すごろく型ボードゲームAIのための フェーズベース強化学習フレームワーク

山田 歩果[1,a)]　魏 廷翰[1,b)]

概要：本稿では，桃太郎電鉄に代表されるすごろく型ボードゲームを対象とした，後続研究の基盤となる強化学習フレームワークを構築する．対象ゲームでは，サイコロ，カード，物件購入など，ゲームの進行段階（フェーズ）によって選択可能な行動の種類と数が大きく変化する．そこで，ゲーム進行をフェーズとして定式化し，各状態で有効な行動を action mask として提示する Gymnasium 形式の環境を実装した．移動候補は，盤面グラフ上の到達可能駅を列挙する「いけるかな」機能によって生成し，固定長の離散行動へ対応付ける．環境の検証として，サイコロだけで移動する課題を用い，動的計画法による最適方策，貪欲方策，ランダム方策，MaskablePPO を比較した．この課題でも最も近い駅へ向かう貪欲方策は最適ではない．観測をそのまま多層パーセプトロンへ入力した MaskablePPO は 100 ターンで平均 0.66 回しか到着せず，ランダム方策（0.92 回）を下回った．一方，各移動候補を同一のネットワークで評価する方策構造を用いると，10 個の学習 seed すべてで平均 13.26 回到着し，貪欲方策と同等の水準に達した．これらの結果は，可変長の候補集合に適した方策構造が本環境での学習の鍵であることを示す．

キーワード：すごろく型ボードゲーム，強化学習環境，行動マスク，MaskablePPO，ベースライン

A Phase-based Reinforcement Learning Framework for Sugoroku-style Board Game AI

Honoka Yamada[1,a)]　Ting-Han Wei[1,b)]

Abstract: We build a reinforcement learning framework for sugoroku-style board games such as Momotaro Dentetsu, intended as a basis for future research. In these games, the kind and number of available actions change sharply with the phase of the game: rolling dice, playing cards, choosing a destination, or buying property. We formalize the game as a sequence of phases and implement a Gymnasium environment that exposes the valid actions in each state as an action mask. Movement candidates are generated by a reachability search on the board graph and mapped to a fixed set of discrete actions. To validate the environment, we use a dice-only movement task and compare the optimal policy obtained by dynamic programming, a greedy policy, a random policy and MaskablePPO. Even in this task, heading for the closest station is not optimal. MaskablePPO with a standard multilayer perceptron reaches the destination only 0.66 times per 100 turns, fewer than a random policy (0.92). With a policy that scores every movement candidate with the same network, it reaches 13.26 arrivals on all 10 training seeds, on par with the greedy policy. A policy architecture suited to variable candidate sets is thus the key to learning in this environment.

Keywords: sugoroku-style board game, reinforcement learning environment, action masking, MaskablePPO, baselines

1. はじめに

すごろく型ボードゲームでは，プレイヤはサイコロやカードを用いて盤面上を移動し，指定された目的地への到着を目指す．桃太郎電鉄に代表されるこの種のゲームで

[1] 高知工科大学情報学群
School of Informatics, Kochi University of Technology
[a)] 270386f@ugs.kochi-tech.ac.jp
[b)] tinghan.wei@kochi-tech.ac.jp

© 2026 Information Processing Society of Japan

は，停止した駅の種類に応じて所持金の増減，カードの入手，物件購入などのイベントが発生する．そのため，ターン開始時にはサイコロを振るかカードを使うか，移動量が決まった後は到達可能な駅のどれへ向かうか，駅に到着した後は物件を買うかなど，ゲームの進行段階（フェーズ）ごとに性質の異なる判断が求められる．選択可能な行動の種類と数がフェーズによって大きく変わり，さらにサイコロによる確率的な遷移と複数プレイヤの相互作用を含む点で，強化学習の題材として興味深い．

