芸術科学会論文誌 Vol. 20, No. 2, pp. 139 – 148

長期的目標達成を考慮した ユーティリティベースゲーム AI に関する研究

古川真帆 [1)](学生会員)　阿部雅樹 [2)](正会員)　渡辺大地 [2)](正会員)

1) 東京工科大学大学院バイオ・情報メディア研究科　2) 東京工科大学メディア学部

A Study on Utility Based Game AI Considering Long-Term Goal Achievement

Maho Furukawa[1)]　**Masaki Abe**[2)]　**Taichi Watanabe**[2)]

1) Graduate School of Bionics, Computer and Media Science, Tokyo University of Technology
2) School of Media Science, Tokyo University of Technology

g3119032f9 @ edu.teu.ac.jp, abemsk @ edu.teu.ac.jp, earth @ gamescience.jp

概要

ゲーム AI におけるユーティリティベース AI の中には，目的地への到達などの最終的な目標達成を保証する評価基準を持つゴールベース指標と，局所的な評価のみにより行動を決定し，最終的な目標達成を保証しない非ゴールベース指標がある．ゴールベースと非ゴールベースを足し合わせて両方の考慮を行った場合，ゴールベースの特性が失われて目的達成の保証ができなくなるという問題がある．本研究は，ゴールベースと非ゴールベースを両方同時に利用しても，最終的な目標をできるだけ迅速に達成する行動の実現を目的とする．本研究では，離散ラプラシアンを用いた盛り土関数を使って局所的極値解を回避し，最終的な目標達成を保証する手法を提案する．例として，ゴールベースを目的地到着，非ゴールベースを追跡エージェントからの回避行動とし，追跡エージェントを回避しつつ目的地到達を迅速に行うという AI を盛り土関数を用いて実現した．また，盛り土関数の調整係数は AI の体力や停滞度の数値を使用しファジー推論を用いて動的に変更するようにした．

Abstract

Among utility-based AI in game AI, there are two types of indicators: goal-based indicators, which have evaluation criteria that guarantee the achievement of the final goal, such as reaching the destination, and non-goal-based indicators, which determine actions based only on local evaluations and do not guarantee the achievement of the final goal. When goal-based and non-goal-based indices are added together and both are taken into account, the problem arises that the goal-based property is lost and the guarantee of goal achievement is lost. The purpose of this research is to realize a behavior that achieves the final goal as quickly as possible, even if both goal-based and non-goal-based are used simultaneously. We propose a method that uses a fill function with a Discrete-Laplacian to maintain the law of transition and guarantee the achievement of the final goal. As an example, we assume that the goal base is the arrival at the destination and the non-goal base is the avoidance of the tracking agents, and the AI that avoids the tracking agents and reaches the destination quickly is realized using the fill function. The adjustment coefficients of the fill function were changed dynamically using fuzzy reasoning based on the numerical values of the AI's fitness and stagnation.

1　はじめに

近年のゲームにおいて，ゲーム AI という高度な技術が重要視されている [1][2]．ゲーム AI とは，ゲーム内の人工的に作り出された知能のことである．その 1 つに，キャラクター AI というプレイヤーが操作しない仲間や敵などのキャラクターに搭載する人工知能があり，『FINAL FANTASY XV』や『ドラゴンクエスト X』など多くのゲーム作品で用いられている．キャラクター AI に関する研究は数多く行われており，例えば，藤井ら [3] はアクションゲーム「Infinite Mario Bros.」において，人間の生物学的制約を課した機械学習によって人間らしい振る舞いを自律的に獲得する AI を作成した．また，星野ら [4] は模倣学習の手法を用いてコンピュータが操作するキャラクターの行動パターンを試合ごとに拡張し，成長する格闘ゲームキャラクター AI を作成した．

キャラクター AI は多様な手法が用いられるが，そのうちの 1 つとしてユーティリティベース AI がある．ユーティリティベース AI とは，現在の状態に対し何かしらの評価基準により評価値を算出し，より良い評価となる行動を選択する手法である．例えば，危険度を評価基準とし，敵と近いほど危険であるとした場合，敵との距離を評価値とする方法が考えられる．この評価値を「コスト」と呼称する．キャラクターが移動可能なマップをグラフとして表現した場合，各ノードにおけるコストを算出し，キャラクターはよりコストが低くなるノードに移動することで，危険を避けることができる．これらの評価基準やコストは，AI の制作者が自由に設定することができるものである．図 1 は，ユーティリティベース AI のイメージ図である．図中の赤い部分がコストが高い領域であり，エージェントは赤い部分を回避しながら移動することが望ましい行動となる．

