

マルチエージェント協調巡回問題における エネルギー消費抑制手法の提案

基幹理工学部 情報理工学科
菅原研究室 4年

松本 航平

Outline

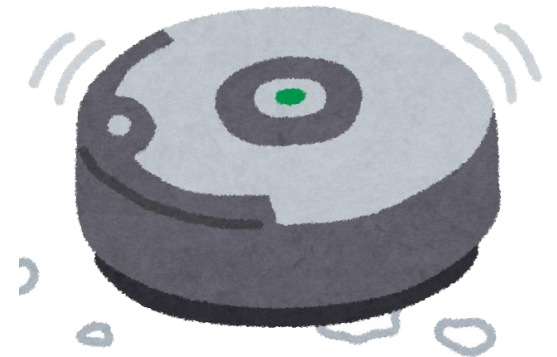
- 研究背景
- 先行研究・研究目的
- 提案手法
- 実験・考察
- 結論

Outline

- 研究背景
- 先行研究・研究目的
- 提案手法
- 実験・考察
- 結論

研究背景

- 近年では、ロボット技術が発達し、様々な分野で活躍するようになった
- 警備や清掃といった、人が日常的に繰り返す作業の代行が望まれる

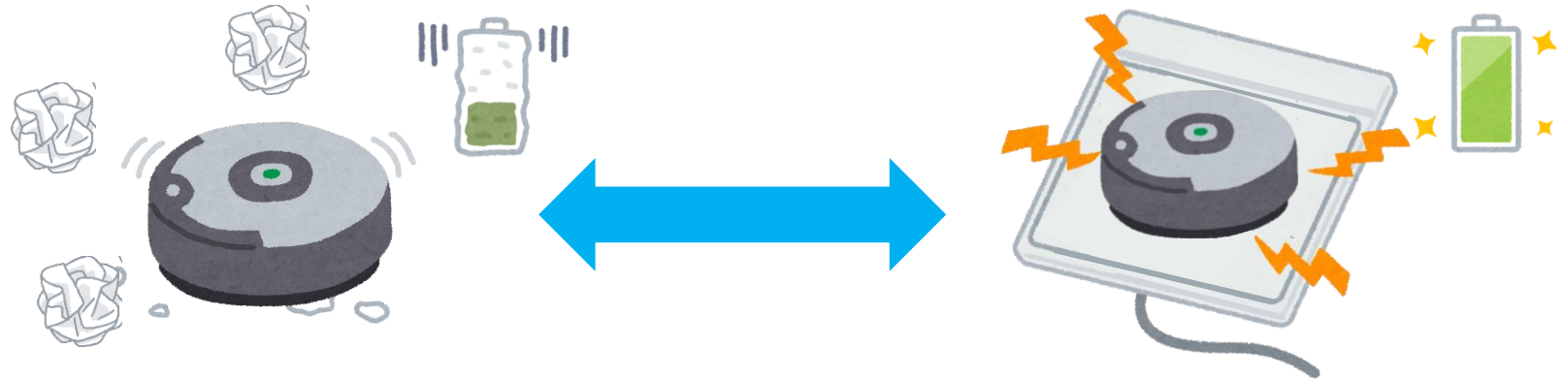


研究背景

- エージェント
 - 周囲の環境から自律的に行動を決定・学習する
- イベント
 - エージェントが観測(作業)するべきもの
 - ごみ(清掃問題), 危険度(セキュリティパトロール)
- マルチエージェント協調巡回問題
(multi-agent cooperative patrolling problem, MACPP)
 - 複数のエージェントが協調しながら共通のタスクを実行する問題の1つ

研究背景

- 自走式のロボットはバッテリーを持ち，巡回と充電を繰り返す



- 巡回効率の最大化(訪問間隔の短縮やごみ回収量の増加)を目指した行動や学習は，必要以上のエネルギーを消費する可能性がある

研究背景

- アプリケーションによっては、想定された品質要求が存在する場合がある
 - 部屋がどのくらいきれいであれば良いか(清掃問題)
 - 訪問間隔がどのくらい短ければ良いか(セキュリティパトロール)
- 過度の巡回作業は単位エネルギー当たりの作業効率を低下させる

Outline

- 研究背景
- 先行研究・研究目的
- 提案手法
- 実験・考察
- 結論

先行研究

- MACPPの解決手法

- ①環境を各エージェントの作業領域に分割し，領域内を巡回する手法[1]

- ②各エージェントが適切な戦略を用いて環境全体を巡回する手法[2]

(adaptive meta-target decision strategy, AMTDS)

- ②が①よりも少なくとも同程度に優れている[3]

[1] Katsuya Hattori and Toshiharu Sugawara. “Effective Area Partitioning in a Multi-Agent Patrolling Domain for Better Efficiency.” In Proc. of the 13th Int. Conf. on Agents and Artificial Intelligence - Volume 1: ICAART, pp. 281-288. INSTICC, SciTePress, 2021.

[2] Keisuke Yoneda, Chihiro Kato, and Toshiharu Sugawara. “Autonomous learning of target decision strategies without communications for continuous coordinated cleaning tasks.” In 2013 IEEE/WIC/ACM Int. Joint Confs. on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Vol. 2, pp. 216-223, 2013.

[3] Yann Chevaleyre, Francois Sempe, and Geber Ramalho. “A theoretical analysis of multi-agent patrolling strategies.” In AAMAS, Vol. 4, pp. 1524-1525, 2004.

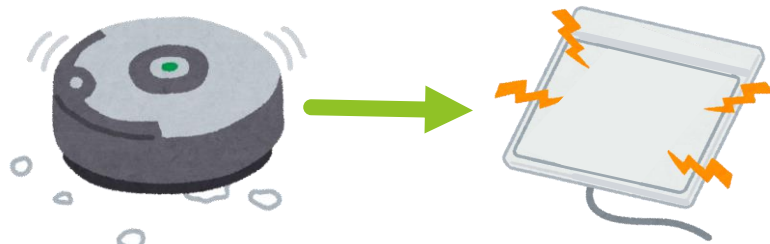
先行研究

AMTDS/ESC (AMTDS for energy saving and cleanliness) [4]

- エネルギー節約行動により, 品質要求を満たしながらエネルギー消費を抑制する手法

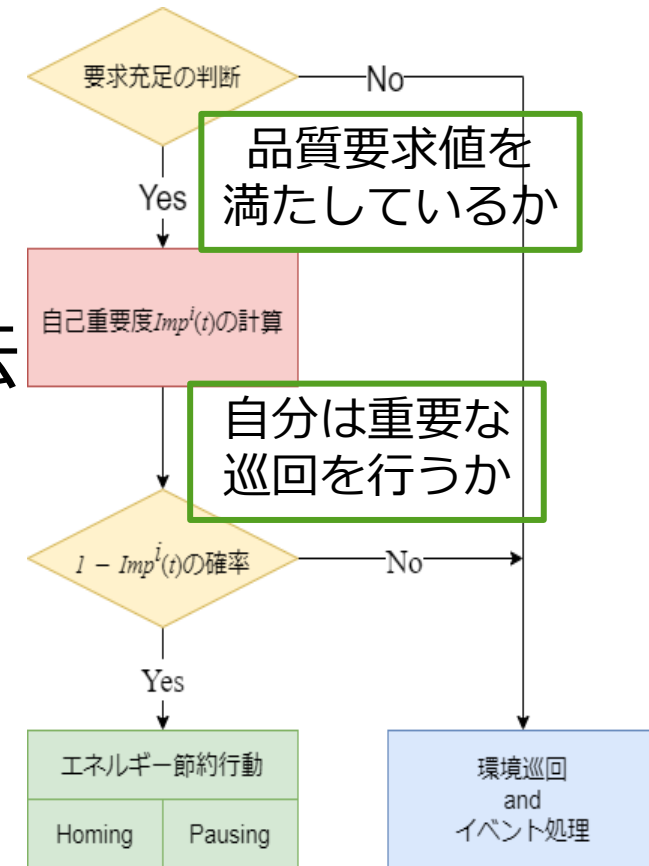
□ Homing(帰還動作)

バッテリーが残っていても
早めに充電基地に戻る



□ Pausing(待機動作)

充電基地で充電後
継続的に待機する



[4] Lingying Wu, Ayumi Sugiyama, and Toshiharu Sugawara. "Energy-efficient strategies for multi-agent continuous cooperative patrolling problems."

先行研究－課題点1

- HomingとPausingはどちらか片方しか行われていない
 - あらかじめどちらを行うか決めなければならない



Homingを行った時点で、環境の要求条件が満たされ、
現在では自分の巡回は重要ではないと判断している

➡ その後にPausingを行うことにより、より多くの
エネルギー消費の抑制が可能

先行研究－課題点2

- 品質要求を満たしているかの見積もりが
エージェントによって異なっていない
- 見積もりの精度が十分ではない



品質要求は満たせているが、エネルギー節約行動が
十分ではなく、よりエネルギー消費を抑制できる

研究目的

研究目的

品質要求を満たしながら，エネルギー消費をさらに抑制する

提案手法

- ① エネルギー節約行動の修正
- ② エネルギー節約行動を適応的に調整する

Outline

- 研究背景
- 先行研究・研究目的
- 提案手法
- 実験・考察
- 結論

提案手法

- **AMTDS/ESC** (AMTDS for energy saving and cleanliness)を拡張
 - エネルギー節約行動の修正
 - 未来のイベント量の予測の自律的補正学習の導入

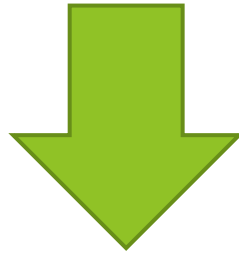


- **AMTDS/ER** (AMTDS for energy saving under the requirement)

エネルギー節約行動の修正

先行研究

- HomingとPausingのどちらかしか行っていなかった

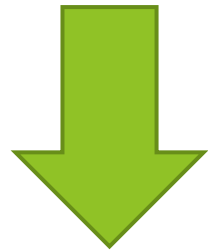


提案手法

- HomingとPausingどちらも行う
- Homing後の充電が完了時には必ずPausingを行う

未来のイベント量の予測

- イベント量の予測はPausing開始時に行うため、待機中の他のエージェントのイベント処理は考慮していない
- 実際の値は $E^i(L_{t+T}(V))$ よりも小さい可能性が高い
- そのずれは各エージェントの重点的に巡回する場所に依存する



v : 環境内のノード
 $L_t(v)$: 時刻 t における v でのイベント量
 $E^i(L_{t+T}(v))$: 時刻 t から T ステップ後の v のイベント予測量

各自が延長分を適切に決定する必要がある

予測の自律的補正学習

- 学習パラメータ K^i を導入
- 品質要求値 D_{req} に対して、以下の式を満たす最大の T を待機時間とする

$$\sum_{v \in V} E^i(L_{t+T}(v)) \div K^i \leq D_{req} (1)$$

※AMTDS/ESCでは待機時間 S_{pause} は固定であったが、
提案手法では T は可変

予測の自律的補正学習

- Pausing終了後, 以下の式に従って K^i を更新

$$\begin{cases} K^i \leftarrow (1 - \alpha)K^i + \alpha \frac{D_{req}}{E^i(D_t)} \times K^i & (\text{if } E^i(D_t) \leq D_{req}) \\ K^i \leftarrow K^i - \left(\frac{E^i(D_t)}{D_{req}} - 1 \right) & (\text{if } E^i(D_t) > D_{req}) \end{cases} \quad (2)$$

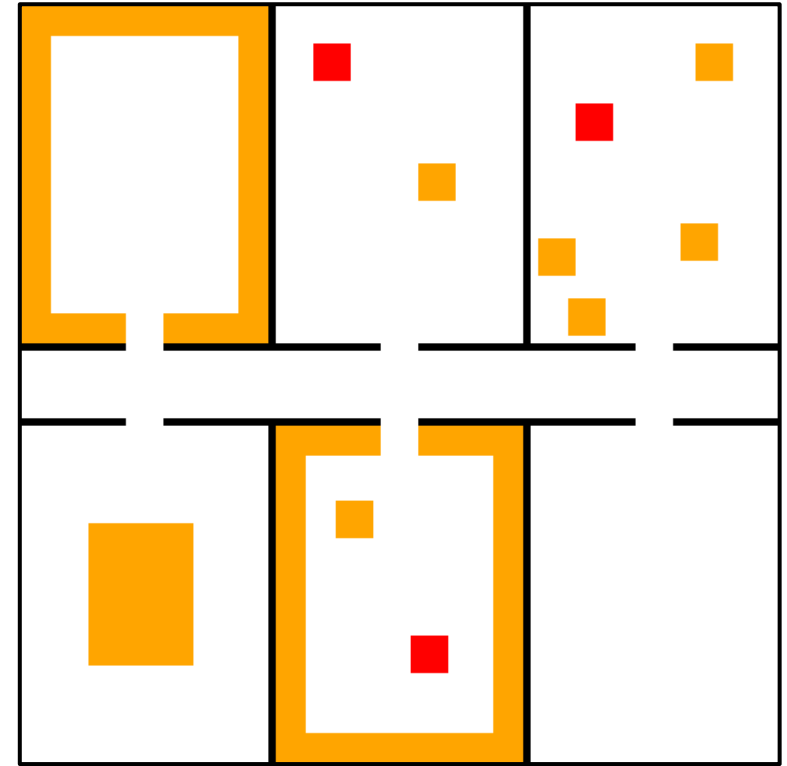
$E^i(D_t)$: エージェント i による予測イベント量

Outline

- 研究背景
- 先行研究・研究目的
- 提案手法
- 実験・考察
- 結論

実験環境

- 101×101 の2次元グリッド(x, y)
($-50 \leq x, y \leq 50$)
- 色の濃いほうがイベント発生確率が高い
(赤 > オレンジ > 白)
- 全エージェントの充電基地 $v_{base} = (0, 0)$
- エージェントは20体
- 連続稼働時間は900ステップで, フル充電に2700ステップ必要
- 1試行で3,000,000ステップ行い, 50試行の平均値をとる



