP 値でそのユニットの HP を確定する
- ⑥ 全てのユニット分、上記②から⑤の手順を繰り返す
- ⑦ 上記②から⑦の処理を、HP の変化量が一定値以下になるか、ループの回数が一定を超えるまで続ける。

5. 実験

5.1 景観の評価

5.1.1 実験方法

佐世保高専電子制御工学科 20 名の被験者に協力してもらいアンケートを用いた実験を実施した。被験者の年齢は 19 から 22 歳であり、男性 19 名、女性 1 名である。

実験の手順は以下の通りである。

- (1) 被験者に TUBSTAP のマップの属性について説明する。
- (2) GoogleForm で作成したアンケートを配布し、40 種類のマップを一通り見てもらう。
- (3) それぞれのマップに関してマップが自然であるかどうかを 1~5 の 5 段階で評価してもらう。
- (4) 他にコメントのある被験者には自由記述欄にその内容を書いてもらう。

評価に用いるマップのサイズは全て縦 10×横 15 マスのマップで行う。マップは大きく分けて 3 種類ある。ランダム生成されたマップ、人の手で作られたマップ、提案手法による生成マップ自動生成（以降：自動生成マップ）である。自動生成は何種類かに分かれる。自動生成は何種類かのパラメータ設定を用いる。各マップ属性の多さや少なさ、生成のされ方が違ってくる。以下に自動生成されたマップの一例を図 11 に示す。

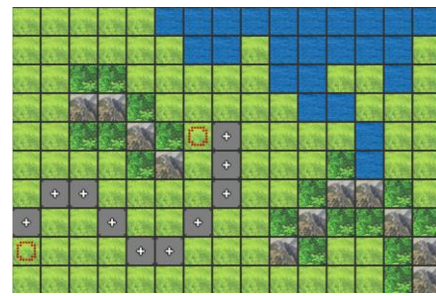


図 11 (20,130,10,10,OFF)

図 17 の () 内の数値の意味はそれぞれ、(海, 平原, 山, 道路, 道路作成補助なし)に関するパラメータである。各数字はマップ作成時での各地形のおおまかなマスの数を指定する。あるいは道路の斜めの隣接を補正する処置の有無を区別する。

実験に用意したマップには以下のものがある。

- 人間作成マップ：人間がデザインした物 5 種類
- ランダムマップ：ランダムに生成された物 5 種類
- 自動生成マップ：(20, 130, 10, 10, OFF) 3 種類
(20, 130, 10, 10, ON) 3 種類
(75, 75, 10, 10, OFF) 3 種類
(130, 20, 10, 10, OFF) 3 種類
(20, 130, 0, 20, OFF) 3 種類
(20, 130, 20, 0, OFF) 3 種類

解りやすさのために、(20, 130, 10, 10, OFF)を「陸多め」、(20, 130, 10, 10, ON)を「陸多め(補助)」, (75, 75, 10, 10, OFF)を「陸海バランス」、(130, 20, 10, 10, OFF)を「海多め」、(20, 130, 0, 20, OFF)を「陸多(山無)」, (20, 130, 20, 0, OFF)を「陸多(道無)」と表記する。

5.1.2 景観実験結果

図 12 に例として、パラメータ「陸多め」のマップ景観に対するアンケート結果をグラフで示す。

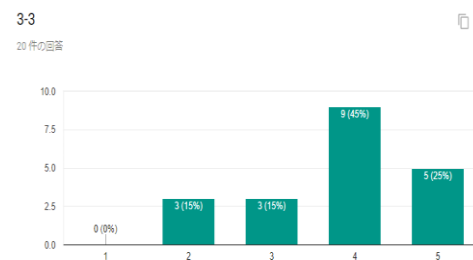


図 12 陸多めのグラフ

自動生成マップは各パラメータに対して 3 種類のマップがあり、それぞれに図 12 のような結果がある。つまり、20 人×各パラメータ (3 マップ) の計 60 のデータがある。この 60 のデータで 95%信頼区間を求めてそのパラメータを評価する。表 1 に各パラメータにおいて、95%信頼区間を用いた結果を示す。

表 1 景観実験結果(カッコ内は 95%信頼区間)

生成手法	パラメータ	評価値
ランダム生成		1.41(±0.127)
手動生成		4.29(±0.178)
自動生成	陸海バランス	3.22(±0.308)
	海多め	2.58(±0.289)
	陸多め	3.90(±0.276)
	陸多め(補助)	3.97(±0.288)
	陸多(山無)	3.55(±0.318)
	陸多(道無)	3.98(±0.283)