一方，このようなゲームを対象とした強化学習研究を行うには，まずゲームを状態，行動，報酬として定式化し，フェーズごとに異なる行動を一貫した形で扱える環境が必要となる．本研究の目的は，すごろく型ボードゲームを対象とした強化学習研究の基盤となる環境を構築することである．強い AI プレイヤの作成やプレイヤ支援は本稿の直接の目的ではなく，それらは本環境を用いた後続研究の対象と位置付ける．

本稿の貢献は以下の 3 点である．

(1) ゲーム進行をフェーズとして定式化し，各状態で有効な行動を action mask として提示する Gymnasium 形式 [10] の環境を設計・実装した．

(2) 盤面グラフ上の到達可能駅を列挙する「いけるかな」機能により，サイコロの出目によって変化する移動候補を固定長の離散行動へ対応付けた．

(3) サイコロだけで移動する課題について，動的計画法による最適値を求めて貪欲方策が最適でないことを示し，MaskablePPO の学習には候補ごとに共有ネットワークを適用する方策構造が必要であることを明らかにした．

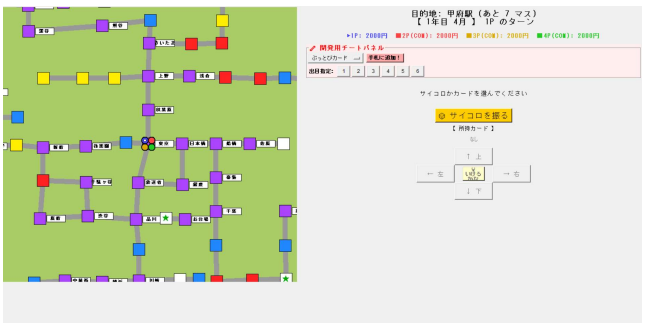


図 1　実装したすごろく型ボードゲームエンジンの画面例

2. 関連研究

物件の売買を含む確率的なボードゲームとして，モノポリーを対象とした強化学習研究がある．Bailis らはモノポリーをマルコフ決定過程として定式化し，Q 学習によるエージェントを構築した [1]．Haliem らは，ルールベースのプレイヤの模倣学習と深層強化学習を組み合わせたエージェントにより，固定方策のエージェントに対して高い勝率を報告している [2]．カタンについては，モンテカルロ木探索による手法 [3] や，異なる形式の入出力を統合するニューラルネットワークを用いた強化学習 [4] が提案されている．これらのゲームはプレイヤ間の交渉を含む点で本研究の対象と異なるが，状態空間が大きく遷移が確率的である点，購入などの経済的判断を含む点は共通しており，状態や行動の表現の設計が性能を大きく左右することが報告されている [2], [4]．

状態によって選択可能な行動が変化する問題では，無効な行動の確率を 0 とする action mask が広く用いられ，その有効性が方策勾配法において検証されている [5]．本研究では，この action mask をフェーズごとの行動集合の表現として用いる．

3. 対象ゲームと定式化

3.1 対象ゲーム

対象ゲームは桃太郎電鉄を参考に研究用に単純化したものである．盤面は 116 駅からなるグラフであり（図 2），駅は物件駅（56），カード駅（21），青駅（15），赤駅（16），カード売り場（4）などに分類される．3 人のプレイヤが順に手番を行い，本稿では学習エージェント 1 人と，目的地までの距離が最も小さくなる方向へ進むルールベースの COM プレイヤ 2 人で対戦する．目的地は物件駅から無作為に選ばれ，エージェントが到着すると新しい目的地が設定される．カードは移動系を中心に 20 種類を実装した．1 エピソードは 100 ターンとする．