評価指標

A : エージェントの集合

$L_t(v)$: ノード v の時刻 t における
イベント量

$E_t(i)$: エージェント i の時刻 t における
エネルギー消費量

- イベント残存時間の総和 (ごみの残存量)

$$D_{t_s, t_e}(s) = \sum_{v \in V} \sum_{t=t_s+1}^{t_e} L_t(v) \quad (4)$$

品質要求値以下ならば
大小は関係なし

- 総エネルギー消費量 (エージェントのエネルギー消費量)

$$C_{t_s, t_e}(s) = \sum_{i \in A} \sum_{t=t_s+1}^{t_e} E_t(i) \quad (5)$$

小さいほうが良い

評価指標

A : エージェントの集合

$L_t(v)$: ノード v の時刻 t における
イベント量

$E_t(i)$: エージェント i の時刻 t における
エネルギー消費量

- イベント残存時間の総和 (ごみの残存量)

$$D_{t_s, t_e}(s) = \sum_{t=t_s}^{t_e} D(s) L_t(v) \quad (4)$$

品質要求値以下ならば
大小は関係なし

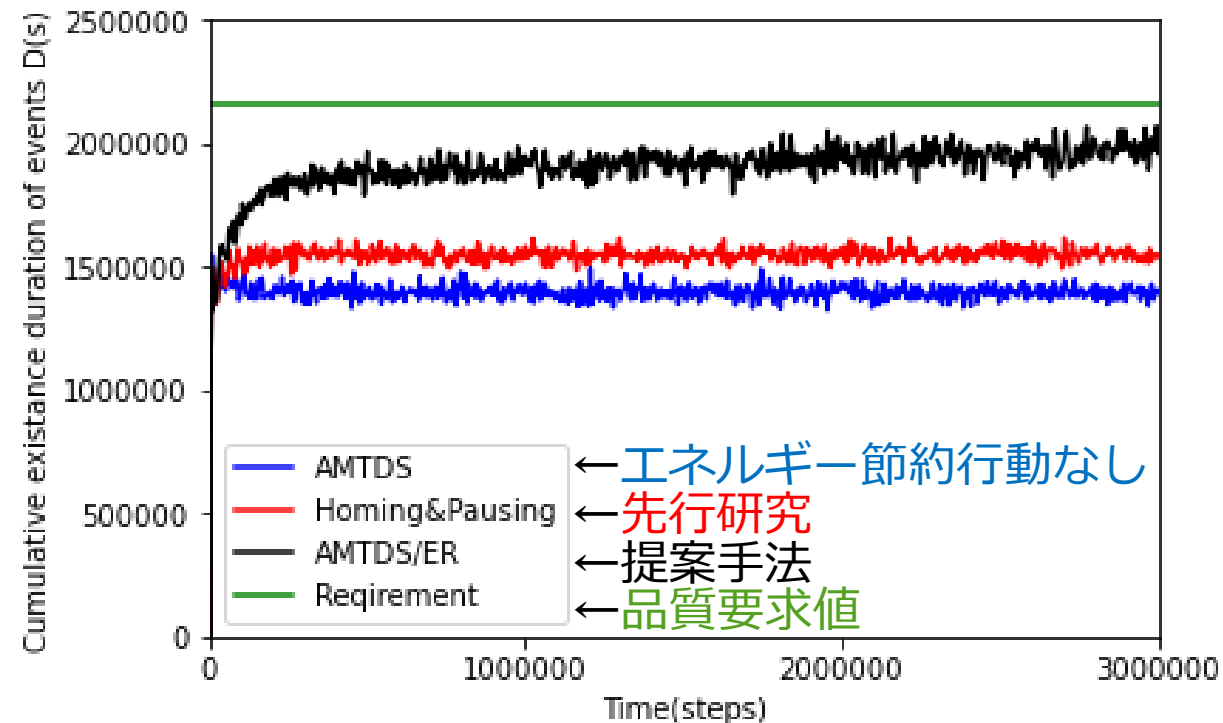
- 総エネルギー消費量 (エージェントのエネルギー消費量)

$$C_{t_s, t_e}(s) = \sum_{t=t_s}^{t_e} C(s) E_t(i) \quad (5)$$

小さいほうが良い

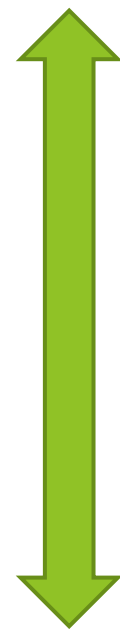
実験結果

品質要求値: $(600/\text{step} \times 3,600) = 2,160,000$

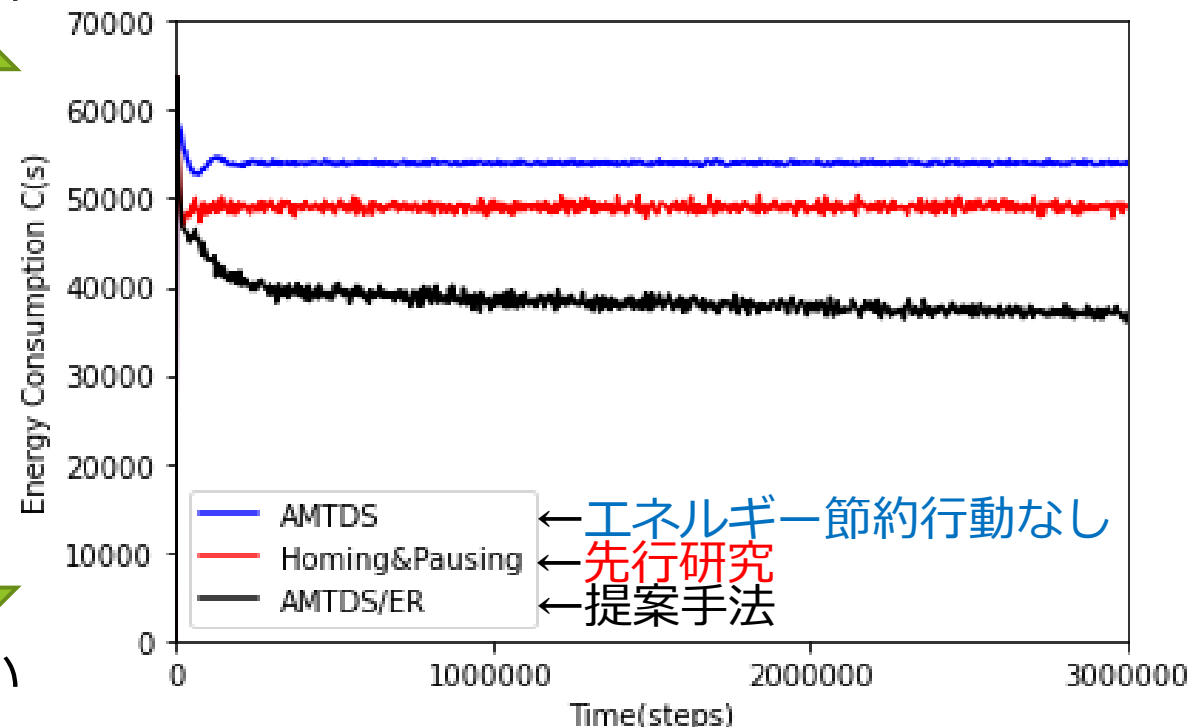


イベント残存時間の総和 D(s)

悪い



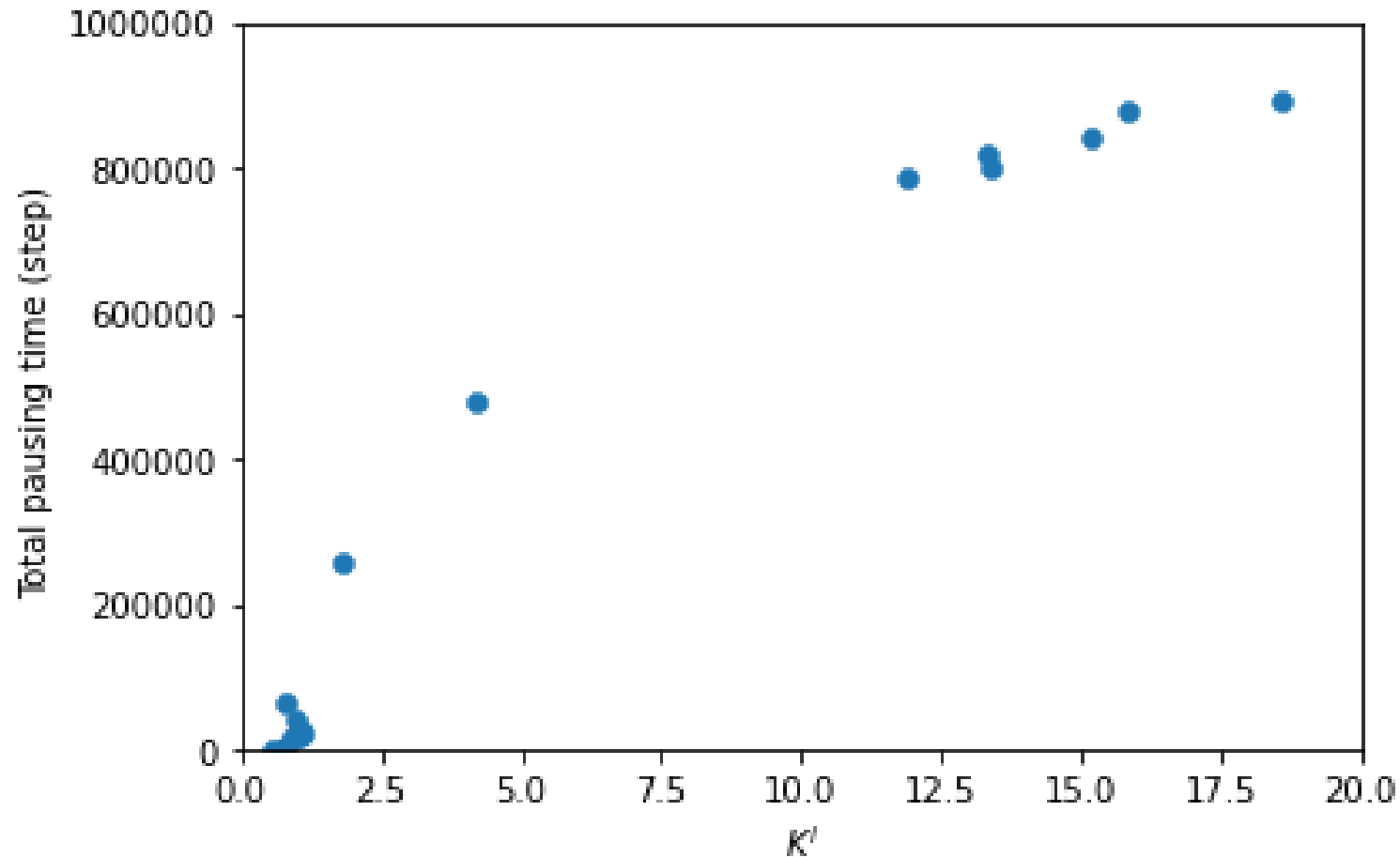
良い



総エネルギー消費量 C(s)

