Generative Flow Networksを用いた動的環境における移動ロボットナビゲーション

○松本 耕平（九州大学）　倉爪 亮（九州大学）

歩行者で混雑した環境での移動ロボットナビゲーション手法として，強化学習を用いた手法が盛んに研究されている．強化学習は最適な行動を学習できるが，行動の多様性を保つことは難しい．一方で，生成モデルの一種である Generative Flow Networks は報酬に比例して行動をサンプルでき，行動の多様性を保てる可能性がある．本研究では GFlowNets を用いた新たな学習ベースの移動ロボットナビゲーションを提案する．特に，異なる 2 種類のモデルアーキテクチャを提案し，歩行者による動的環境のシミュレーションによって評価する．

1. はじめに

生活環境で動作するサービスロボットにとって，歩行者が行き交う動的な環境での自律移動は重要な課題である. 近年，この課題に対して深層強化学習を利用した手法が盛んに研究されている．強化学習は最適な行動方策を学習可能な一方で，行動生成の多様性を保つことは難しい．一方で，近年様々な分野で生成モデルが利用されており，多様な目標データを生成するという観点において，多くの成功を収めている．特に，Generative Flow Networks (GFlowNets) [1] は強化学習と近い問題設定で研究が進められており，既存の強化学習への応用が期待される．しかしながら，これまでに学習ベースの移動ロボットナビゲーションには応用されておらず，有効に利用可能であるか未知数である．

そこで，本研究では，GFlowNets を学習ベースの移動ロボットナビゲーション手法に適用可能であるかを検証する．特に，2 種類のモデルアーキテクチャを提案し，それらを比較する．

2. Generative Flow Networks

GFlowNets は目標とするオブジェクト生成を状態のシーケンスとして，サンプルするモデルである．このシーケンスの集まりは有向非循環グラフ (DAG) $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ として表される．ここで，$s \in \mathcal{S}$ は状態を表し，$(s \to s') \in \mathcal{A}$ は状態の遷移を表す.

GFlowNets の学習では，中間状態では，通過するフロー $F(s)$ が，終端状態（目標のオブジェクトが構築された状態）がフローの報酬 $R(s)$ を吸収するシンクになるように保存され，入ってくるフローと出ていくフローが同量になるという，フローのバランス条件を満たすようにモデルを訓練する．この条件は，状態の軌跡に対して以下のように記述できる．

$$F(s_n) \prod_{i=n}^{m-1} P_F(s_{i+1} \mid s_i) = F(s_m) \prod_{i=n}^{m-1} P_B(s_i \mid s_{i+1}). \tag{1}$$

ここで，P_F と P_B はフォワードポリシーとバックワードポリシーであり，ある状態から前方と後方に向かうフローの割合を表す．この条件を満たすことで，終端状態は報酬に比例する確率でサンプリングされることが保証されている．

3. 問題設定

本研究では，GFlowNets を歩行者における動的環境での移動ロボットナビゲーションに適用するために，状態と状態遷移を以下のように設定する．

- 状態：歩行者とロボットの位置情報を状態として利用する．ロボットの状態はベクトル (p_x^r, p_y^r) であり，ゴールからロボットの位置の差分を表す．歩行者の状態は (p_x^i, p_y^i) であり，i 番目の歩行者において，ロボットからの相対位置を表す.
- 状態遷移：ホロノミックな全方位移動ロボットを想定し，2 次元空間におけるロボットの x 軸方向の入力速度 v_x と y 軸方向の入力速度 v_y からなる 2 次元ベクトル (v_x, v_y) を用いる．

ロボットはスタート位置から歩行者との衝突を回避しつつゴール位置 (p_x^g, p_y^g) に辿り着くことを目指す.