以降，本論文では経路グラフ上を移動可能なキャラクター AI の行動を議論の対象とし，経路グラフ上の各ノードにはコスト値が設定されるものとする．

ユーティリティベース AI は局所的な状況判断には優れた性能を発揮するが，長期的な目標を達成する保証ができないという問題がある．一方で，長期的な目標を達成するためのゲーム AI 手法としてゴールベース AI がある．『F.E.A.R』『Deus Ex: Human Revolution』『Middle-earth: Shadow of Mordor』『サカつく DS』など，多くのゲーム作品でゴールベースが用いられている [5][6]．また，Studiawan ら [7] はゴールベース AI の一種であり，ゴールと初期条件の間をアクションの連鎖でつないで行く方法である GOAP 手法をリアルタイムストラテジーゲームに適用する手法について述べている．ゴールベース AI は目標達成を保証するという意味で有用な手法であるが，行動途中の動的な状況変化には対応できないという問題がある．

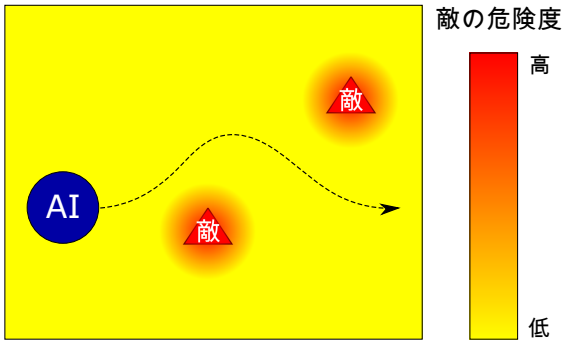


図 1　ユーティリティベース AI のイメージ

ゴールベース AI の典型例として，目的地への経路探索がある．経路探索では，マップ上の各経路点 (ノード) から目的地までの道程距離をコスト値として算出し，現在位置からコストが低くなる隣接地点に移動することを繰り返すことにより最短経路を導出する手法であるが，このコスト値をユーティリティベースにおける評価関数と想定することで，ゴールベース AI をユーティリティベース AI の一種と見なすことができる．本論文では，このようなゴールベース AI 由来による指標を「ゴールベース指標」と呼称するものとする．ゴールベース指標は，キャラクターがコスト値が低くなる箇所へ移動することを繰り返すことにより，必ず目的地にたどり着くことができるという特性を持つ．

ユーティリティベース AI はスカラー値による評価関数であることから，複数の評価指標を足し合わせて同時に考慮することが可能である．このことから，任意のユーティリティベース AI による評価指標とゴールベース AI 由来の評価指標を足し合わせることにより，長期目標達成と局所的状況判断の両方の特性を併せ持つ AI を実現することができると考えられる．

しかしながら，このような指標を足し合わせて両方の考慮を行った場合，ゴールベース指標の特性が崩れ，目標達成の保証ができなくなるという問題がある．例えば，

目的地に向かって経路探索をしているエージェントが周囲を敵に囲まれて袋小路の状態になった場合，敵を避け続けて目的地に辿り着かなくなる可能性が考えられる．

そこで，本研究ではゴールベースと非ゴールベースを両方同時に利用しても，最終的な目標をできるだけ迅速に達成する行動の実現を目的とし，離散ラプラシアンを用いた盛り土関数を使ってゴールベース指標の特性を復帰し，最終的な目標達成の割合を高くする手法を提案する．具体的には，ゴールベースを目的地到着，非ゴールベースを追跡エージェントからの回避行動とし，追跡エージェントを回避しつつ目的地到達を迅速に行うという AI を盛り土関数を用いて実現した．

我々は，本研究の初期成果を NICOGRAPH2020 にて発表した [8]．これに対し調整係数の動的変更機能を追加し，AI による行動がより状況に応じて適切となるよう拡張を行った．また，その動的変更が有効となるようにシミュレーション用ゲームルールの拡張を行い，逃亡エージェントの体力値の設定や追跡エージェントによる攻撃を追加した．

2 提案手法

2.1 ゲーム内エージェントの前提条件

本研究で扱うマップおよびエージェントの想定を以下のように定める．

- 移動可能なエージェントとして 1 体の「逃亡エージェント」と，逃亡エージェントを追跡する複数の「追跡エージェント」がある．
- マップ内には「目的地」が 1 箇所設定されており，逃亡エージェントは，追跡エージェントを回避しつつ目的地に到達することを目指す．
- エージェントが移動可能な領域は離散グラフによって表現されるものとする．
- エージェントの移動は離散的なものではなく，エージェントごとに設定されている時間経過と共に設定速度に従って移動する．
- マップ内のあるノード i に対し，位置を $\mathbf{P}_i$ と表す．
- あるノード i の隣接ノード集合を $N(i)$，$N(i)$ の要素数を $|N(i)|$ と表す．

2.2 ユーティリティベース AI

ゲーム中でエージェントが移動できる領域において，位置 $\mathbf{P}_i$ におけるコスト評価関数 $T(\mathbf{P}_i)$ が定義できるものとする．ここで，$T(\mathbf{P}_i)$ はエージェントの「望ましさ」をスカラー値で表したものであり，小さいほど望ましいものと定義する．例えば，エージェントに目的地があり，目的地に近いほど望ましいのであれば，目的地に近づくほど $T(\mathbf{P}_i)$ は小さい値となる．また，同様に敵との距離が離れている方が望ましい場合，敵に近い位置では $T(\mathbf{P}_i)$ の値は増大することになる．また，$T(\mathbf{P}_i)$ は時間が経過し状況に変化があれば，その評価値も動的に変化する．先述の例では目的地が変更となった場合や，敵が移動した場合などがそれにあたる．