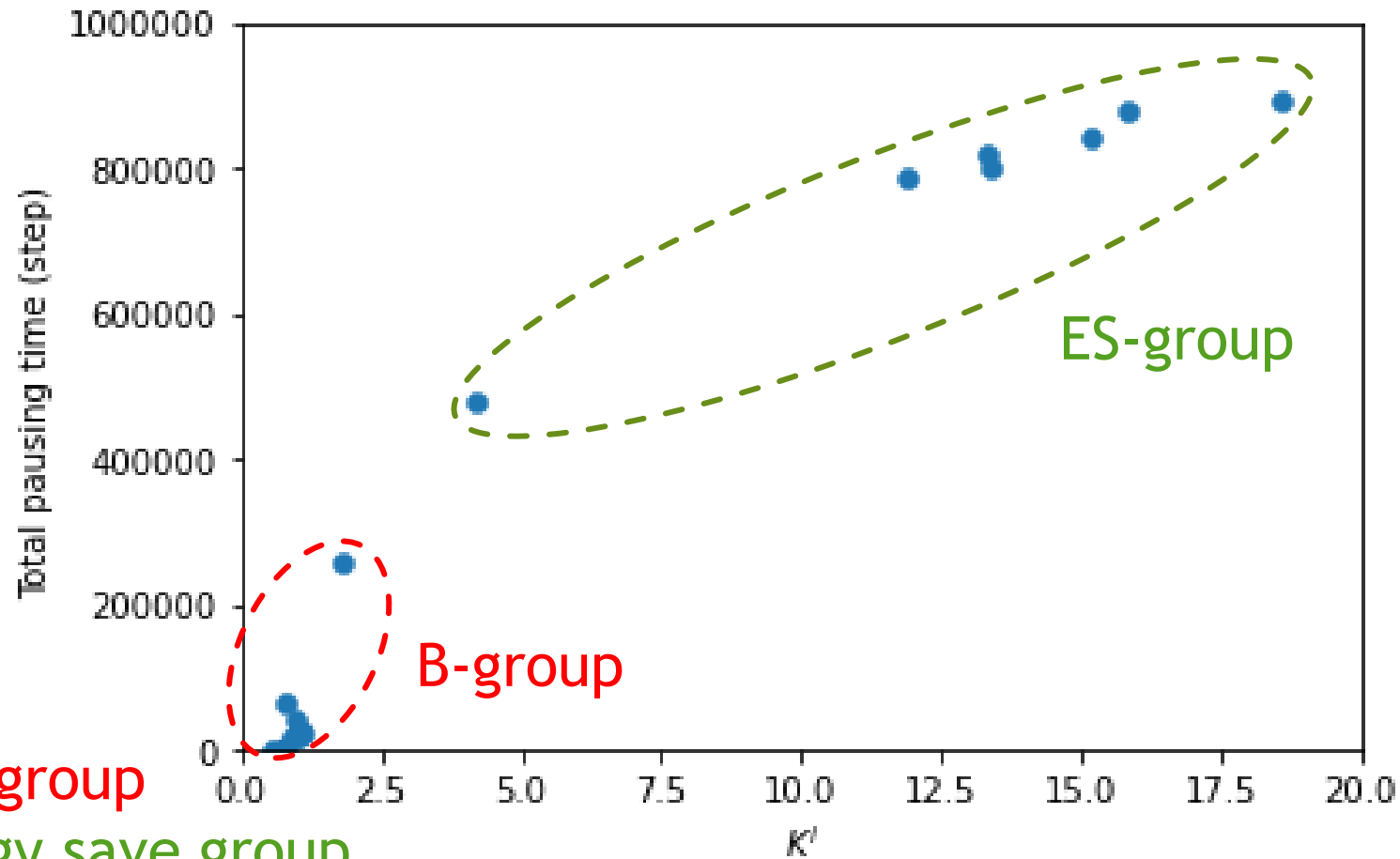
提案手法が最も品質要求値に近く、エネルギー消費量が小さい

パラメータ K^i と待機時間



エージェントによって K^i の値は大きく異なる

パラメータ K^i と待機時間



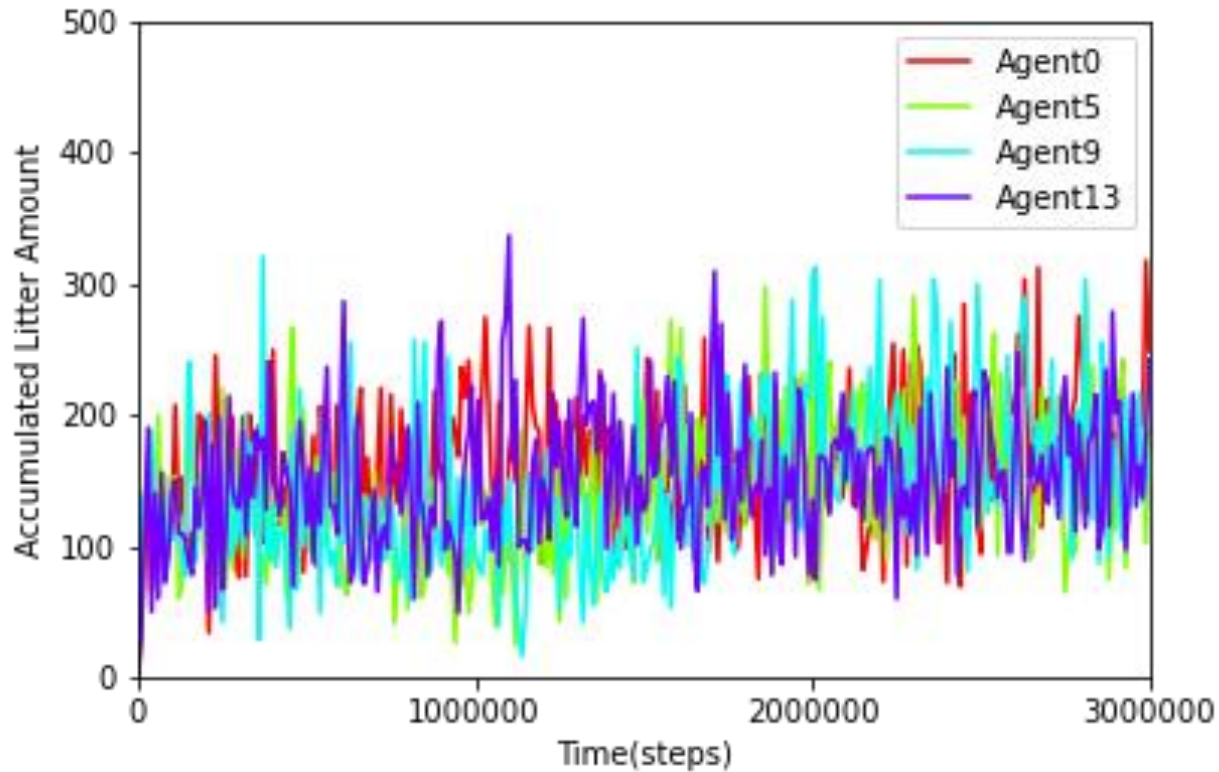
B-group : Busy group

ES-group : Energy save group

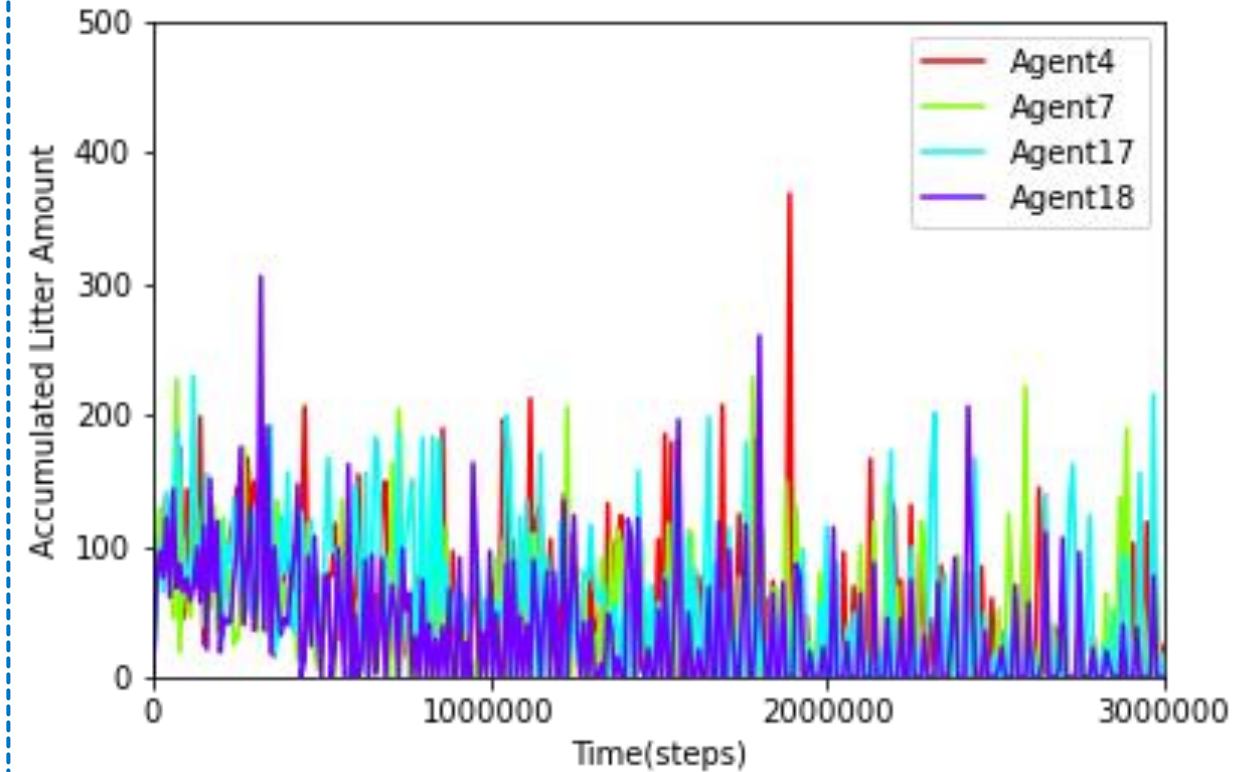
行動が異なる2つのグループに自律的に分かれた

各グループのイベント処理数

B-group 4体



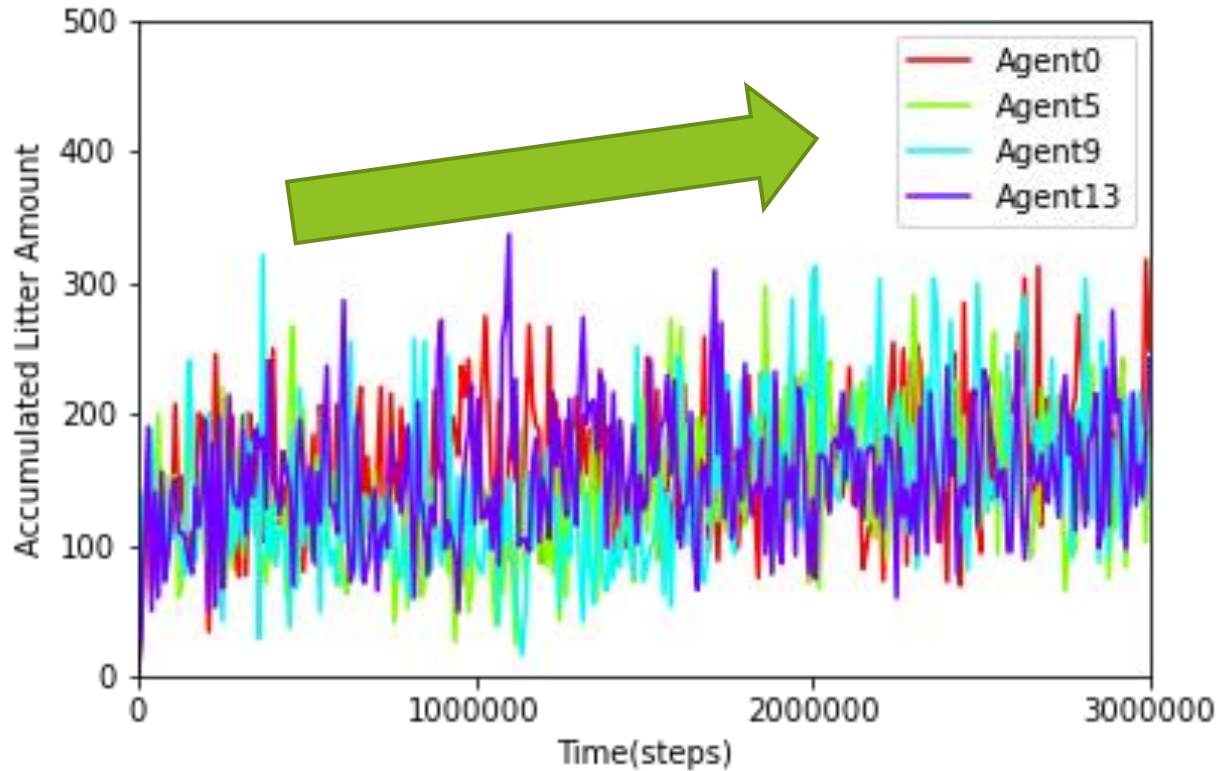
ES-group 4体



グループによってイベント処理量が異なる

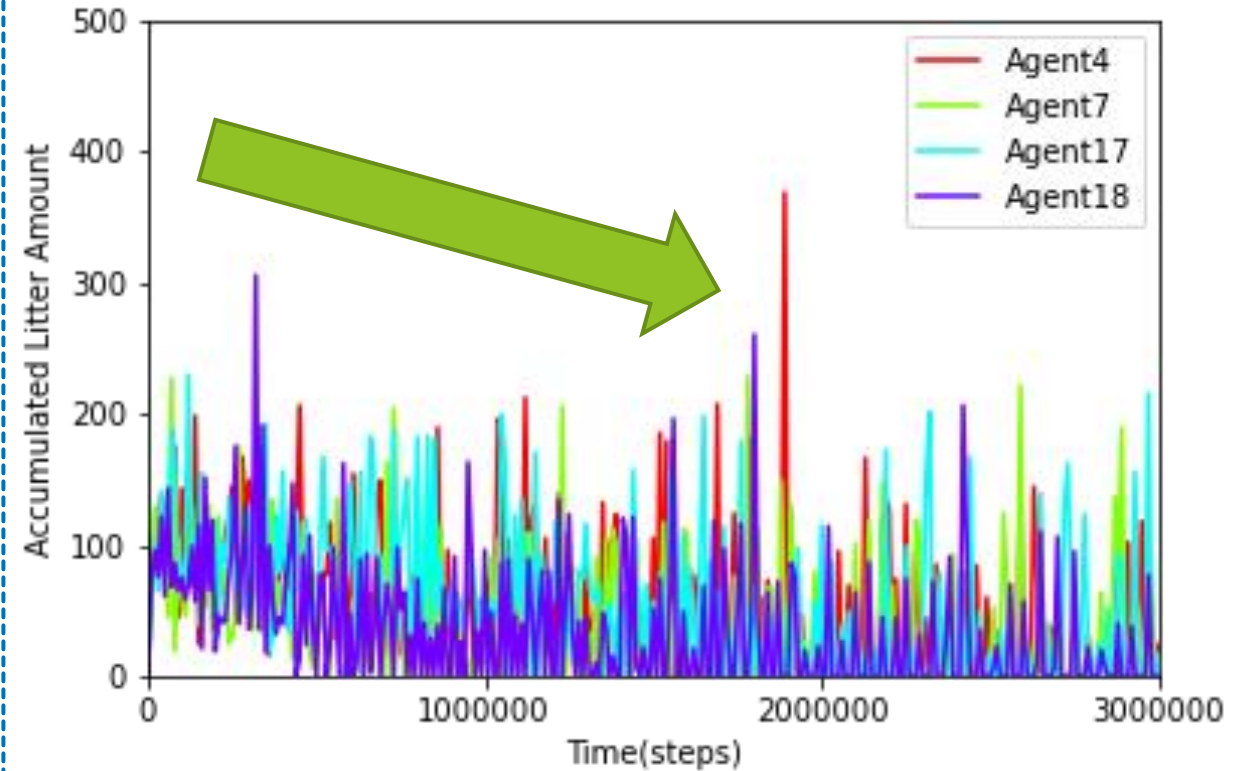
各グループのイベント処理量

B-group 4体



イベント処理数が増加

ES-group 4体



待機時間が長くなり,
イベント処理数が減少

グループによってイベント処理量が異なる

Outline

- 研究背景
- 先行研究・研究目的
- 提案手法
- 実験・考察
- 結論

結論

- 提案手法が従来手法よりも要求条件を満たしながら、
エネルギー消費を抑制することができた
- 予測の自律的補正学習の導入により、2つのグループ
(B-groupとES-group)に分かれた
- ES-groupを停止することにより、必要最低限の
エージェント数まで減らすことが可能