3.1 提案手法

本節では提案手法のモデルアーキテクチャ，報酬の設定，学習手法について述べる．

3.2 モデルアーキテクチャ

本稿では 2 種類のモデルアーキテクチャを提案する．この 2 つのモデルはそれぞれ 3 つのモジュール，状態フロー推定モデル，フォワードポリシーバックワードポリシーを有している．フォワードポリシーは最終的に学習したいモデルであり，状態フロー推定モデルと，バックワードポリシーは 2. 節にて説明した条件を満たすようにフォワードポリシーを学習するために利用される．フォワードポリシーは状態遷移，つまり行動に対する確率の高さを推定する．行動はあらかじめ速度空間からサンプルしておいた値を利用することとし，離散的な値を取る．そのため確率の高さはカテゴリカル分布の logit として出力する．

提案する 2 種類のモデルのアーキテクチャをそれぞれ図 1, 2 に示す．これらのモデルの大きな違いはフォワードモデルとバックワードモデルの入力に行動を入力するかどうかである．モデル 1 ではロボットと歩行者の状態を入力として行動空間のすべての行動に対する logit を出力する．一方でモデル 2 では，ロボットと歩行者の状態に加えて行動を入力し，入力した行動に対する logit を出力する．状態を固定して，行動空間すべての行動を入力することで，モデル 1 と同様にすべての行動に対する logit を得ることができる．

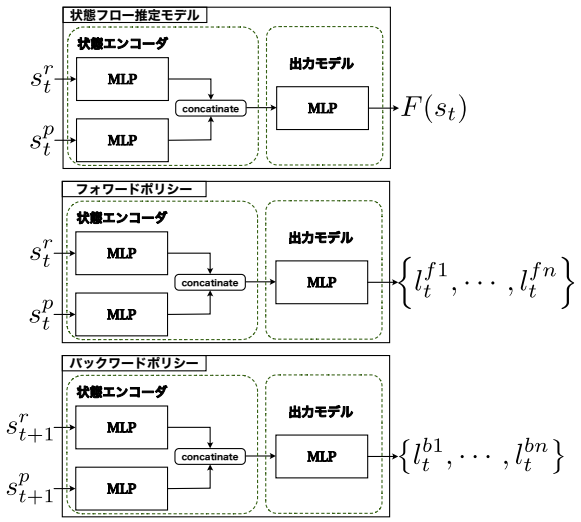

図 1　モデル 1 のアーキテクチャ

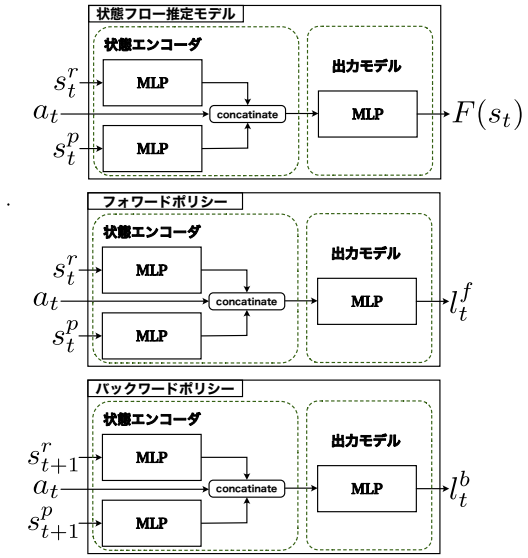

図 2　モデル 2 のアーキテクチャ

3.3　報酬の設定

本研究で扱う問題には通常，式 (2) のような報酬が利用される．

$$R(s_t) = \begin{cases} -0.25 & \text{if } d_t < 0 \\ -0.1 + d_t/2 & \text{else if } d_t < 0.2 \\ 1 & \text{else if } \boldsymbol{p}_t^r = \boldsymbol{p}_g \\ 0 & \text{otherwise} \end{cases} . \quad (2)$$

ここで，d_t はロボットと周囲の歩行者間の最小距離を表し，p_t^r は時刻 t におけるロボットの位置，p_g はロボットの目標位置を表す．この報酬では，衝突に対して負の値を与えることで，ペナルティを課し，衝突を回避する方策を学習することを促している．しかしながら，GFlowNets の性質上，負の報酬を利用することはできず，衝突に対するペナルティをどのように扱うかが課題である．そこで，本稿では衝突時には 0 より大きいが非常に小さな値を報酬として与える．学習後の GFlowNets は報酬に比例する確率でサンプリングを行うため，報酬が極めて小さい状態はサンプルされる可能性が低くなる．この報酬を式 (3) に示す．

$$R(s_t) = \begin{cases} 1.0 \times 10^{-4} & \text{if } d_t < 0 \\ 2.0 & \text{else if } \boldsymbol{p}_t^r = \boldsymbol{p}_g \\ 1.0 \times 10^{-1} & \text{otherwise} \end{cases} . \quad (3)$$