現在のエージェント位置がノード i であるとき，次に移動する先のノードは i の隣接ノードのうち T がもっとも小さくなるノード，つまり

$$\underset{j \in N(i)}{\operatorname{argmin}} T(\mathbf{P}_j) \tag{1}$$

となる．

2.3 ゴールベース指標と非ゴールベース指標

本研究では，ユーティリティベース AI で用いる評価関数指標を「ゴールベース指標」と「非ゴールベース指標」の 2 種類に分類する．

「ゴールベース指標」とは，評価関数が単峰性を持ち，最小値以外の極小値が存在しないような指標のことである．ゴールベース指標では，評価関数が減少する方向にエージェントが移動することで，必ず最小値に到達できることが保証される．このような指標では，評価関数が最小となるようなパラメータは最急降下法 [9] 等を用いることで容易に求めることができる特性がある．経路探索においては，各経路点からゴールまで到達する距離を「コスト値」と設定する手法が多く，このコスト値を指標に用いると，ゴールに近いほどコスト値は下がるため，経路探索コスト指標はゴールベース指標と言うことができる．

一方，「非ゴールベース指標」は「ゴールベース指標」を満たさない指標のことである．例えば，出発地から障害物を避けつつ目的地を目指すような状況を考える．ゴールから現在値までの直線距離を指標とした場合，迂回をしなければ目的地へ到達できないような状況では，ゴールへの過程で指標が増加してしまうことがありえる．こ

のような指標は「非ゴールベース指標」と言える．

前述したように，ユーティリティベース AI では，複数の指標を足し合わせることが可能であるが，そのときにゴールベース指標と非ゴールベース指標の両方を足してしまった場合，評価関数の単峰性が崩れてしまうため，最急降下方のような単純なアルゴリズムは一切利用できなくなるという問題が発生する．例えば，経路探索によって求めた目的地への最適経路をゴールベース指標とし，追跡エージェントとの距離を非ゴールベース指標とした場合，これらを足し合わせると評価関数の局所的極値が発生してしまい，目的地への到達が保証されなくなることがある．図 2 は，逃亡エージェントが追跡エージェントに囲まれ，目的地への到達が困難な状況を表したものである．

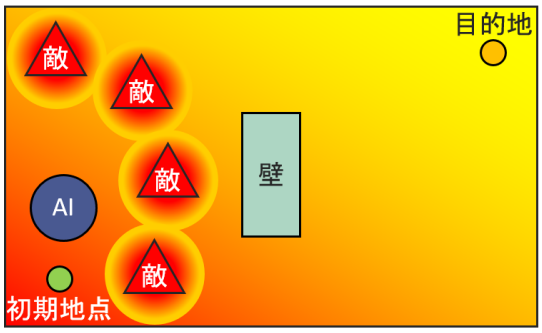


図 2　目的地への到達が困難な状況

本手法は，このような状況において，盛り土関数を用いてゴールベース指標と非ゴールベース指標を足し合わせた評価関数に対して単峰性を復帰し，局所的極小値から脱却して真の最小値状態（目的地）に到達することを目的とする．

2.4　手法概要

ここでは，目的地への移動距離によるコスト関数を $C_g(\mathbf{P}_i)$ とおく．図 3 は，$C_g(\mathbf{P}_i)$ をスカラー場としてコスト値に応じて着色した様子である．目的地への移動距離によるコスト値は，目的地に近づく方向に単調減少するため，$C_g(\mathbf{P}_i)$ はゴールベース指標と言える．

また，時刻 t における敵からの距離によるコスト関数を $C_u(\mathbf{P}_i)$ とおく．$C_u(\mathbf{P}_i)$ も $C_g(\mathbf{P}_i)$ と同様にスカラー場として捉えることができる．図 4 は，$C_u(\mathbf{P}_i)$ のスカラー場を表している．$C_u(\mathbf{P}_i)$ は，複数の敵が存在するとスカラー場として局所的な極値が存在するため，