業績リスト

- 国内会議・研究会（査読なし）

- 松本航平, 米田圭佑, 菅原俊治. “マルチエージェント協調巡回問題におけるエネルギー消費抑制手法の提案.” 2022 Summer Symposium on Multi Agent Systems for Harmonization (SMASH22 Summer Symposium). 2022年9月

- 国際会議（査読あり）

- Kohei Matsumoto, Keisuke Yoneda and Toshiharu Sugawara. “Autonomous Energy-Saving Behaviors with Fulfilling Requirements for Multi-Agent Cooperative Patrolling Problem.” Proceedings of the 15th International Conference on Agents and Artificial (ICAART 2023). February, 2023. (accepted as a full paper)

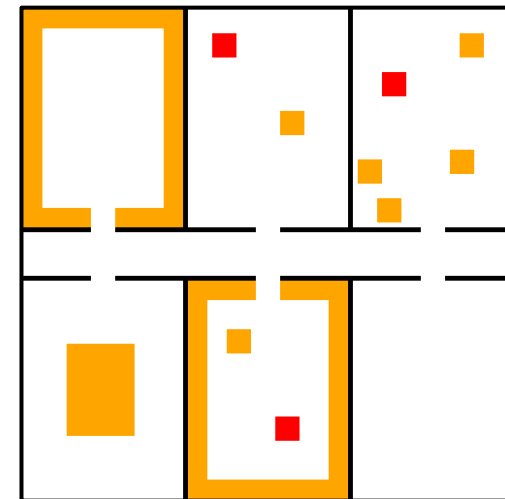
- 表彰

- 松本航平, 米田圭佑, 菅原俊治. “マルチエージェント協調巡回問題におけるエネルギー消費抑制手法の提案.” 2022 Summer Symposium on Multi Agent Systems for Harmonization. 2022. 優秀賞, 奨励賞

Appendix

今後の課題

- ES-groupを停止していき，品質要求を満たしたうえで，エージェント数を減らす
- AMTDS/ERをイベント発生確率を学習できるように拡張する



AMTDS

- バッテリーを持つ複数のエージェントが充電基地で充電しながら継続的な巡回を行う手法
- 環境内のごみの発生確率は**既知**
- 目標決定は4つの戦力から良い戦略を**Q学習**で選ぶ
 1. ランダム法
 2. 貪欲法
 3. 訪問間隔優先法
 4. 戦略目標決定法

$E^i(L_{t+T}(v))$ の求め方

$$E^i(L_{t+T}(v)) = p(v) \times ((t + T) - t_{vis}^v)$$

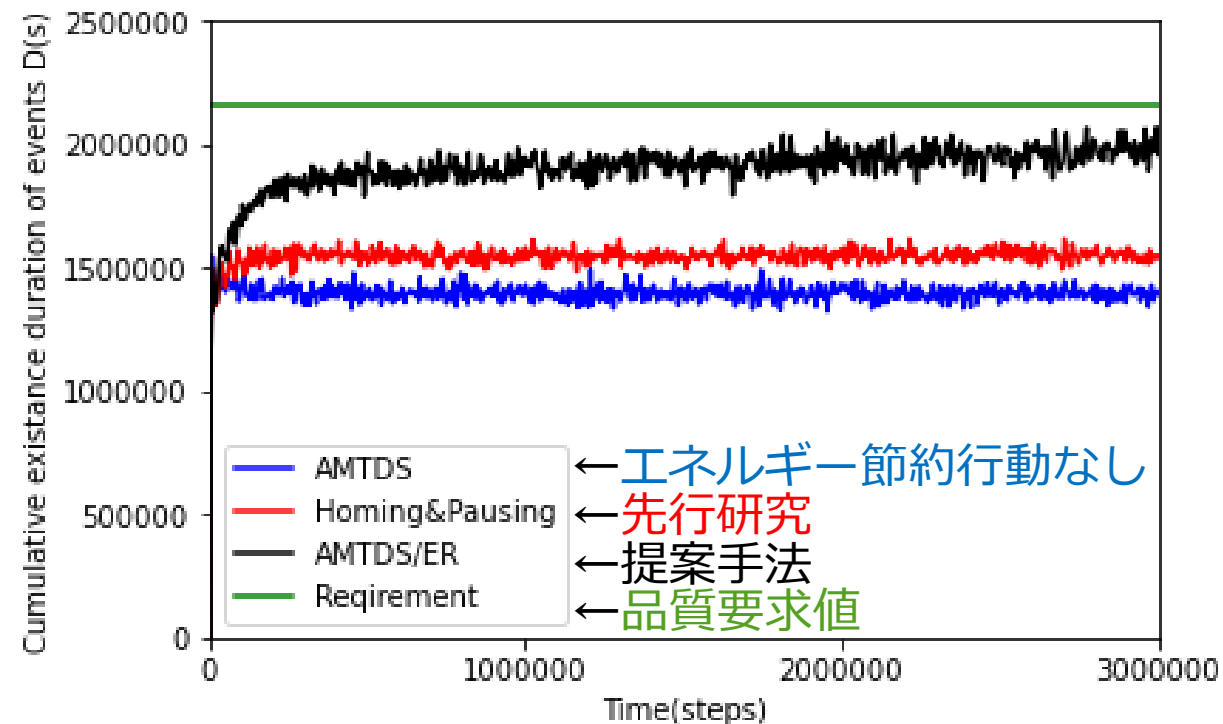
$p(v)$: v のイベント発生確率

t_{vis}^v : $\forall i \in A$ が v に訪問した最終時刻

実験結果(実験1)



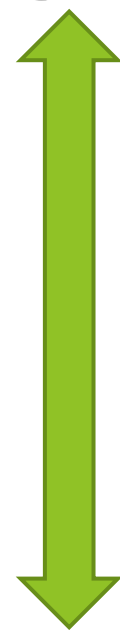
品質要求値: $(600/\text{step} \times 3,600) = 2,160,000$



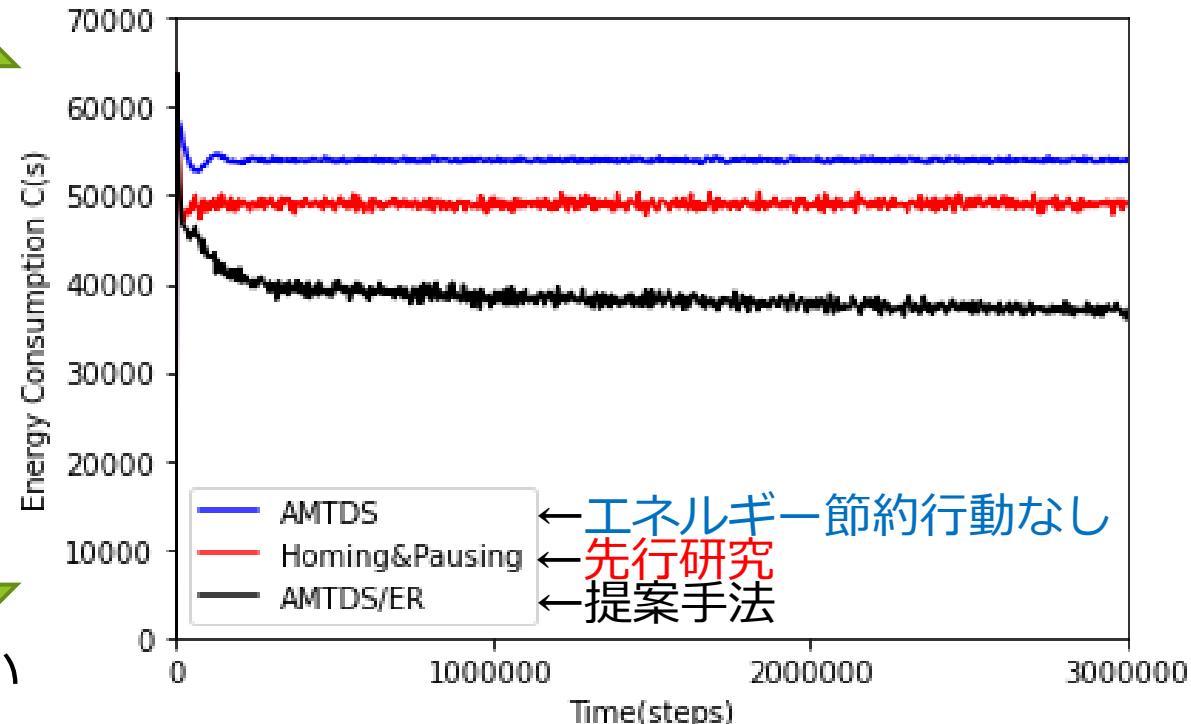
イベント残存時間の総和 $D(s)$

+10.6% $\begin{matrix} 1396642.2 \\ 1544735.5 \\ 1902030.4 \end{matrix}$ +36.2%

悪い



良い

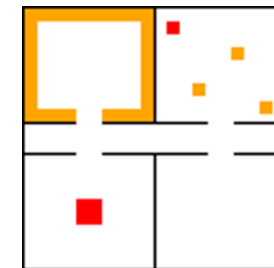


総エネルギー消費量 $C(s)$

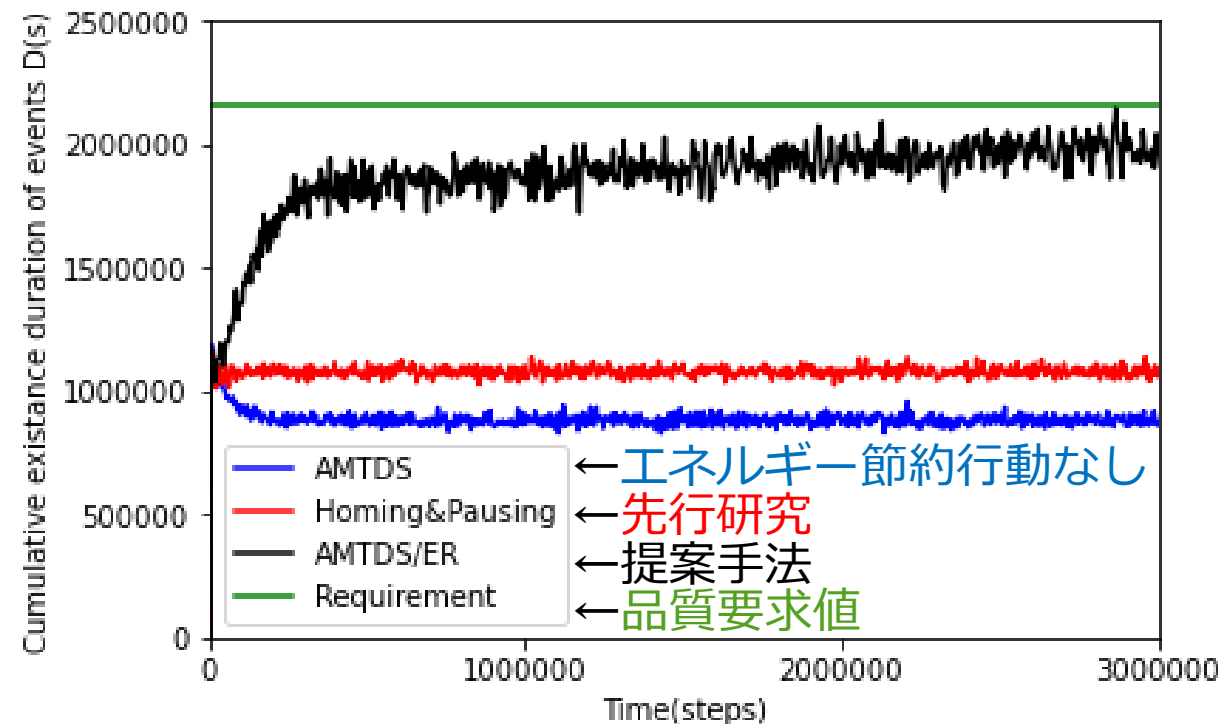
-9.2% $\begin{matrix} 53975.1 \\ 48996.3 \\ 38517.1 \end{matrix}$ -28.6%

K^i の導入により, より多くのエネルギー削減が実現

実験結果(実験2)