3.4　学習手法

モデルの学習方法として Subtrajectory balance (SubTB) [2] を採用した．SubTB ではエピソードの軌跡から部分的な軌跡を切り出し，式 (4) に従って損失を計算する．

$$\mathcal{L}_{\text{SubTB}}(\tau) = \left(\log \frac{F(s_m;\theta) \prod_{i=m}^{n-1} P_F(s_{i+1} \mid s_i;\theta)}{F(s_n;\theta) \prod_{i=m}^{n-1} P_B(s_i \mid s_{i+1};\theta)} \right)^2 . \quad (4)$$

ただし，x が終端状態である場合は，$F(x;\theta) = R(x)$ とする．

この手順を，エピソード全体の軌跡に渡って様々な軌跡長で行い，式（5）に従って統合することで最終的な損失を得る．

$$\mathcal{L} = \frac{\sum_{0 \le i < j \le n} \lambda^{j-i} \mathcal{L}_{\text{SubTB}}(\tau_{i:j})}{\sum_{0 \le i < j \le n} \lambda^{j-i}} . \quad (5)$$

ここで，λ は異なる長さの部分軌跡に割り当てる重みをコントロールするパラメータである．この損失関数を用いて，アルゴリズム 1 に従ってフォワードポリシー，バックワードポリシー，状態フロー推定モデルを学習する．

Algorithm 1: GFlowNets を用いたモデルの学習アルゴリズム

フォワードポリシー P_f，バックワードポリシー P_b, 状態フロー推定モデル F を初期化
for $i = 1$ **to** E **do**
　探索を行い軌跡 $\left(\boldsymbol{s}^r, \boldsymbol{s}^h, \boldsymbol{a}, \boldsymbol{r}\right)$ をバッファ $\mathcal{B}$ に保存
　$\mathcal{B}$ から M 個の軌跡をサンプルする
　for $j = 1$ **to** M **do**
　　以下の軌跡を取得する
　　ロボットの状態 $\boldsymbol{s}^{r,j} = \left\{s_1^{r,j}, s_2^{r,j}, \ldots, s_T^{r,j}\right\}$,
　　歩行者の状態 $\boldsymbol{s}^{h,j} = \left\{s_1^{h,j}, s_2^{h,j}, \ldots, s_T^{h,j}\right\}$,
　　行動 $\boldsymbol{a}^j = \left\{a_1^j, a_2^j, \ldots, a_T^j\right\}$,
　　終端状態の報酬 r^j
　　式 (5) に従い損失 $\mathcal{L}^j$ を計算
　end
　$\frac{1}{T}\sum_{i=1}^{T} \mathcal{L}^i$ を最小化するように P_f,P_b,F を更新
end

4.　シミュレーション実験

GFlowNets と式 (3) で表される報酬を用いて，回避行動を獲得することができるか提案する 2 つのモデルに関して，シミュレーション上での比較を行った．評価項目としては，成功率，衝突率，到達時間を用いた．

4.1 シミュレーション環境

シミュレーション環境には，CrowdNav 環境の circle crossing シナリオを利用する [3, 4]．circle crossing シナリオでは，ロボットは初期位置 $(x, y) = (0, -4)$ からゴール地点 $(x, y) = (0, 4)$ を目指して進む．歩行者の行動は ORCA によって生成されており，歩行者の初期位置はエピソードごとに半径 4m の円上にランダムに設定される．評価は 500 パターンのテストケースを用いて行う．ロボットの行動のサンプル数は 81 とし，環境内の歩行者数は 5 人に設定した．

4.2 実験結果

各モデルの結果の軌跡の例を図 3 に示す．モデル 1 は全体的に直線的な動作をしており，衝突前に速度を落として歩行者が過ぎ去るのを待つ傾向があった．一方，モデル 2 は左右に大回りしてゴールに辿り着く傾向が見られた．両モデルとも軌跡 4 は歩行者との衝突が発生してしまった例を示している．どちらのモデルにおいても，成功時との傾向は同じであるが，歩行者を回避できずに衝突してしまっていることがわかる．次