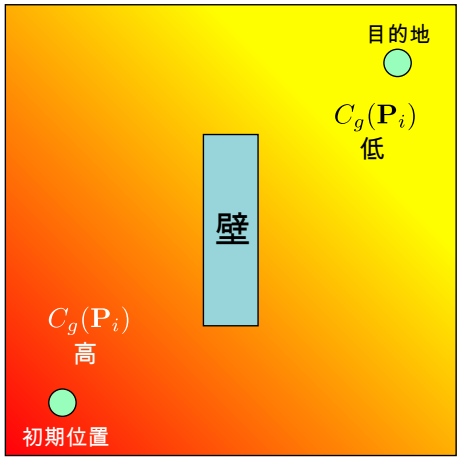


図 3　$C_g(\mathbf{P}_i)$ のスカラー場

非ゴールベース指標と言える．

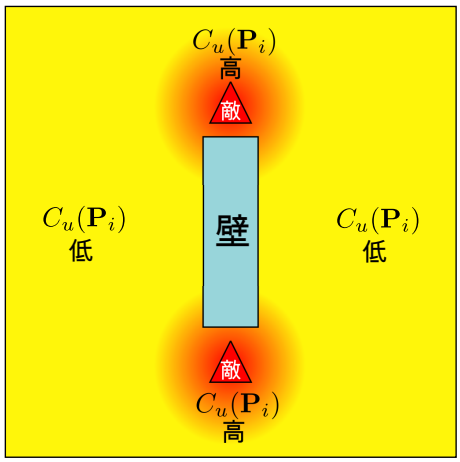


図 4　$C_u(\mathbf{P}_i)$ のスカラー場

ここで，両方の指標を考慮する場合，時刻 t におけるコスト評価関数 $T(\mathbf{P}_i)$ を以下のようにすることが考えられる．

$$T(\mathbf{P}_i) = k_g C_g(\mathbf{P}_i) + k_u C_u(\mathbf{P}_i) \qquad (2)$$

ここで k_g, k_u は調整係数である．k_g, k_u の初期値はユーザーが任意の数値に設定して調整を行う．k_u の動的調整については 2.6 節で説明する．k_g は固定値とした．この評価関数は両方の指標を元にコストが構成されるが，前述したように単峰性が保たれないため，$T(\mathbf{P}_i)$ は「非ゴールベース」となってしまい，目的地到達が保証されなくなる．

本手法は，$T(\mathbf{P}_i)$ の単峰性を保つため，盛り土関数

$C_f(\mathbf{P}_i)$ と調整係数 k_f により

$$T(\mathbf{P}_i) = k_g C_g(\mathbf{P}_i) + k_u C_u(\mathbf{P}_i) + k_f C_f(\mathbf{P}_i) \quad (3)$$

とし，$T(\mathbf{P}_i)$ が単峰性となるように補正する．k_f は k_g, k_u と同様に，ユーザーが任意の数値に設定して調整するものとし，現状は固定値である．$C_f(\mathbf{P}_i)$ については 2.5 節で説明する．

この $T(\mathbf{P}_i)$ を用いることで，エージェントは目的地への到達を保証できる．

2.5 離散ラプラシアンによる盛り土関数

これまで述べてきたように，非ゴールベース指標となった評価関数の場合，最終的な目的値への到達が保証されないという問題がある．その理由は，コスト値が減少する方向に移動を続けた際，目的地とは異なる局所的極値に至ってしまい，目的地への到達経路が不明となるからである．

この状況を避けるためには，局所的極値のコストを増加することで，エージェントが局所的極値に向かうことを回避することが望ましい．このコスト増加を盛り土関数を用いて実現し，追加するコスト値は離散ラプラシアンを用いて算出するものとした．

ラプラシアン ∇^2 は，本来は連続ベクトル場 ϕ において以下の式によって定義されるスカラー値である [10]．

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} \quad (4)$$

この値は，流れ場においては流体が増加する箇所で正，減少する箇所で負の値が出るものである．ラプラシアンは，画像処理や 3DCG など様々な分野で広く使われている [11][12][13][14]．しかしながら，本手法で扱っているエージェントの移動空間はグラフネットワークによる離散空間であり，各経路点においてラプラシアンを算出することはできない．

そこで，本手法では以下の式によって (4) 式によるラプラシアンを離散的に算出するものとした．

$$\nabla^2(\mathbf{P}_i) = \frac{1}{|N(i)|}\sum_{j\in N(i)} (T(\mathbf{P}_i) - T(\mathbf{P}_j)) \quad (5)$$

離散ラプラシアンの模式図を図 5 に示す．例えば，コスト値が 40 の場所の場合は，周囲のコスト値が 50 の場所からの差分である +10 と，30 の場所との差分である-10 を合計し，結果的に離散ラプラシアン値は 0 となる．

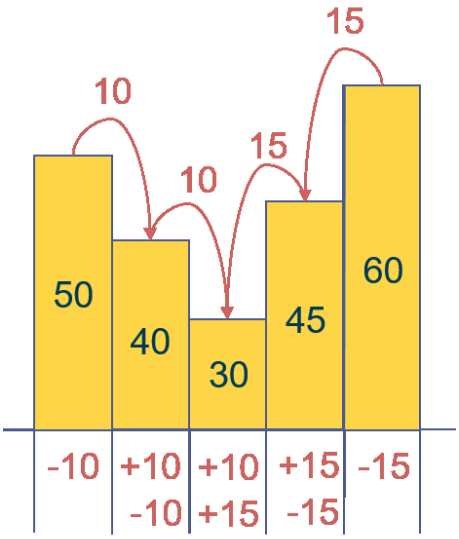


図 5 離散ラプラシアンの概要図

各経路点は，盛り土関数用の補正値を保持するものとし，初期値を 0 とする．各経路点の離散ラプラシアン値を算出し，一定の設定値を超えた場合に現在の補正値に追加する．その補正値を返値する関数を盛り土関数 $C_f(\mathbf{P}_i)$ とした．