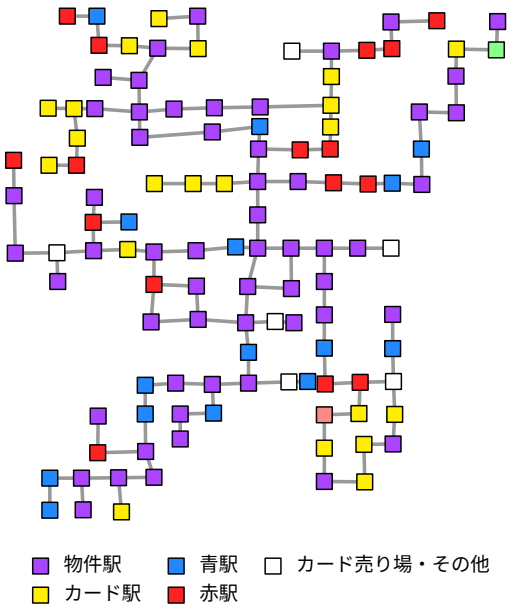


図 2　盤面グラフ（116 駅）．色は駅の種類を表す

3.2 フェーズに基づく定式化

本環境では，ゲームの状態 s に現在のフェーズ $\phi(s)$ を含め，各状態で選択可能な行動集合 $A(s) \subseteq \{0, \ldots, 36\}$ を定

© 2026 Information Processing Society of Japan

める．エージェントは各ステップで観測 o と，$A(s)$ を表す二値ベクトル（action mask）を受け取り，$A(s)$ から 1 つの行動を選ぶ．1 ターンは，ターン開始時の `menu`（サイコロまたはカードの選択），移動先を選ぶ `destination`，駅イベントの処理などの複数のステップからなり，エージェントの手番が終わると COM プレイヤの手番が環境内で処理される．したがって，エージェントから見た遷移はサイコロと COM プレイヤの行動による確率的なものとなる．

この定式化により，行動の種類と数がフェーズによって異なるゲームを，単一の離散行動空間と状態依存の action mask で表現できる．また，フェーズごとに学習対象とするか否かを切り替えられるため，カード使用や物件購入などの判断を段階的に追加する研究が可能となる．

3.3 ゲームの複雑さ

本来のゲームでは，移動先の選択に加えて多くの判断が絡み合う．移動系カードは出目の数や候補駅の集合そのものを変え，物件購入や所持金は到着とは別の目的を生み，3 人のプレイヤが同じ目的地を争う．このような課題には単純な規則で表せる最適行動は存在せず，強化学習による長期的な戦略獲得の題材となる．

4. 環境の実装

4.1 フェーズベースの API

主なフェーズは，通常サイコロまたはカードを選択する `menu`，到達可能な候補駅から移動先を選択する `destination`，カード固有の数値選択や駅選択を扱う `card_select_num` および `card_select_station`，物件駅，カード売り場，借金返済，カード破棄の各処理であり，計 10 フェーズを定義した（図 3）．学習対象としないフェーズでは，終了・復帰操作だけを有効とする．

4.2 観測空間と行動空間

観測は辞書形式で与えられ，ゲーム状態を表す 59 次元の `observation` と，37 次元の `action_mask` から構成される．`observation` の内訳を表 1 に示す．候補が 20 件未満の場合，候補駅 ID は −1，候補距離は 100 で埋める．

目的地までの距離（最短経路の駅数）は環境が盤面グラフから計算して与える．この距離がなければ，エージェントは駅 ID だけから，あらゆる駅の組の間の位置関係を盤面ごとに学習しなければならない．

行動空間は `Discrete(37)` とした．action 0 は通常サイコロまたは各フェーズの終了・復帰操作，action 1 から 16 は手札カード，数値選択，カード破棄，借金返済時の物件売却など，フェーズに応じた操作へ割り当てる．action 17 から 36 は最大 20 件の移動候補またはカード効果による駅候補へ割り当てる．例えば，カードを所持していない `menu` では action 0 だけが有効となり，`destination` では実際に

表 1 観測（`observation`，59 次元）の内訳

添字	内容	次元
0	目的地の駅 ID	1
1	自分の現在駅 ID	1
2	自分の所持金	1
3	現在地から目的地までの距離	1
4–11	手札のカード ID	8
12–15	他の 2 プレイヤの駅 ID と所持金	4
16	現在のフェーズ	1
17	サイコロの出目	1
18	使用中のカード ID	1
19–38	移動候補の駅 ID	20
39–58	各移動候補から目的地までの距離	20