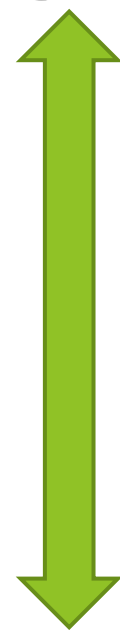


品質要求値: $(600/\text{step} \times 3,600) = 2,160,000$

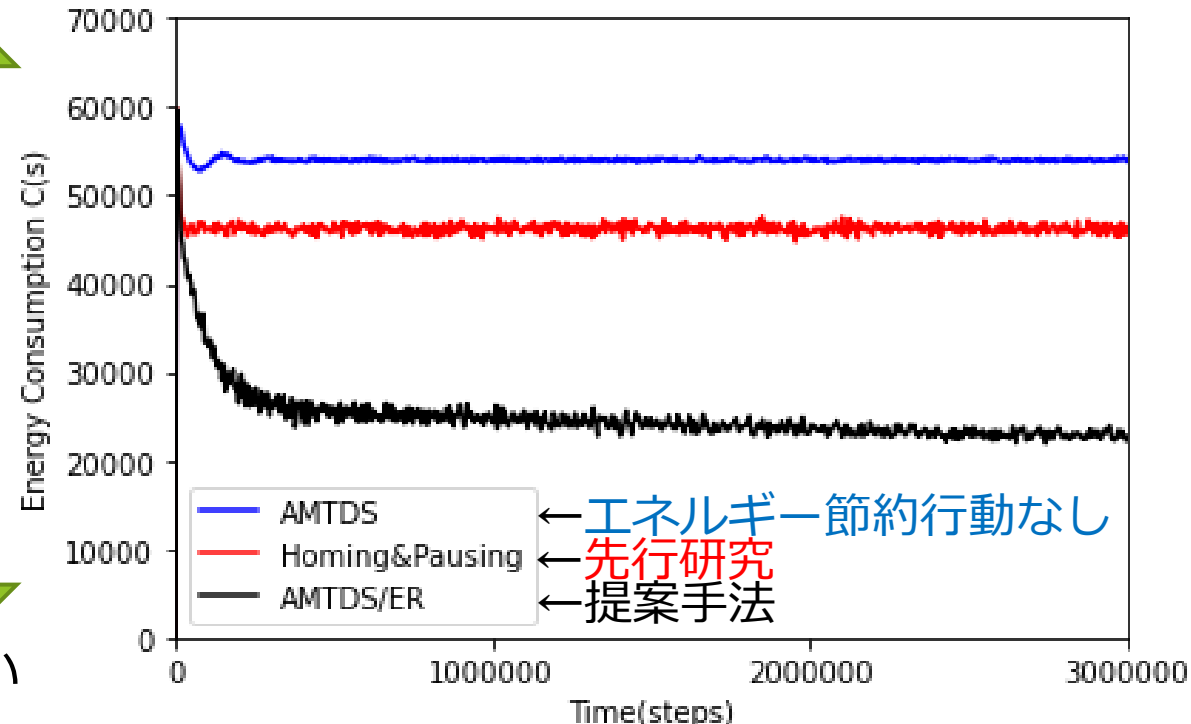


イベント残存時間の総和 $D(s)$

悪い



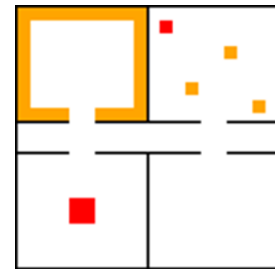
良い



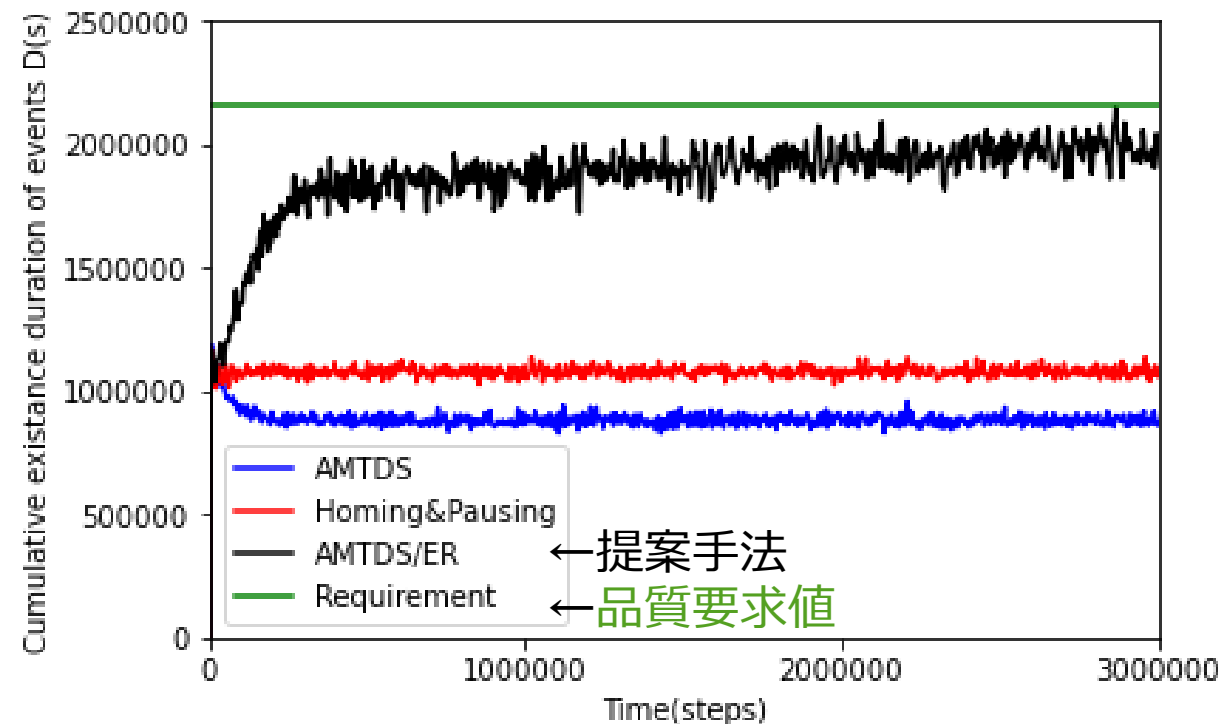
総エネルギー消費量 $C(s)$

環境に応じて**エネルギー節約行動**を調整できる

実験結果(実験2)



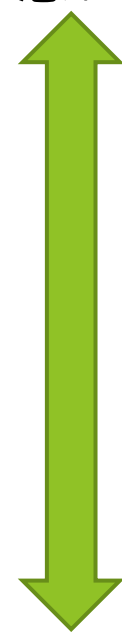
品質要求値: $(600/\text{step} \times 3,600) = 2,160,000$



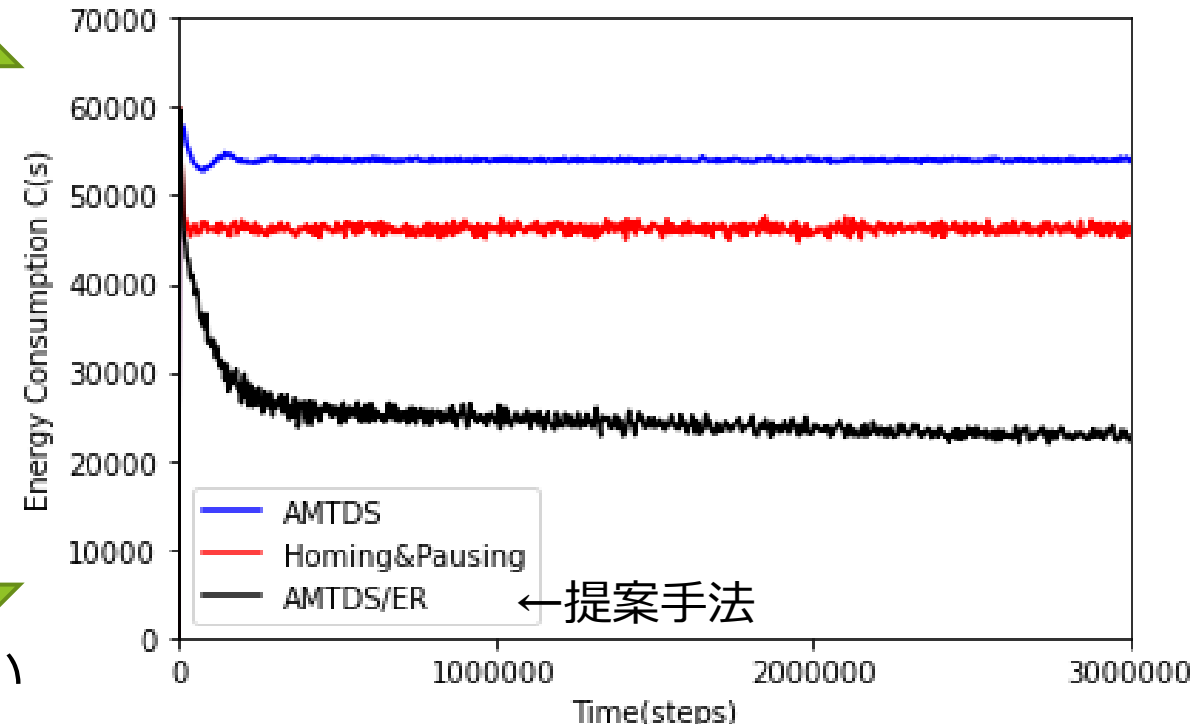
イベント残存時間の総和 D(s)

+21.9% $\begin{matrix} 883730.4 \\ 1076829.1 \\ 1869647.5 \end{matrix}$ +111.6%

悪い



良い

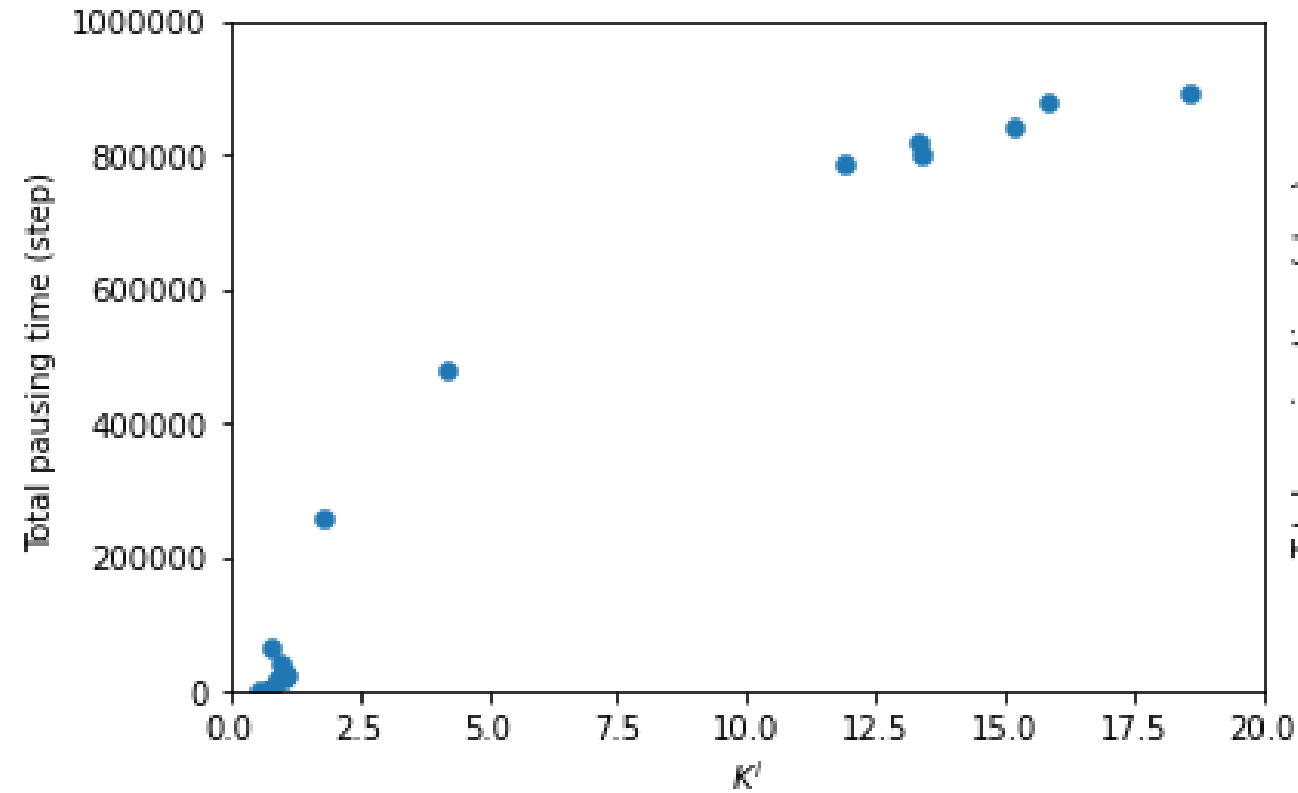


総エネルギー消費量 C(s)

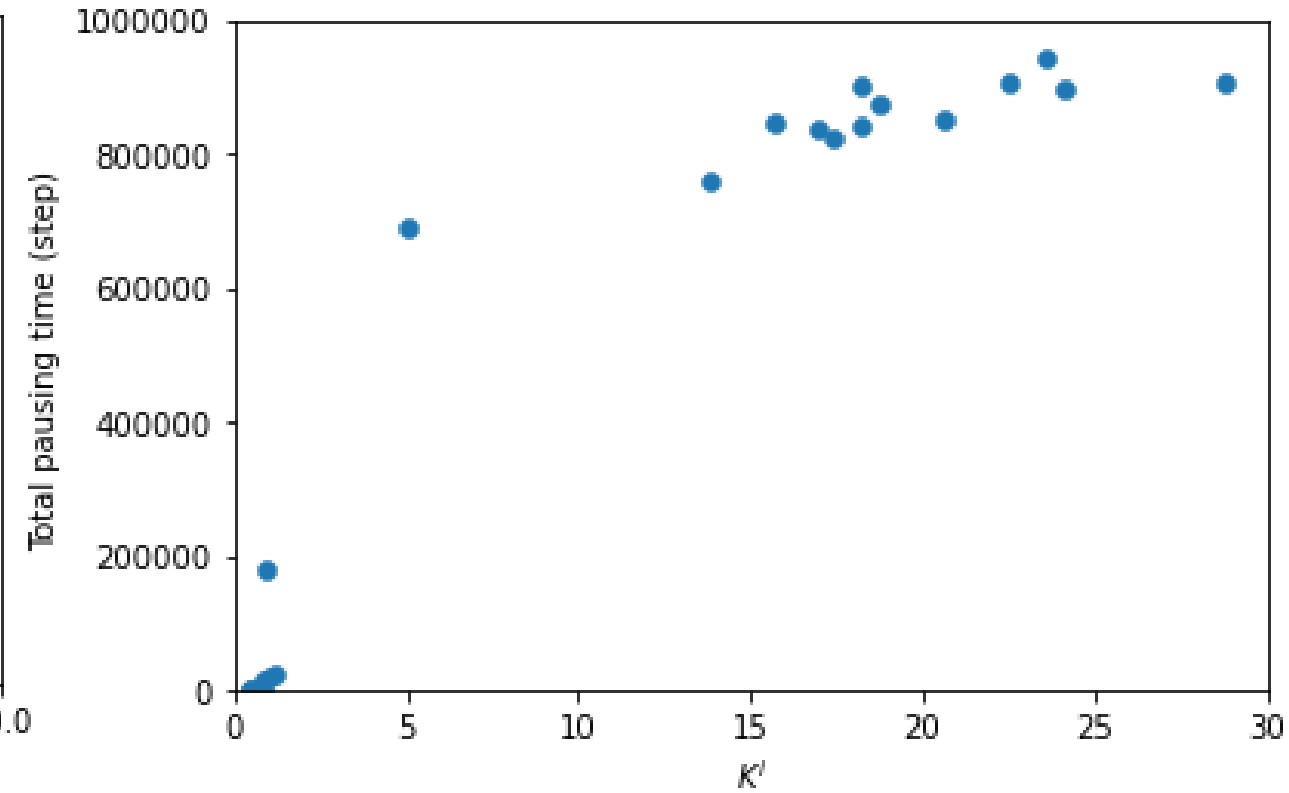
-14.4% $\begin{matrix} 53975.0 \\ 46209.2 \\ 24925.7 \end{matrix}$ -53.8%