表 1 をみると、ランダム生成が他の値と大きく離れて悪い値となっている。最も良い値は手動生成となっている。一方自動生成の値は各パラメータによって多少の差がある事がわかる。しかし、全てのパラメータにおいてランダム生成より大きく値が高いが、手動生成より少し値が低い点が共通している。

5.2 戦力最適化

5.1 で生成したマップに適当にユニットを配置して戦力の最適化を施す。今回は陸多め(補助)のマップ(図 13)と陸海バランス(図 14)、海多め(図 15)の 3 種類のマップを用いて実験した。

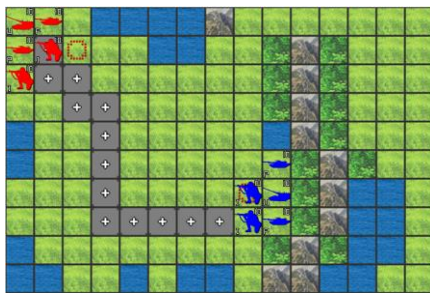


図 13 陸多め(補助)マップ

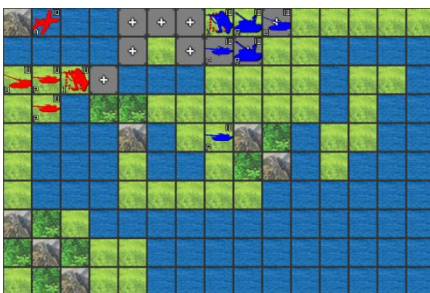


図 14 陸海バランスマップ

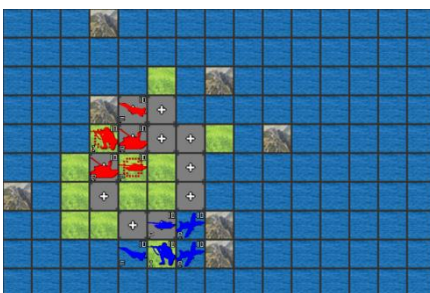


図 15 海多めマップ

5.2.1 戦力最適化実験方法

4.2 の手法で調整されたマップに対し、最適化を適用した。評価用の強い AI として TUBSTAP 同梱の MUC-T をシミュレーション数 2000 で、弱い AI として sample を使用した。そして最適化のループを、ユニットの HP 総量の変化量が 5 未満になった時点で打ち切った。

5.2.2 戦力最適化実験結果

表 2 から 4 に各マップの評価値の推移を示す。評価値は強い AI の勝利数から弱い AI の勝利数を引いた値である。総対戦数は 5000 なので、評価値は最大で 5000、最小で -5000 となる。調整されるパラメータである HP 値の変化量が 5 を下回れば最適化のループを打ち切っており、全て順調に 5 ループ程度で最適化が終了している。そしてある程度適切な戦力バランスに収束する事が確認できた。

表 2 陸多め(補助)最適化結果

状態	評価値	前回との HP 変化量
最適化 1 回目	3761	
最適化 2 回目	3846	10
最適化 3 回目	3761	6
最適化 4 回目	3741	7
最適化 5 回目	3637	1

表 3 陸海バランス最適化結果

状態	評価値	前回との HP 変化量
最適化 1 回目	3478	
最適化 2 回目	4082	24
最適化 3 回目	4232	21
最適化 4 回目	4244	22
最適化 5 回目	4129	4

表 4 海多め最適化結果

状態	評価値	前回との HP 変化量
最適化 1 回目	-571	
最適化 2 回目	1249	19
最適化 3 回目	2221	8
最適化 4 回目	1624	2

6. 考察

6.1 景観について

自動生成は全てのパラメータ設定の組合せにおいてランダム生成より良く、手動生成より悪い評価値となった。しかし、手動生成に届くほど良い結果のパラメータの組み合わせもあれば、大きく劣るものもある。全体的に平原が多いパラメータ設定が良い結果になっていることが表 1 からわかる。これは、平原が少ないと道路や山を配置しづらく窮屈なマップが生成されやすく景観が悪くなったためだと考えられる。

また道路作成補助に関しては表 1 の結果から補助を行った場合のほうが良い結果となっている。このことから、ある程度は道路を配置するときは、斜めに隣接させる際に多少の補間を施した方が良さそうだと考えられる。

陸多め(補助)と陸多め(道無)の結果を比べてみると、僅差ではあるが陸多め(道無)の結果が優れている。この2つのパラメータの大きな違いは道路の存在の有無である。このゲームには6つの属性があるが、人工物は道路と陣地だけである。特に道路はある程度の数をマップで占める割に、周りの属性は人工物ではない。この人工物と自然物のマップ属性の混在が評価に影響しているものと考えられる。そして、結果からマップに人工物がなくても景観の自然さの印象に対して大きなマイナスでは無いことが解った。