到達可能な候補駅に対応する action だけが有効となる．

4.3 いけるかな機能

「いけるかな」機能は，現在地と移動量を入力として，盤面グラフ上でちょうどその移動量で到達できる駅を列挙する．探索にはメモ化深さ優先探索を用いる．直前に通過した駅へただちに戻る U ターンは禁止する一方，環状線を持つ路線構造へ対応するため，経路中の駅を一律に再訪禁止とはしない．現在駅，直前駅，残り移動量の組を探索状態としてメモ化し，残り移動量が 0 となった駅を集合へ追加した後，駅 ID 順に整列して候補列を生成する．候補の先頭から最大 20 件を action 17 から 36 へ順に対応付け，候補に含まれない action は action mask によって無効化する．これにより，サイコロの出目によって選べる移動先が変化するという性質を，固定長の行動空間で扱える．図 4 に例を示す．

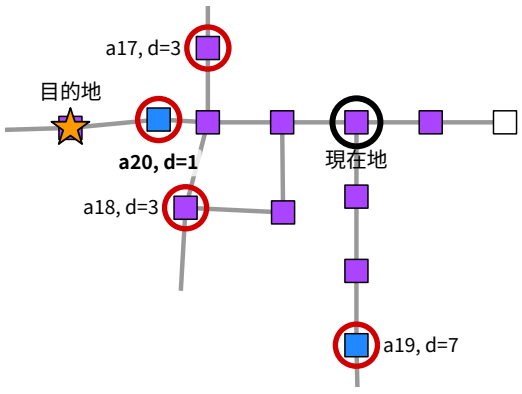


図 4 「いけるかな」機能の例．現在地から出目 3 で到達できる 4 駅が候補となり，駅 ID 順に action 17–20 へ対応付けられる．d は各候補から目的地までの距離（駅数）であり，この例では action 20 が最短候補である

4.4 設定可能な課題

環境は JSON 形式の設定により，初期手札，有効なカードの種類，駅イベントの有無，報酬の各係数を切り替えられる．本稿の移動課題で用いた報酬を表 2 に示す．選択し

© 2026 Information Processing Society of Japan

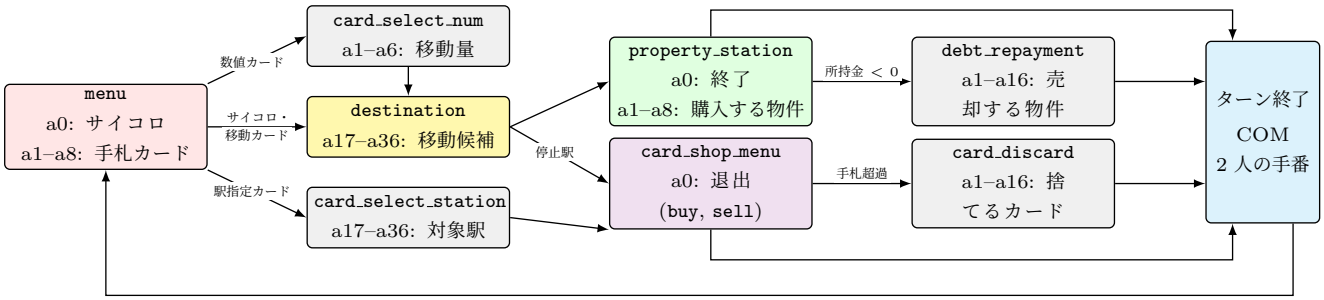


図 3 フェーズの遷移と各フェーズで action mask が有効にしうる行動（an は action n）．青駅・赤駅・カード駅などに停止した場合は，駅イベントの処理後にそのままターン終了となる．ターン終了時に COM プレイヤ 2 人の手番が環境内で処理される．カード駅でのカード入手によっても `card_discard` へ遷移する