なお，補正値は時間経過により減衰するものとし，その減衰の割合は調整できるものとした．減衰を施す理由は，第一に減衰しない場合にこの補正値はマップ全体で著しく増加してしまい，C_g や C_u の効果が次第に影響を持たなくなってしまうこと，第二に既に各エージェントが移動した後の古い情報が誤判断を生じてしまうことを避けるためである．

減衰は以下の式によるものとした．

$$C_f{}^t(\mathbf{P}_i) = (1-\epsilon)\left(C_f{}^{t-\Delta t}(\mathbf{P}_i) + \nabla^2(\mathbf{P}_i)\right) \quad (6)$$

ここで，$C_f{}^t(\mathbf{P}_i)$ は時刻 t における $C_f(\mathbf{P}_i)$ の値を示し，Δt はシーン更新時間間隔である．本研究では，$\Delta t = \frac{1}{60}$ 秒 として扱った．また，ϵ は正の微少量を意味し，本研究における実装では 0.01 を用いた．

2.6 調整係数の動的変更

逃亡エージェントに体力を付与し，体力が少なくなるほど目的地への到達より追跡エージェントを避ける方を優先するといった，逃亡エージェントの状態による調整係数 k_u を動的に変更するようにした．

追跡エージェントが複数いるような状況のとき，逃亡エージェントが単純に追跡エージェントを回避する行動を繰り返し続けると，同じ場所を右往左往するような行動となってしまい，結果的に複数の追跡エージェントに囲まれる可能性が高くなることが想定される．そのため，逃亡エージェントがある程度停滞状態にある場合は，状況打開のために追跡エージェント回避よりも目的地到達の優先度を上げることが，結果的に有効となることが多い．この行動を実現するため，逃亡エージェントの停滞の度合いによって係数 k_u を動的に調整するよう実装を行った．

本研究では，逃亡エージェントの時刻 t の時点での体力値 H^t を以下の法則になるよう実装した．

- 上限値を 1，下限値を 0 とし，初期値を 1 とする．
- 追跡エージェントと接触した場合，一定量が減少する．ただし，接触時から一定時間は接触しても減少しない (つまり無敵状態になる) ものとした．
- 追跡エージェントと接触していない状態で，かつ H^t が上限値でない場合は一定量回復する．

時刻 t の時点での停滞度 L^t は，逃亡エージェントの現在位置 $\mathbf{P}^t$ と一定時間 l 前の時点での位置 $\mathbf{P}^{t-l}$ との距離を反映するものであり，以下の式で算出した．

$$L^t = \frac{lV - \left|\mathbf{P}^t - \mathbf{P}^{t-l}\right|}{lV}. \tag{7}$$

ここで V は逃亡エージェントの速さを表す．結果として，L^t は上限を 1，下限を 0 として，大きいほど停滞した状態であることを意味する．本研究では，$l = 10\Delta t$ として算出した．

k_u を動的に変更する手法は，これら 2 つの数値を使用しファジー推論 [15][16] を用いた．基本的な方針としては，「体力値が大きい場合」と「停滞度が大きい場合」は追跡エージェントからの回避を軽視し，目的地到達を優先するというものである．重心法によるファジー推論関数 Z により

$$k_u = U\left(1 - Z(H^t, L^t)\right) \tag{8}$$

を算出し，調整係数 k_u の自動調整を実現した．U は k_u の初期設定値であり，k_u の上限値を意味する．Z の具体的な算出方法は付録 A に示す．

2.7 本研究における実装

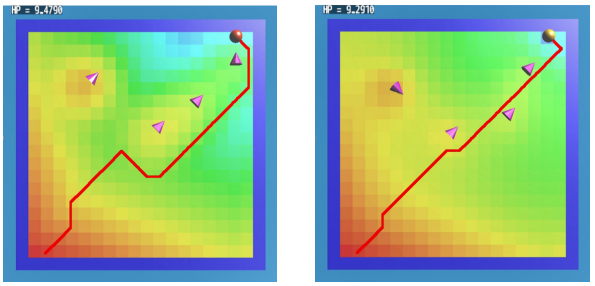
$C_g(\mathbf{P}_i)$ は，経路探索手法のダイクストラ法 [17] を用いて，ゴールベースの経路部分を実現した．また，非ゴールベース指標を表す $C_u(\mathbf{P}_i)$ は，

$$\sum_e \max\left(D_{max} - |\mathbf{P}_i - \mathbf{P}_e|^2,\ 0\right) \tag{9}$$