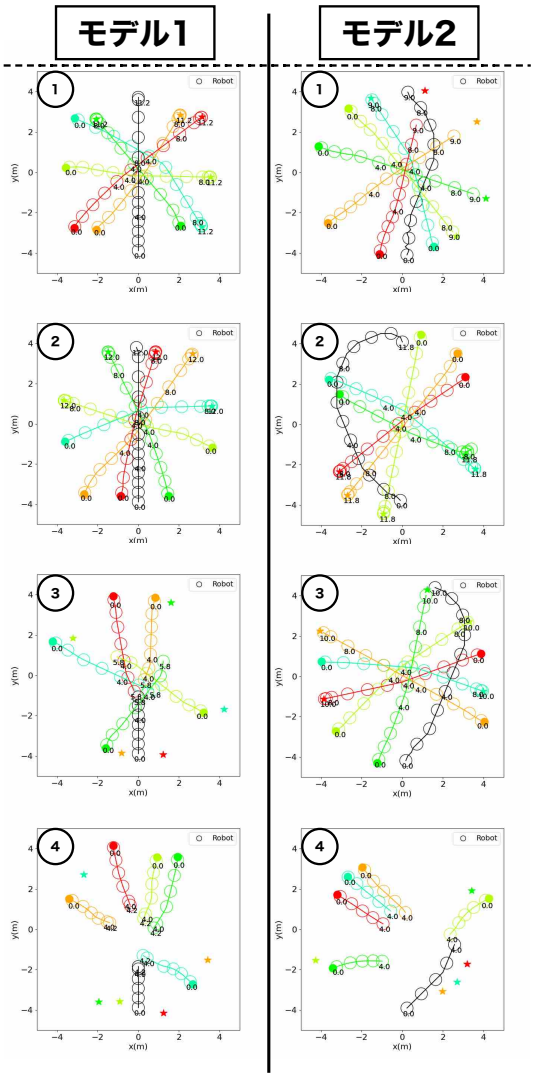

図 3　2 種類の提案モデル間の結果の軌跡の比較

に，モデル 1 とモデル 2 の結果の数値比較を表 1 に示す．モデル 1 では成功率が 33.6% しかないが，モデル 2 は 85.2% であり，モデル 2 の方が高い性能であった．ゴールまでの平均到達時間に関しては軌跡の比較からもわかるようにモデル 1 の方が優れた結果となっている．しかしながら，今回の報酬の設計ではより早くゴールに到達することに対してより大きな報酬が与えられる設定にはなっておらず，より成功率が高いモデル 2 の方が学習性能が高く優れていると言える．

表 1　モデル 1 とモデル 2 の結果の数値比較

Method	成功率 [%]	衝突率 [%]	到達時間 [s]
モデル 1	33.6	66.0	11.61
モデル 2	85.2	13.8	16.5

5.　結論

本研究では GFlowNets を歩行者が行き交う動的な環境での移動ロボットナビゲーションに適用した．提案する 2 種類のモデルをシミュレーション環境にて比較した結果，モデル 2 の方が優れていることを確認した．今後は，実機での実験や，グラフニューラルネットなどを利用したより複雑なモデルアーキテクチャの構築，生成可能な行動の多様性の観点における他の強化学習手法との比較に取り組む予定である．

参　考　文　献

[1] E. Bengio, M. Jain, M. Korablyov, D. Precup, Y. Bengio, Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation, Advances in Neural Information Processing Systems (NeurIPS), pp. 27381–27394, 2021.

[2] K. Madan, J. Rector-Brooks, M. Korablyov, E. Bengio, M. Jain, A.C. Nica, T. Bosc, Y. Bengio, and N. Malkin, “Learning GFlowNets From Partial Episodes For Improved Convergence And Stability,” in Proceedings of the International Conference on Machine Learning (ICML), pp. 23467–23483, 2023.

[3] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, “Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 6015–6022, 2019.

[4] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, “Relational graph learning for crowd navigation,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 10007–10013, 2020.