た候補から目的地までの距離を d_{sel}，候補中の最短距離を d_{best} とし，最短候補を選んだ場合に報酬 1.0，それ以外では $0.1(d_{\mathrm{sel}} - d_{\mathrm{best}})$ のペナルティを与える．また，到着しなかった移動では目的地までの距離の減少量に 0.1 を掛けた報酬を与え，到着時には報酬 10.0 を与える．学習と評価のためのスクリプト，評価に用いる seed，学習済みモデルは公開リポジトリ[*1]で管理している．

表 2 移動課題の報酬設定

報酬項目	値
最短候補選択報酬	1.0
候補距離差ペナルティ係数	0.1
1 行動当たりの報酬	−0.01
目的地到着報酬	10.0
距離変化報酬係数	0.1

5. ベースライン実験

5.1 移動課題と評価方法

環境の検証として，サイコロ（1–6）だけで移動する課題（以下，移動課題）を用いた．初期手札と有効カードを空とし，カード駅からのカード取得と青駅・赤駅の所持金イベントを無効化した．この課題でのエージェントの判断は，毎ターンの `destination` における移動先の選択だけとなる．

評価は seed 10000 から 10099 の 100 エピソードで行い，各エピソードの開始時に Python 標準乱数，NumPy，環境へ同一の seed を設定した．学習済み方策は決定的に行動を選択する．指標は 1 エピソード（100 ターン）当たりの平均目的地到着回数，および移動先選択において候補中で目的地に最も近い駅を選んだ割合（最短候補選択率）とした．なお，移動先選択の約 5%は候補が 1 つだけの場面である．同一条件の評価を 2 回実行し，エピソード単位の結果が完全に一致することを確認した．

[*1] `https://github.com/RLHS-Lab-KUT/Momotetsu-RL`

5.2 移動課題の難しさと最適方策

移動課題では，目的地に最も近い候補へ向かう貪欲な選択で十分に思えるが，以下の理由からそうとは限らない．

- 目的地に到着するには，出目ちょうどで目的地に止まる必要がある．そのため，目的地により近い駅よりも，次のターンに多くの出目で目的地に止まれる駅の方が有利な場合がある．
- 移動中の U ターンは禁止されており，盤面には行き止まりの駅が 22 駅ある．このため，目的地までの距離が同じ駅でも，次のターンに到達できる駅の集合は大きく異なる．
- 到着すると新しい目的地が無作為に設定されるため，どの駅で到着するかもその後の到着回数に影響する．

移動課題では COM プレイヤと所持金がエージェントの移動に影響しないため，状態を（現在駅，目的地，残りターン数）に縮約でき（$116 \times 56 \times 100$ 状態），動的計画法により 100 ターンでの期待到着回数を最大化する最適方策を厳密に求められる．その結果，最適方策の期待到着回数は 14.32 回であり，同じモデルで評価した貪欲方策の 13.60 回より約 5%多かった．また，（状態，出目）の組の 4.4%では，最適な選択がどの最短候補とも異なっていた．なお，このモデルでは，新しい目的地が COM プレイヤのいる駅を除いて選ばれる点を省略している．

5.3 比較する方策

以下の 5 つを比較した．

- ランダム方策：有効な行動から一様に選択する．
- 貪欲方策：`destination` では目的地までの距離が最小の候補を選び（同点の場合は先頭），それ以外のフェーズでは action 0 を選ぶ．
- MaskablePPO：Stable-Baselines3 および SB3-Contrib の実装 [6], [7] を用い，観測をそのまま入力する．
- MaskablePPO ＋観測正規化：観測の各次元を学習中の移動平均と分散で正規化する（`VecNormalize`）[8]．
- MaskablePPO ＋候補スコアリング：各候補を同一の

© 2026 Information Processing Society of Japan

ネットワークで評価する方策（次項）．
上の 2 つの MaskablePPO の方策には `MultiInputPolicy`（隠れ層 64 ユニット× 2 層）を用いた．