という式で算出した．ここで D_{max} は C_u の適用範囲を設定する調整値，$\mathbf{P}_e$ は追跡エージェントの位置を表す．

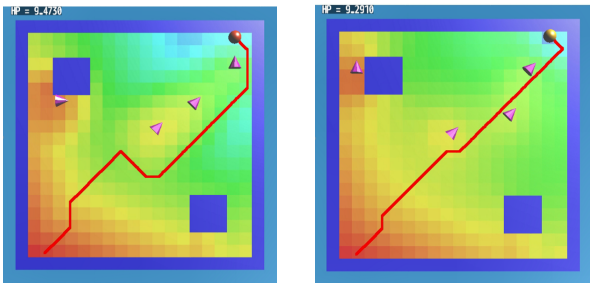
3 評価

3.1 検証結果



$k_f = 0$ $k_f = 0.3$

図 6 マップ 1 の検証結果

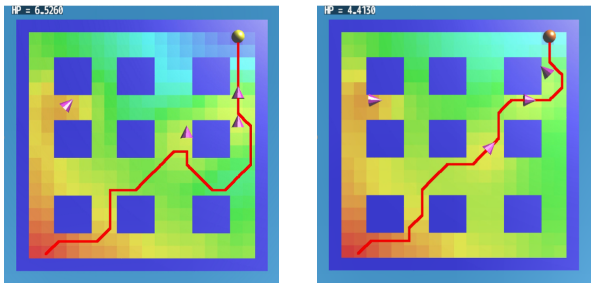


$k_f = 0$ $k_f = 0.3$

図 7 マップ 2 の検証結果

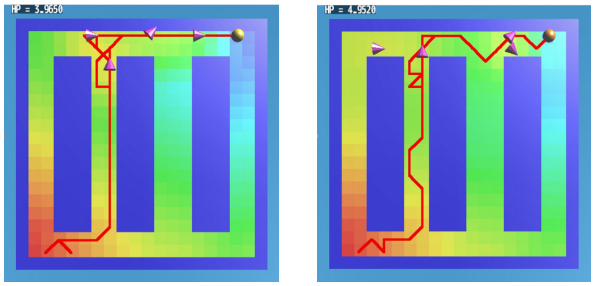
図 6 ～ 10 は 5 種類のマップに対し本手法を適用したものである．各図の左側が調整係数 k_f を 0 とし，盛り土関数の効果を無効としたものであり，右側は $k_f = 0.3$ を設定して盛り土関数による単峰化補正を有効としたものである．全ての実行結果において，2.6 節で述べた k_u の動的調整を有効としている．

全てのマップで左下を初期地点，右上が目的地と設定しており，マップ中の青い部分はエージェントが通過で



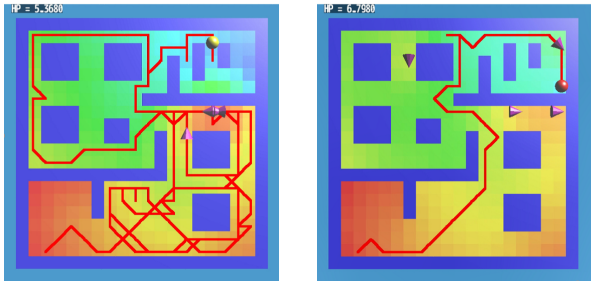
$k_f = 0$　　$k_f = 0.3$

図 8　マップ 3 の検証結果



$k_f = 0$　　$k_f = 0.3$

図 9　マップ 4 の検証結果



$k_f = 0$　　$k_f = 0.3$

図 10　マップ 5 の検証結果

きない場所となっている．球体が逃亡エージェント，紫色の円錐が追跡エージェントとなっており，逃亡エージェントは追跡エージェントに近づくほど赤色になるように設定している．逃亡エージェントの体力 (HP) については，本来の最大値は 1 であるが，閲覧性をよくするために 10 倍した数値を表記している．体力は最大値の状態から実行を開始する．さらに，逃亡エージェントが追跡エージェントのある一定範囲内に近づくと，体力が 0.1 減るという追跡エージェントによる近接攻撃を表現した．また，コストを可視化するためにマップ内でコストが高い箇所が赤，低い箇所を青となるよう擬似カラー [18] にて彩色した．

また，これらの実行結果に加え，2.6 節の動的調整を行わず，k_u を固定値とした場合の検証も実施した．表 1 は，各パターンの目的地到達にかかる時間 (秒) と追跡エージェントとの衝突回数を集計した結果である．表中の「×」は 120 秒間実行しても目的地に到達できず，実験を中止したケースである．

表 1　各マップでの検証結果集計

		動的調整あり		動的調整なし	
	k_f	時間	衝突	時間	衝突
マップ 1	0	18.8	1	18.7	1
	0.3	16.0	1	16.1	1
マップ 2	0	18.7	1	18.7	1
	0.3	16.1	1	16.1	1
マップ 3	0	22.4	4	22.5	4
	0.3	18.9	6	19.3	6
マップ 4	0	34.5	7	72.2	10
	0.3	32.7	6	62.9	6
マップ 5	0	×	×	×	×
	0.3	29.8	4	×	×