環境に応じてエネルギー節約行動を調整できる

パラメータ K^i と待機時間



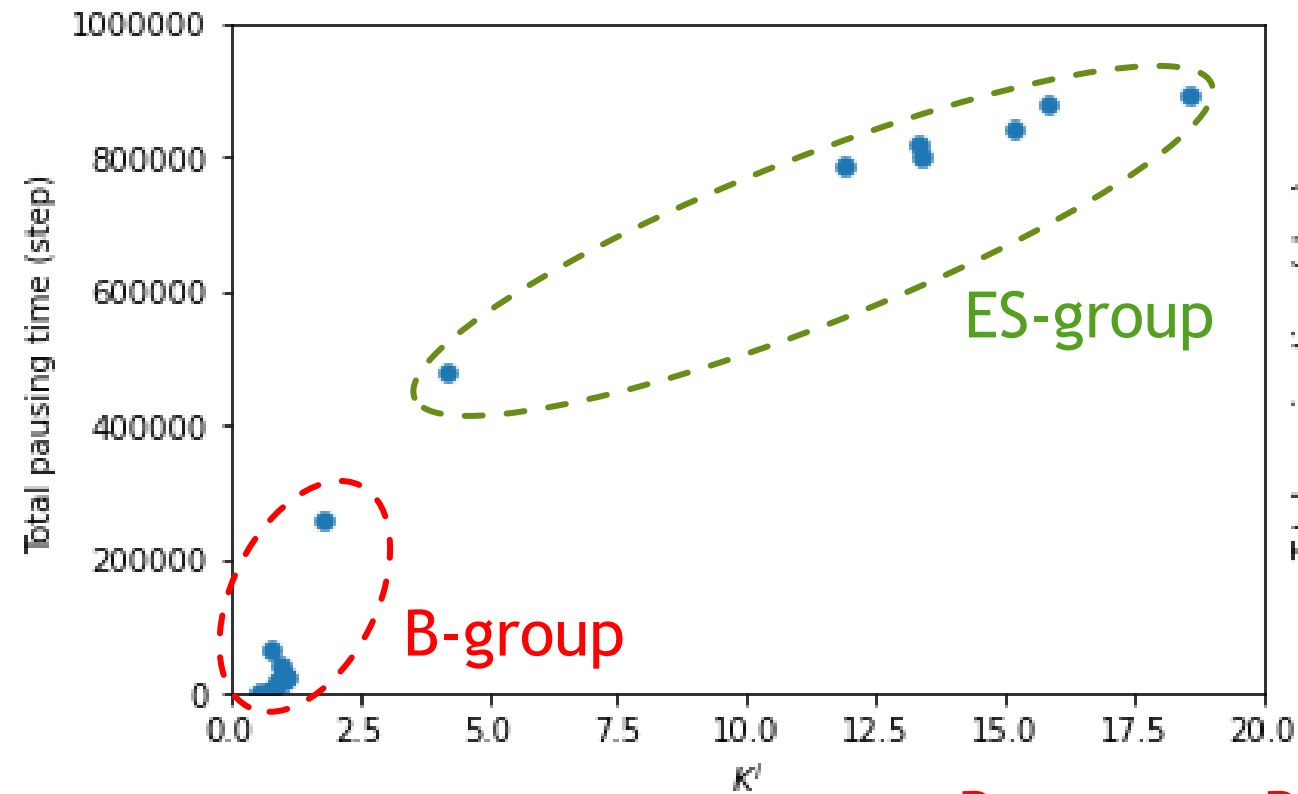
実験1



実験2

エージェントによって K^i の値は大きく異なる

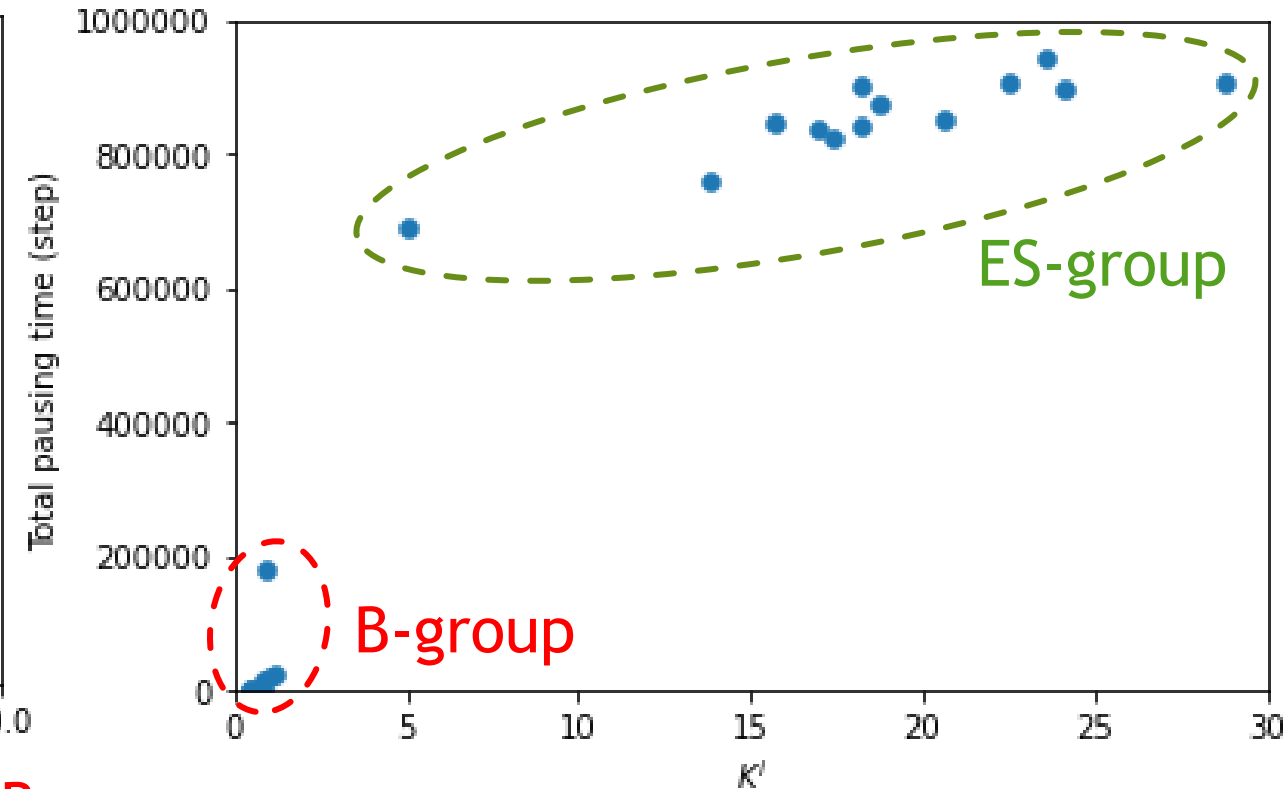
パラメータ K^i と待機時間



実験1

B-group : Busy group

ES-group : Energy save group



実験2

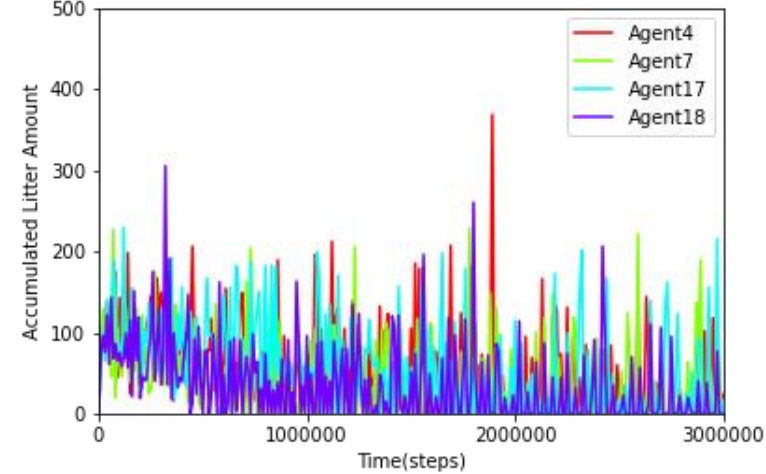
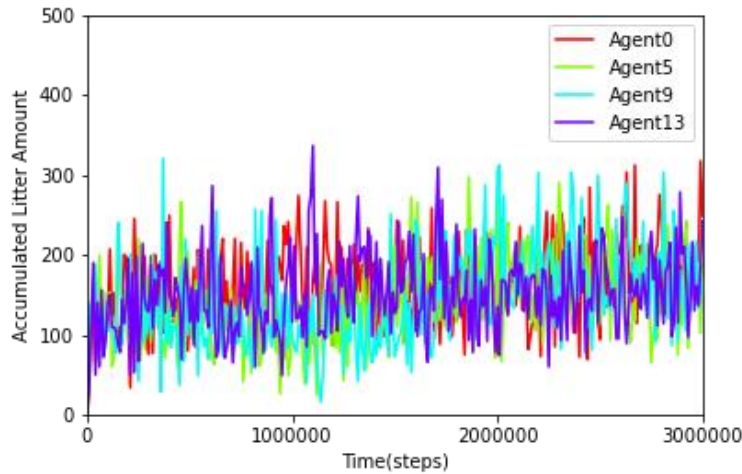
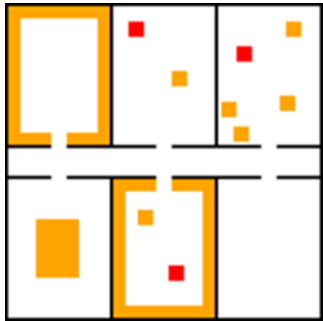
行動が異なる2つのグループに**自律的に**分かれた

各グループのイベント処理量

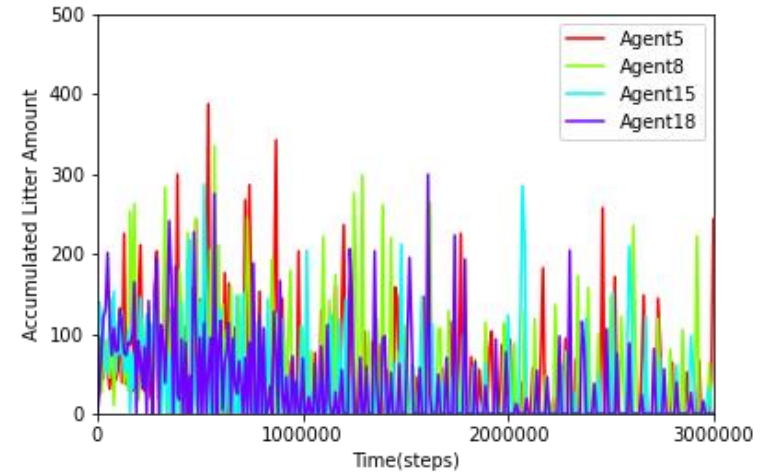
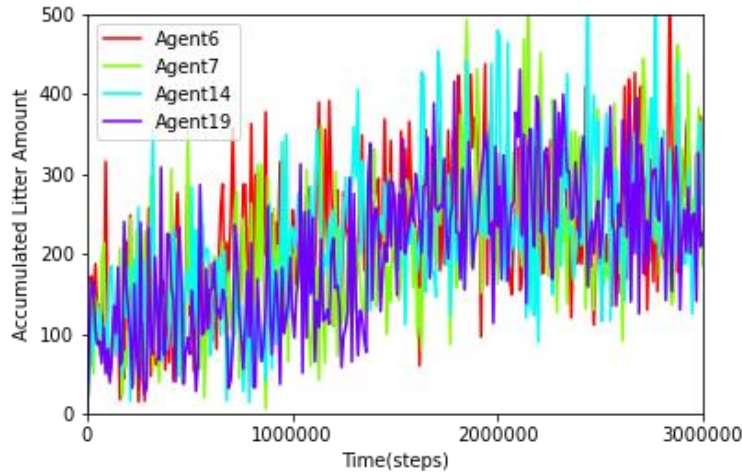
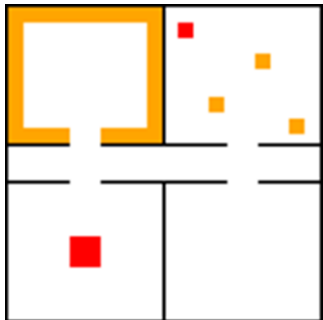
B-group 4体