候補スコアリング　標準の方策（図 5(a)）は，観測全体を 1 つの多層パーセプトロンへ入力し，37 個の logit を同時に出力する．各出力は別々の重みを持つため，「i 番目の候補が目的地に最も近ければ action 17+i を選ぶ」という規則を，20 個の候補位置ごとに別々に学習しなければならない．

候補スコアリング（図 5(b)）では，候補 i ごとに特徴 $x_i = (d_i,\ d_{\mathrm{cur}},\ d_i - d_{\mathrm{cur}},\ \text{出目})$ を作る．ここで d_i は候補 i から目的地までの距離，d_{cur} は現在地から目的地までの距離である．そして，すべての候補に同じネットワーク f_θ（隠れ層 32 ユニット× 2 層）を適用し，その出力を action 17+i の logit とする．

$$\ell_{17+i} = f_\theta(x_i), \quad \pi(17{+}i \mid s) = \frac{\exp \ell_{17+i}}{\sum_{a \in A(s)} \exp \ell_a} \tag{1}$$

無効な行動は action mask によって和から除かれる．重み θ はすべての候補で共有されるため，一度学習した評価規則は候補の数や並び順によらず，すべての候補位置で成り立つ．action 0–16 の logit は観測からの線形層で，価値関数は多層パーセプトロン（64 ユニット× 2 層）で計算する．どちらも，観測の各次元を固定の定数で割って縮尺を揃えた上で入力する．図 5(b) の数値は，学習済みのモデル（seed 0）を図 4 の局面に適用した結果である．目的地までの距離が 1 の候補（action 20）の確率は 0.997 である．

主な設定を表 3 に示す．各条件で学習 seed を 0 から 9 とした 10 回の学習を行った．参考として，5.2 節の動的計画法による最適値も示す．

表 3　MaskablePPO の主な設定

項目	値
学習ステップ数	100,352
学習率	3.0×10^{-4}
`n_steps`	1024
バッチサイズ	64
エポック数	10
割引率 γ	0.99

5.4　結果

結果を表 4 に示す．貪欲方策は平均 13.88 回到着し，すべてのエピソードで 1 回以上到着した．これに対し，観測をそのまま入力した MaskablePPO の到着回数は平均 0.66 回であり，10 個の学習 seed すべてでランダム方策の 0.92 回を下回った．観測を正規化すると平均 1.52 回となり，すべての seed でランダム方策を上回ったが，貪欲方策には遠く及ばなかった．候補スコアリングを用いると，10 個の seed すべてで最短候補選択率が 100%となり，平均 13.26 回到着した．貪欲方策との差は，距離が同じ候補が複数ある場合の選び方の違いによる．学習開始時には 10 個中 6 個の seed が最も遠い候補を選ぶ方策（到着 0 回）であったことから，この結果は初期化によるものではなく，学習によって得られたものである．

表 4　移動課題における比較（100 エピソード．MaskablePPO は学習 seed 10 個の平均±標準偏差．最適は動的計画法による期待値）

方策	平均到着回数	最短候補選択率
最適（参考）	14.32	–
貪欲	13.88	100%
ランダム	0.92	30.1%
MaskablePPO	0.66 ± 0.12	41.9 ± 1.6%
＋正規化	1.52 ± 0.27	45.8 ± 1.8%
＋候補スコアリング	**13.26 ± 0.01**	**100.0 ± 0.0%**

なお，観測をそのまま入力した MaskablePPO を約 300,000 ステップまで継続学習した予備実験でも，到着回数は平均 0.66 回であり，改善は見られなかった．

6.　考察

6.1　入力表現と方策構造

本課題では，各候補から目的地までの距離が観測に含まれており，その最小値を選ぶだけで貪欲方策とほぼ同じ性能が得られる．それにもかかわらず，標準的な多層パーセプトロンを用いた MaskablePPO がランダム方策にも及ばなかった原因として，三つの要因が考えられる．