3.2　分析

マップ 1, 2, 3 における逃亡エージェントの経路は，盛り土関数を適用しない場合はやや大回りとなることに対し，盛り土関数を有効とした場合には最低限の回避行動に留め，より迅速な目的地到達を実現していることがわかる．マップ 4,5 のように経路が狭い箇所が多い場合は目視による特性は見出しづらいが，到達時間は全般的に減少していることから，マップの特性にかかわらず本手法は目的地到達が迅速となる傾向があると考えられる．

一方，マップ 3 の衝突回数は補正が有効な方が多数となった．補正が行われることで C_u による影響が薄れ，結果的には敵の回避よりも目的地到達を優先するような行動となったことが原因と考えられる．このような行動は本手法においては想定通りと言えるものであるが，実運用の際には注意が必要な点である．

また，k_u の動的調整については，単純な構造のマップについてはあまり効果が見られないが，複雑な場合には著しい到達速度の向上が見られた．また，マップ 5 のケースでは盛り土関数と動的調整の両方を有効としなければ目的地に到達ができなかった．このようなことから，それぞれの手法は片方だけでもそれなりに効果を持つ一方で，マップが複雑な状況においては併用することでより効果を発揮すると言える．

4 まとめ

本研究は，ユーティリティベース AI の中のゴールベース指標と非ゴールベース指標を足し合わせて両方の考慮を行った場合，ゴールベース AI の特性である目的達成の保証性が失われるという問題に着目した．本研究では，離散ラプラシアンを用いた盛り土関数を使って評価関数の単峰性を復帰し，最終的な目標達成の可能性を高める，あるいはより迅速な達成を実現する手法を提案した．ゴールベース指標にダイクストラ法によるコストマップを使用し，非ゴールベース指標に追跡エージェントからの危険度コストを使用した．また，調整係数は AI の体力や停滞度の数値を使用しファジー推論を用いて動的に変更するようにした．以上により算出されたコスト値に従って，局所的なユーティリティベース AI と目標達成を保証するゴールベース AI を融合した特長を持つ AI を実現した．

検証の結果，単純なマップにおいては盛り土関数が迅速な目的地到達に有効であるが，係数動的調整についてはあまり影響が見られないことが判明した．一方で，複雑なマップにおいては盛り土関数と動的調整の両者が性能向上に有効であることや，併用することでより高い性能を発揮することがわかった．

一方で，追跡エージェントを必要以上に回避してしまい目的地到達が遅れてしまう場面や，追跡エージェントの包囲からの脱却が適切に行えない場面が見受けられた．今後，より適切な指標や係数調整について検討し，性能の向上に努めたい．

参考文献

[1] 三宅陽一郎. ゲーム AI 技術入門. 技術評論社, 2019. ISBN: 978-429710828-1.

[2] 三宅陽一郎. ディジタルゲームにおける人工知能技術の応用の現在. 人工知能学会論文誌, Vol. 30, No. 1, pp. 45 – 64, 2015.

[3] 藤井叙人, 佐藤祐一, 若間弘典, 風井浩志, 片寄晴弘. 生物学的制約の導入によるビデオエージェントの「人間らしい」振舞いの自動獲得. 情報処理学会論文誌, Vol. 55, pp. 1655 – 1664, 2014.

[4] 星野准一, 田中彰人, 濱名克季. 模倣学習により成長する格闘ゲームキャラクタ. 情報処理学会論文誌, Vol. 49, pp. 2539 – 2548, 2008.

[5] Jeff Orkin. Three states and a plan:the ai of f.e.a.r. `http://alumni.media.mit.edu/~jorkin/`. GDC, 2006, 参照 : 2020/08/05.

[6] 安藤毅. 「サカつく」のサッカー試合 AI システム. `https://cedil.cesa.or.jp/cedil_sessions/view/379`. CEDEC, 2010, 参照 : 2020/11/24.

[7] Restuadi Studiawan, Mochamad Hariadi, and Surya Sumpeno. Tactical planning in space game using goal-oriented action planning. *JAREE (Journal on Advanced Research in Electrical Engineering)*, Vol. 2, pp. 5 – 9, 05 2018.

[8] 古川真帆, 阿部雅樹, 渡辺大地. ゲーム AI における評価関数による最終目標達成保証と局所問題回避の両立の実現. *NICOGRAPH2020*, pp. 87–92, 2020.

[9] 茨木俊秀. 最適化の数学. 共立出版, 2011. ISBN: 978-4320015654.

[10] 山崎昶. 法則の辞典. 朝倉書店, 2006. ISBN: 978-4254101973.

[11] Peter-Pike Sloan, Jan Kautz, and John Shyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans and SIGGRAPH*, Vol. 21, No. 3, pp. 527 – 536, 07 2002.

[12] Bruno Vallet and Bruno Lvy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum and Wiley*, Vol. 27, No. 2, pp. 251 – 260, 2008.

[13] Rhaleb Zayer, Christian Rssl, Zachi Karni, Hans - PeterSeidel. Harmonic guidance for surface deformation. *Euro Graphics*, Vol. 24, No. 3, 2005.