6.2 戦力最適化について

戦力の最適化をほどこしたマップ上で本当に互角にゲームができるのかを別の観点から確かめるために、最適化し終わったマップ上で既存の同じ AI 同士で対戦し結果を確かめる。この結果として勝敗の回数に大きな差がなければ、確かにマップの戦力は互角になったと考えられる。その結果が表5である。使用した AI は4種で、sample と military と M-UCT と M3Lee である。各 AI のペアごとの、対戦回数は AI につき100回である。この表の値は4つの AI ごとの赤青勝利数差分を足した数字である。例えば sample 同士の対戦で先攻70勝-後攻30勝、military 同士で先攻45勝-後攻55勝、M-UCT 同士で先攻50勝-後攻50勝、M3Lee 同士で先攻50勝-後攻50勝となれば、引き算の結果は絶対値とし、 $40 + 10 + 0 + 0$ で50となる。この値が小さい程、より互角に遊びやすいマップといえる。

表5 AI ごとの差分総和

状態	AI ごとの差分総和		
	陸多め(補助)	陸海バランス	海多め
無調整	81	193	335
最適化1回目	129	384	44
最適化2回目	119	271	331
最適化3回目	178	158	214
最適化4回目	138	297	296
最適化5回目	65	339	

6.2.1 陸多め(補助)

このマップは標準的なマップでユニットも対称だったため無調整でも勝利数に大きな差はなかった。表5より、多少のノイズはあるものの最適化を繰り返すことで初期 HP が最適化され5回目では無調整時より良い結果となっている。

6.2.2 陸海バランス

このマップはユニットの数と兵種を非対称に配置しており、無調整時には攻撃機がある先攻陣営がどの AI においても有利だった。

HP の変化量が収束するまで繰り返し最適化を行ったが、表5の結果は悪化した。これは航空ユニットの存在が要因だと考えられる。

攻撃機に敵対する陣営には対空砲を2つ配置していた。しかし対空砲が殲滅されると攻撃機に一切反撃できなくなり航空機に殲滅されている。無調整時はすべての HP が10

であるため対空砲が戦車の攻撃を1回は耐えるが調整後は HP が10より減っているため場合によっては対空砲が瞬殺されるため悪い結果になり、航空ユニットの存在がこのゲームにおいて影響が大きいことがわかった。

6.2.3 海多め

このマップは狭い陸地に多種多様な兵種を詰め込んだマップだった。表5の結果より、最適化を繰り返すことで無調整時よりは結果が改善されたことがわかる。しかし、改善された陸多め(補助)に比べ依然勝利数に大きな偏りがあった。これは、このマップにおいて先攻陣営が極端に有利なためだと考えられる。実際、対戦を観察しているとマップが狭く先攻陣営が1ターン目から先制攻撃を仕掛け、戦いを有利に運んでいた。先制攻撃を受けると強い AI でも弱い AI に勝てなくなるため最適化が上手く動作しなかったと考えられる。

7. 将来の課題

航空ユニットや先制攻撃などの要因が絡むと最適化のパラメータが HP だけでは上手く機能しないことがわかった。今後、最適化のパラメータにユニットの配置場所変更や兵種の変更等のパラメータを加えることでより多くのマップへの対応が見込める。

また、最適化の手法を検討し直し、遺伝的アルゴリズムなどを組み込むことで性能が改善する可能性考えられる。

参考文献

- [1] “ターン制戦略ゲーム 学術用基盤プロジェクト” . <http://www.jaist.ac.jp/is/labs/ikeda-lab/tbs/>, (参照 2021-05-25).
- [2] 田口紫織. ターン制ストラテジーゲームにおける強さの異なる AI プレイヤを用いた Map 自動生成. 佐世保高専 平成30年度卒業論文, 2019
- [3] “迷路自動生成アルゴリズム” . <http://www5d.biglobe.ne.jp/stssk/maze/make.html>, (参照 2020-01-20).
- [4] Jonathon Doran and Ian Parberry. “Controlled Procedural Terrain Generation Using Software Agents” , IEEE Transactions on Computational Intelligence and AI in Games Volume: 2, Issue: 2, 2010
- [5] Thomas Lechner, Pin Ren Ben, Watson Craig and Brozefski Uri Wilenski. “Procedural Modeling of Urban Land Use” , Proceeding SIGGRAPH '06 ACM SIGGRAPH 2006 Research posters Article No. 135, 2006
- [6] 佐藤直之, 藤木翼, 池田心, ターン制戦略ゲームにおける局面評価値構成のための局面分割および単純化ゲームのオフライン木探索, The 20th Game Programming Workshop 2015, 2015