第一に，20 件の候補の特徴を固定長の観測としてまとめて入力するため，「i 番目の候補距離が最小のとき i 番目の action を選ぶ」という対応を，候補位置ごとに別々に学習しなければならない．候補は駅 ID 順に並ぶため，最短候補はどの位置にも現れうる．一方，後方の位置に候補が入ることはまれで，その位置の学習機会は少ない．第二に，駅 ID や所持金は大きさに意味のない値，あるいは桁の大きい値であり，学習を妨げる．観測の正規化によって到着回数が増加したことは，この要因の寄与を示している．第三に，到着には距離 0 の候補を正確に選ぶ必要があり，「比較的近い候補」を選ぶだけの方策では到着回数はほとんど増えない．

候補スコアリングは，すべての候補に同じ評価規則を適用するため，候補の数や並び順によらず，一度学習した規則がすべての位置で成り立つ．この構造の変更だけで，10 個の seed すべてが貪欲方策と同等の水準に達した．すごろく型ゲームでは，サイコロの出目と現在地によって候補の数と内容が毎ターン変わるため，候補ごとの評価を基本とする方策構造が適していると考えられる．なお，本稿の候補スコアリングは環境が与える距離だけを入力としており，その性能は貪欲方策と同程度にとどまる．最適方策と

© 2026 Information Processing Society of Japan

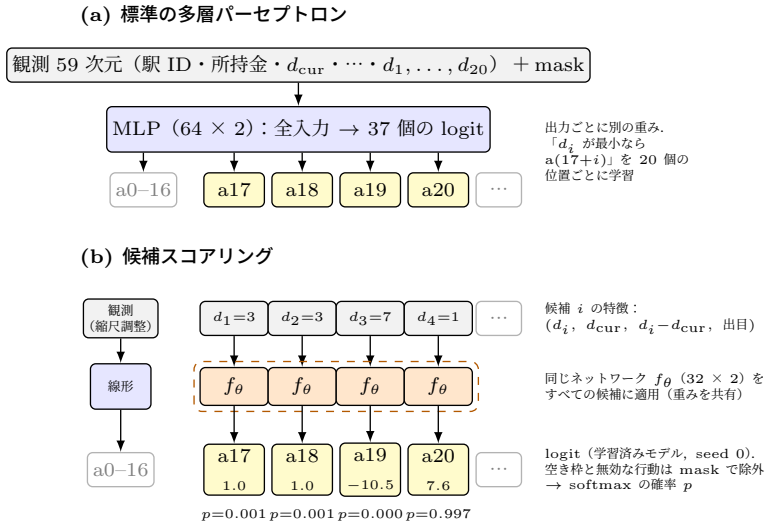


図 5 標準の方策（a）と候補スコアリング（b）．(b) では，同じネットワーク f_θ をすべての移動候補に適用し，その出力をそれぞれの action の logit とする．例は図 4 の局面（$d_{\mathrm{cur}} = 4$，出目 3）

の差（14.32 回と 13.26 回）を埋めるには，次のターン以降の到達可能性を考慮する必要がある．

6.2 報酬設計

本課題の報酬は，到着報酬に加えて最短候補選択と距離変化に基づく補助報酬を含む．これらはポテンシャルに基づく報酬整形 [9] の形をとっていないため，最適方策を変えてしまう可能性がある．実際，観測をそのまま入力した MaskablePPO の 1 つ（平均報酬 13.39，到着 0.85 回）は，ランダム方策（平均報酬 5.06，到着 0.92 回）より報酬が高いにもかかわらず，到着回数は上回らなかった．そのため，モデルの評価と選抜には，到着回数のようにゲームの目的を直接表す指標を用いる必要がある．

6.3 後続研究の課題としての本環境

本環境では，カード，物件購入，駅イベントを設定によって段階的に有効化できる．移動課題では候補スコアリングにより貪欲方策と同等の方策が得られたため，今後は最適方策との差を埋める先読みを伴う学習に加え，カード使用判断（例えば急行カードによる移動量の増加と候補数の増加），物件購入と資産管理，COM プレイヤの強さの変更などを順に加えることで，難易度の異なる課題を構成できる．また，貪欲方策は移動課題における強力な基準となる一方，カードや資産を含む課題では明確な最適行動が存在しないため，強化学習による長期的な戦略獲得の題材となる．