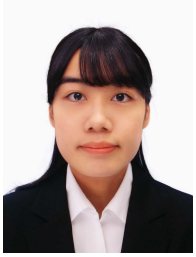
[14] Patrick Prez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans and SIGGRAPH*, Vol. 22, No. 3, pp. 313 – 318, 2003.

[15] David M. Bourg, GlennSeemann 著, 株式会社クイープ訳. ゲーム開発者のための AI 入門. オライリー・ジャパン, 2005. ISBN: 4-87311-216-8.

[16] MatBuckland 著, 松田晃一訳. 実例で学ぶゲーム AI プログラミング. オライリー・ジャパン, 2007. ISBN: 978-4-87311-339-5.

[17] B. コルテ, J. フィーゲン. 組合せ最適化. シュプリンガー・ジャパン, 2009. ISBN: 978-4431100218.
[18] 渡部秀文, 斎藤隆文. 二色塗り分けによる擬似カラー表示を用いた情報可視化. VC/GCAD 合同シンポジウム 2000 予稿集, pp. 119–124, 2000.

古川 真帆



2019 年東京工科大学メディア学部卒業．2021 年同大学大学院修士課程バイオ・情報メディア研究科メディアサイエンス専攻修了．芸術科学会会員．

阿部 雅樹

2008 年東京工科大学メディア学部卒業．2010 年同大学大学院修士課程バイオ・情報メディア研究科メディアサイエンス専攻修了．2016 年より同大学メディア学部実験助手，現在に至る．コンピュータグラフィックスやゲーム制作に関する研究に従事．芸術科学会会員．

渡辺 大地

1994 年慶応義塾大学環境情報学部卒業．1996 年慶応義塾大学政策・メディア研究科修士課程修了．2016 年岩手大学工学研究科デザイン・メディア工学専攻博士後期課程修了．博士 (工学)．1999 年より東京工科大学メディア学部講師．2017 年より同准教授，2020 年より同教授，現在に至る．コンピュータグラフィックスやゲーム制作に関する研究に従事．芸術科学会，情報処理学会，画像電子学会，人工知能学会会員．

付録 A　ファジー推論

ファジー推論とは，複数の入力からファジー値を算出する手段であり，最小最大法，代数積加算重心法，簡略化推論法など様々なものがある．本研究では，このうち代数積加算重心法を用いた．本章では，この手法の計算方法について述べる．なお，ここでは本文中の式 (8) における 2 つのスカラー値 H^t, L^t を入力値として想定する．

ファジー推論においては，複数のメンバーシップ関数を事前に規定する必要がある．典型的なメンバーシップ関数は，Bad(B)，Normal(N)，Good(G) の 3 つの関数を用いるものであり，それぞれ以下のような式となる．

$$B(x) = \begin{cases} -2x+1, & (0 \leqq x < \frac{1}{2}) \\ 0 & (\frac{1}{2} \leqq x \leqq 1) \end{cases} \tag{10}$$

$$N(x) = \begin{cases} 2x, & (0 \leqq x < \frac{1}{2}) \\ -2x+2 & (\frac{1}{2} \leqq x \leqq 1) \end{cases} \tag{11}$$

$$G(x) = \begin{cases} 0, & (0 \leqq x < \frac{1}{2}) \\ 2x-1 & (\frac{1}{2} \leqq x \leqq 1) \end{cases} \tag{12}$$

本研究においても，メンバーシップ関数は上記関数を用いた．

次に，各メンバーシップ関数における最小値を求める．これを B_m, N_m, G_m とし，入力値を H^t, L^t とすると

$$\begin{cases} B_m = \min\left(B\left(H^t\right), B\left(L^t\right)\right) \\ N_m = \min\left(N\left(H^t\right), N\left(L^t\right)\right) \\ G_m = \min\left(G\left(H^t\right), G\left(L^t\right)\right) \end{cases} \tag{13}$$

となる．

次に，各メンバーシップ関数において先ほどの値 B_m, N_m, G_m より下側の部分の面積を求め総和を取る．つまり，

$$\begin{aligned} F(x) = &\min\left(B(x), B_m\right) \\ &+ \min\left(N(x), N_m\right) + \min\left(G(x), G_m\right) \end{aligned} \tag{14}$$

として $F(x)$ を定義した上で，

$$\int_0^1 F(x)dx \tag{15}$$

を求める．

最後に，全体のグラフの重心，つまり全面積を二等分する箇所を算出する．具体的には，

$$\int_0^\alpha F(x)dx = \frac{1}{2}\int_0^1 F(x)dx \tag{16}$$

となる α を求める．この α が式 (8) 中の $Z(H^t, L^t)$ の値となる．

ファジー推論において，厳密解にはあまり有用な意味はなく，ほとんどの場面においては区分求積法などの近似値計算で実用上問題ない．本手法で用いたアルゴリズムは以下の通りである．

1. 各推論グラフの面積を求め，全部を合計する．
2. 合計値の半分を求める．
3. 値 0 から Δx ごとに順番に各推論グラフの面積を追加し，合計値の半分に達したら処理を終了する．