ES-group 4体

実験1

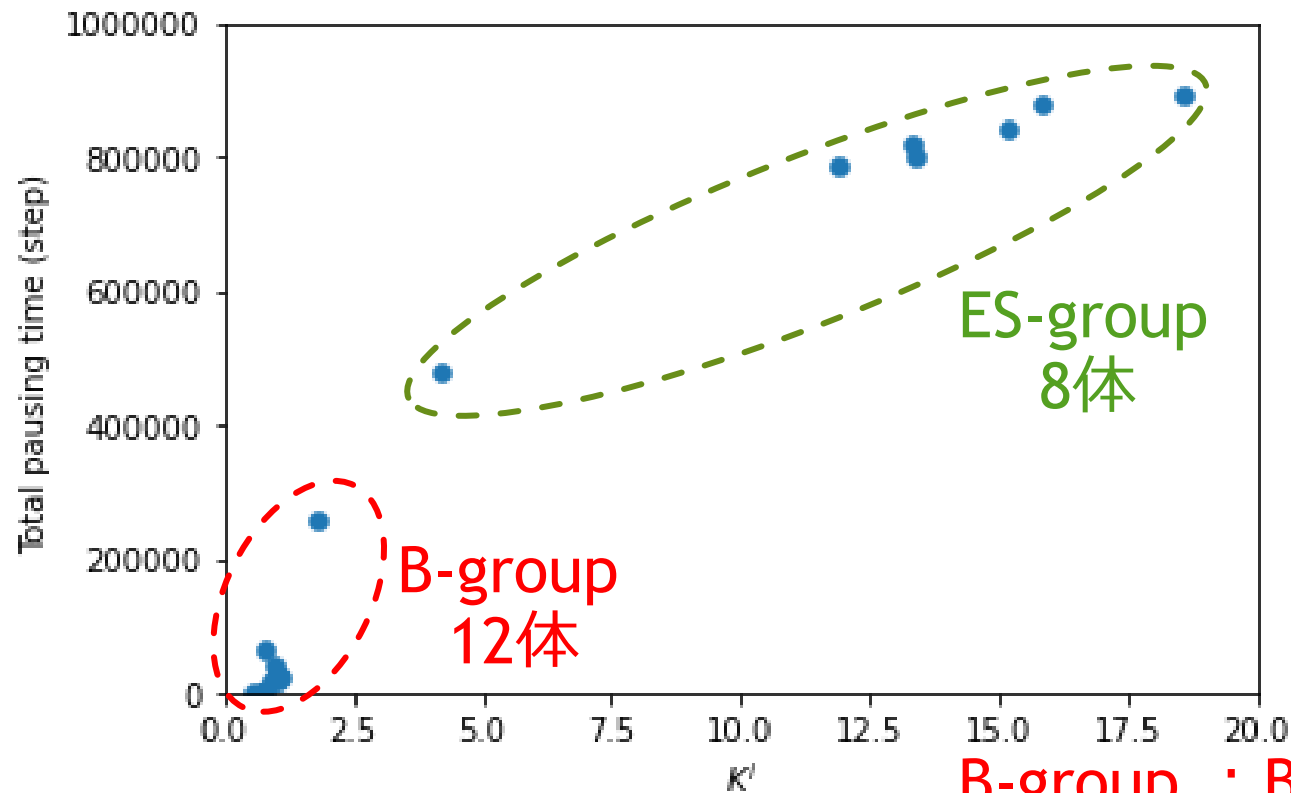


実験2

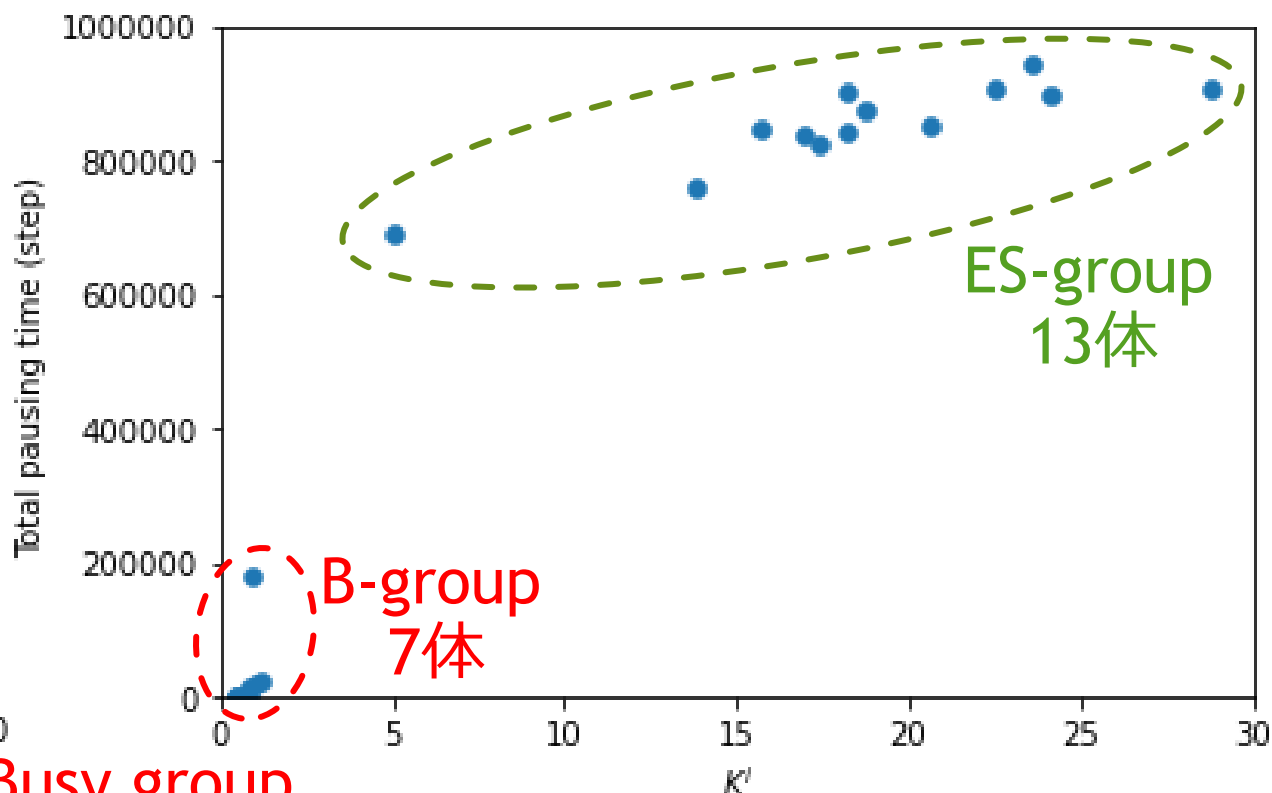


グループによってイベント処理量が異なる

パラメータ K^i と待機時間



実験1



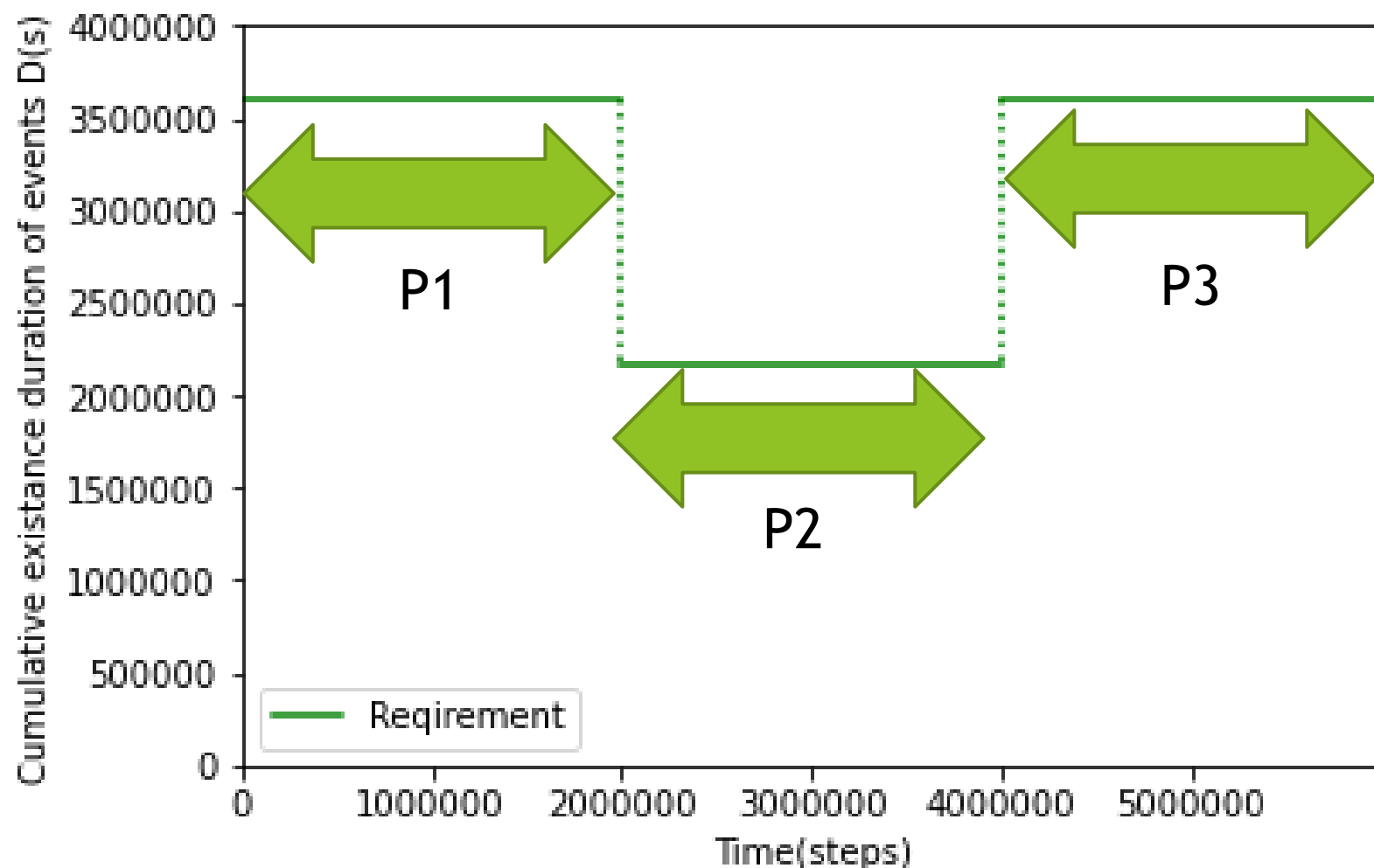
実験2

B-group : Busy group

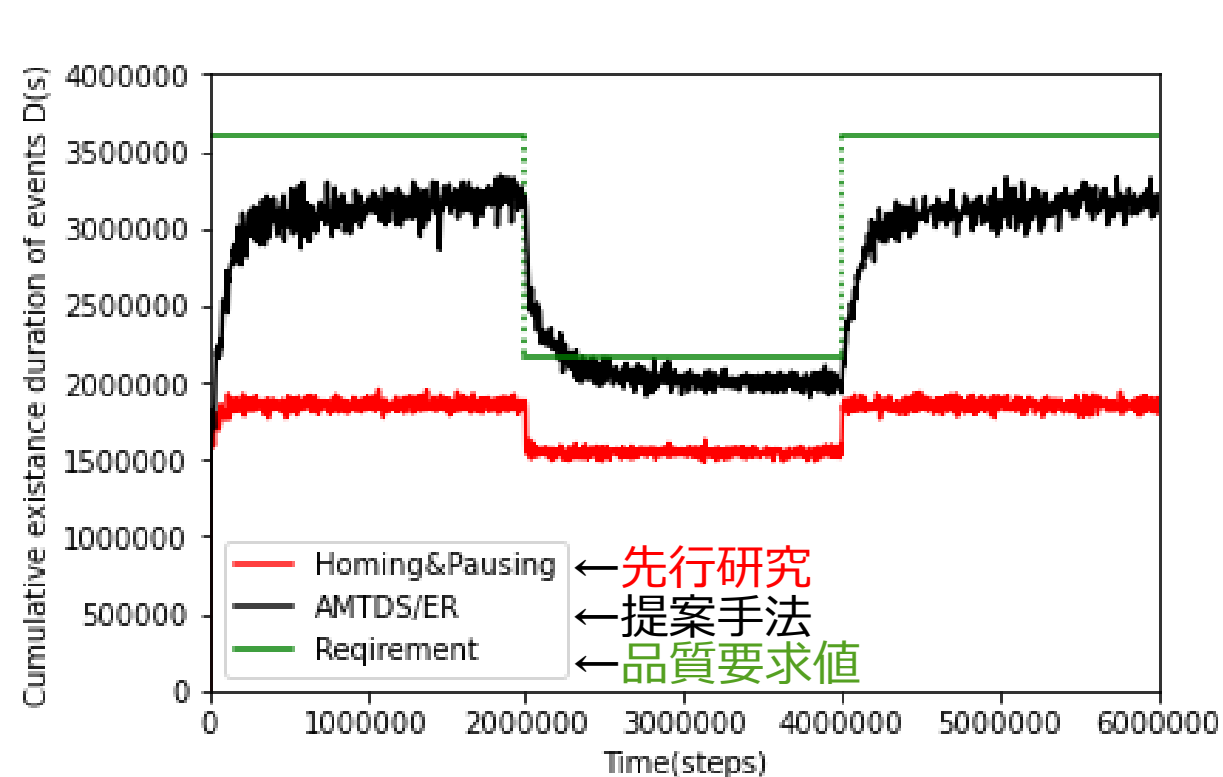
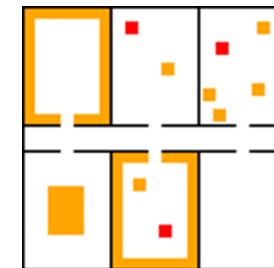
ES-group : Energy save group

環境のイベント量と品質要求値を考慮し、
ボトムアップ的に2つのグループに分かれた

品質要求値の変化に対する対応

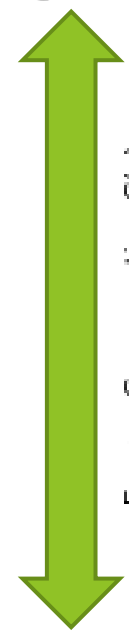


実験結果(実験3)