6.4 本研究の制約

本研究には以下の制約がある．

(1) 実際の桃太郎電鉄の全ルールを再現したものではなく，研究用に単純化した環境である．例えば，エージェントは移動量ちょうどで目的地に止まる必要がある．

(2) 学習の実験はサイコロだけで移動する課題に限られ，学習ステップ数も約 100,000 と短い．

(3) 動的計画法のモデルは，新しい目的地の選び方を一部簡略化している．

(4) 移動候補は駅 ID 順の先頭 20 件に限られる．サイコロ 1 個の移動課題で 21 件以上の候補が生じるのは（駅，出目）の組の 0.7%であるが，移動量の大きいカードでは無視できない．

(5) COM プレイヤは単純なルールベースであり，複数の学習エージェントによる対戦は扱っていない．

7. おわりに

本稿では，すごろく型ボードゲームを対象とした強化学習研究の基盤として，ゲーム進行をフェーズとして定式化し，フェーズごとに有効な行動を action mask として提示する環境を構築した．「いけるかな」機能により，サイコロの出目によって変化する移動候補を固定長の離散行動へ対応付けた．サイコロだけで移動する課題でも貪欲方策は最適ではなく，動的計画法による最適方策は約 5%多く到着する．標準的な多層パーセプトロンを用いた MaskablePPO はランダム方策を下回った一方，各候補を同一のネットワークで評価する方策構造を用いると，すべての学習 seed で貪欲方策と同等の水準に達した．今後は，先読みを伴う学習によって最適方策との差を埋めるとともに，カード使用，物件購入，資産管理を段階的に加え，長期的な戦略の学習へ取り組む．

© 2026 Information Processing Society of Japan

参考文献

[1] Bailis, P., Fachantidis, A. and Vlahavas, I.: Learning to Play Monopoly: A Reinforcement Learning Approach, *Proc. 50th Anniversary Convention of the AISB* (2014).

[2] Haliem, M., Bonjour, T., Alsalem, A., Thomas, S., Li, H., Aggarwal, V., Bhargava, B. and Kejriwal, M.: Learning Monopoly Gameplay: A Hybrid Model-Free Deep Reinforcement Learning and Imitation Learning Approach, arXiv:2103.00683 (2021).

[3] Szita, I., Chaslot, G. and Spronck, P.: Monte-Carlo Tree Search in Settlers of Catan, *Advances in Computer Games (ACG 2009)*, LNCS, Vol.6048, pp.21–32, Springer (2010).

[4] Gendre, Q. and Kaneko, T.: Playing Catan with Cross-Dimensional Neural Network, *Neural Information Processing (ICONIP 2020)*, LNCS, Vol.12533, pp.580–592, Springer (2020).

[5] Huang, S. and Ontañón, S.: A Closer Look at Invalid Action Masking in Policy Gradient Algorithms, *Proc. International FLAIRS Conference*, Vol.35 (2022).

[6] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O.: Proximal Policy Optimization Algorithms, arXiv:1707.06347 (2017).

[7] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M. and Dormann, N.: Stable-Baselines3: Reliable Reinforcement Learning Implementations, *Journal of Machine Learning Research*, Vol.22, No.268, pp.1–8 (2021).

[8] Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., Gelly, S. and Bachem, O.: What Matters for On-Policy Deep Actor-Critic Methods? A Large-Scale Study, *Proc. ICLR* (2021).

[9] Ng, A. Y., Harada, D. and Russell, S.: Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping, *Proc. ICML*, pp.278–287 (1999).

[10] Towers, M. et al.: Gymnasium: A Standard Interface for Reinforcement Learning Environments, arXiv:2407.17032 (2024).

© 2026 Information Processing Society of Japan