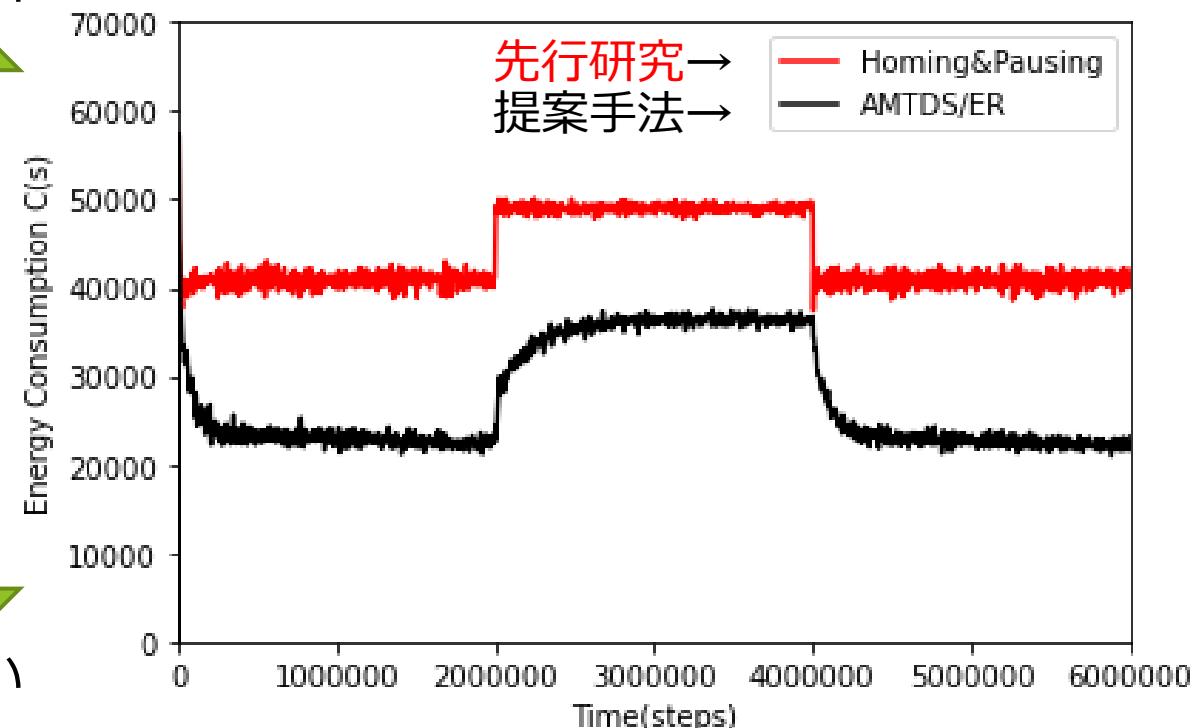


イベント残存時間の総和 $D(s)$

悪い



良い



総エネルギー消費量 $C(s)$

品質要求値が変化しても対応できる

不要なエージェントの停止

- 要求条件を満たしたうえで環境巡回に必要なエージェントが十分のときは、一部のエージェントの巡回の停止が有効
- K^i の大きいエージェントから停止していく
- 停止のメリット
 - 停止したエージェントを他の環境で利用できる
 - それらのエージェントを待機させ、巡回エージェントが故障時に利用できる
 - 定期点検や交代運用を行ってシステムを延命できる

Deactivation on Count

- $D_{int}(> 0)$ ステップごとに以下の式を満たしたら停止する手法

$$x_{pause} \geq N_{deact} \text{ and } \max_{i \in A} K^i \geq K_{deact}$$

x_{pause}	:	D_{int} ステップ内にPausingを試みた回数
N_{deact}, K_{deact}	:	閾値

※Pausingを試みて, 0ステップでも x_{pause} にカウントする

Deactivation on Time

- $D_{int}(> 0)$ ステップごとの待機時間を用いて停止するか決める手法
- エージェントがB-groupとES-groupに役割分担が行われた後で、エージェントを停止しても、他のエージェントがその分をカバーできることが保証できたら停止する

通常変化量 d^i の導入

- $\beta_s (> 1)$ を用いて, 以下の式で求められる D_{learn} ステップごとに d^i を学習する

$$D_{learn} = \frac{D_{int}}{\beta_s}$$

- $|P_n^i - P_{n-1}^i| \leq \beta_s \times d^i$ を満たすエージェントの集合を A_n とし, 以下の式を満たしたら停止する

$$\sum_{i \in A} P_n^i \geq D_{int}$$

P_n^i : D_{int} ステップごとのエージェント i の総待機時間

通常変化量 d^i の導入

- $\beta_s (> 1)$ を用いて, 以下の式で求められる D_{learn} ステップごとに d^i を学習する

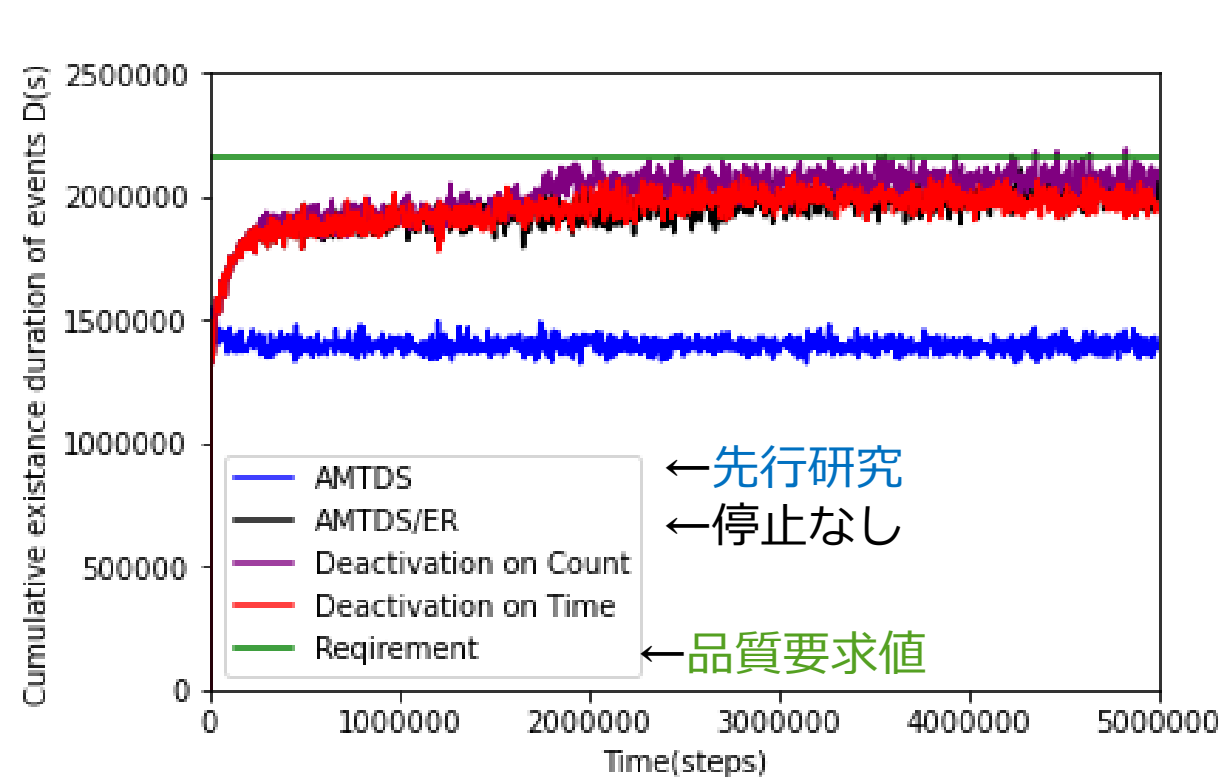
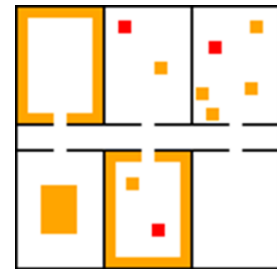
$$D_{learn} = \frac{D_{int}}{\beta_s}$$

- エージェント i の待機時間の通常変化量 d^i を以下の式に従って更新する

$$d^i \leftarrow \alpha_d \times |T_n^i - T_{n-1}^i| + (1 - \alpha_d) \times d^i$$

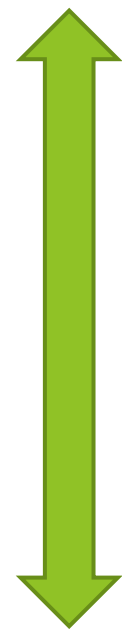
$|T_n^i - T_{n-1}^i|$: D_{learn} ステップごとの総待機時間の変化量

実験結果(実験5)

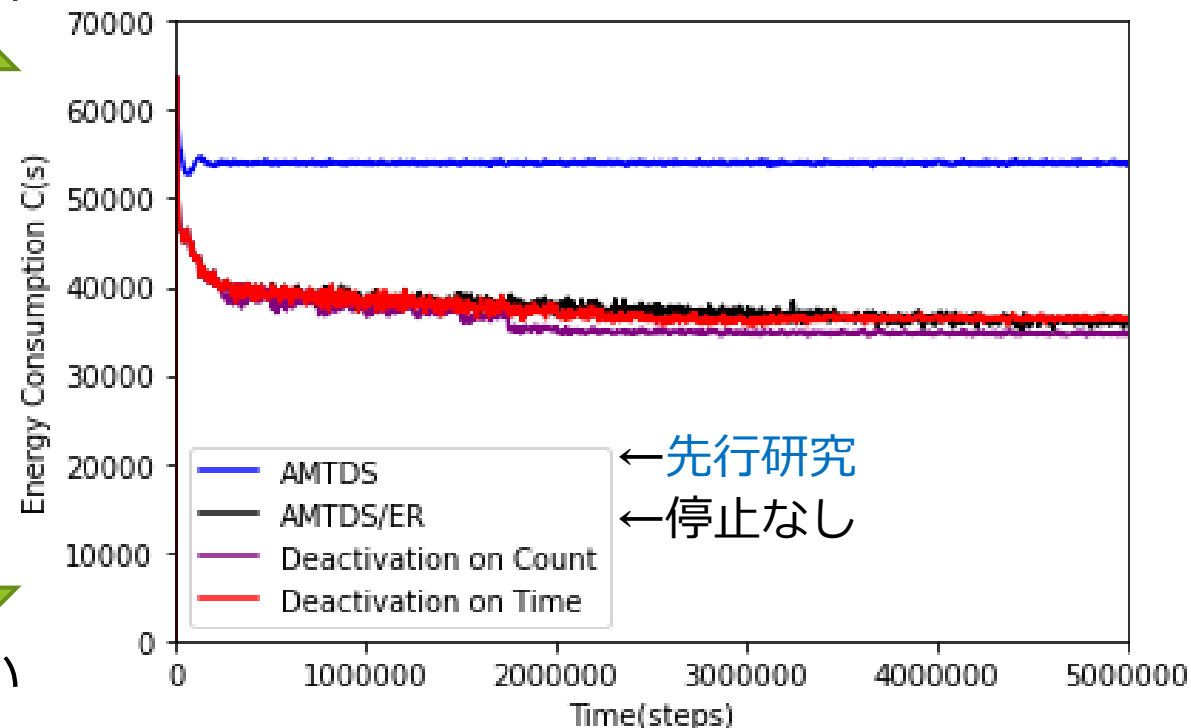


イベント残存時間の総和 $D(s)$

悪い



良い

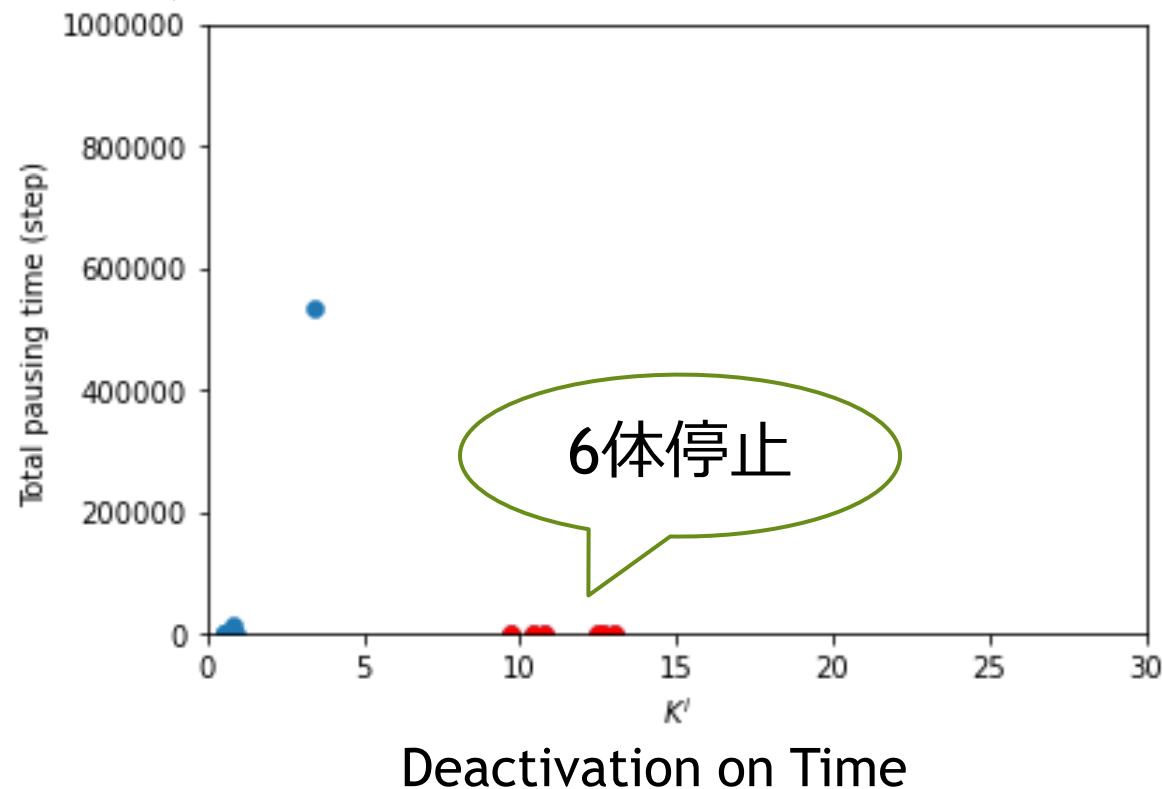
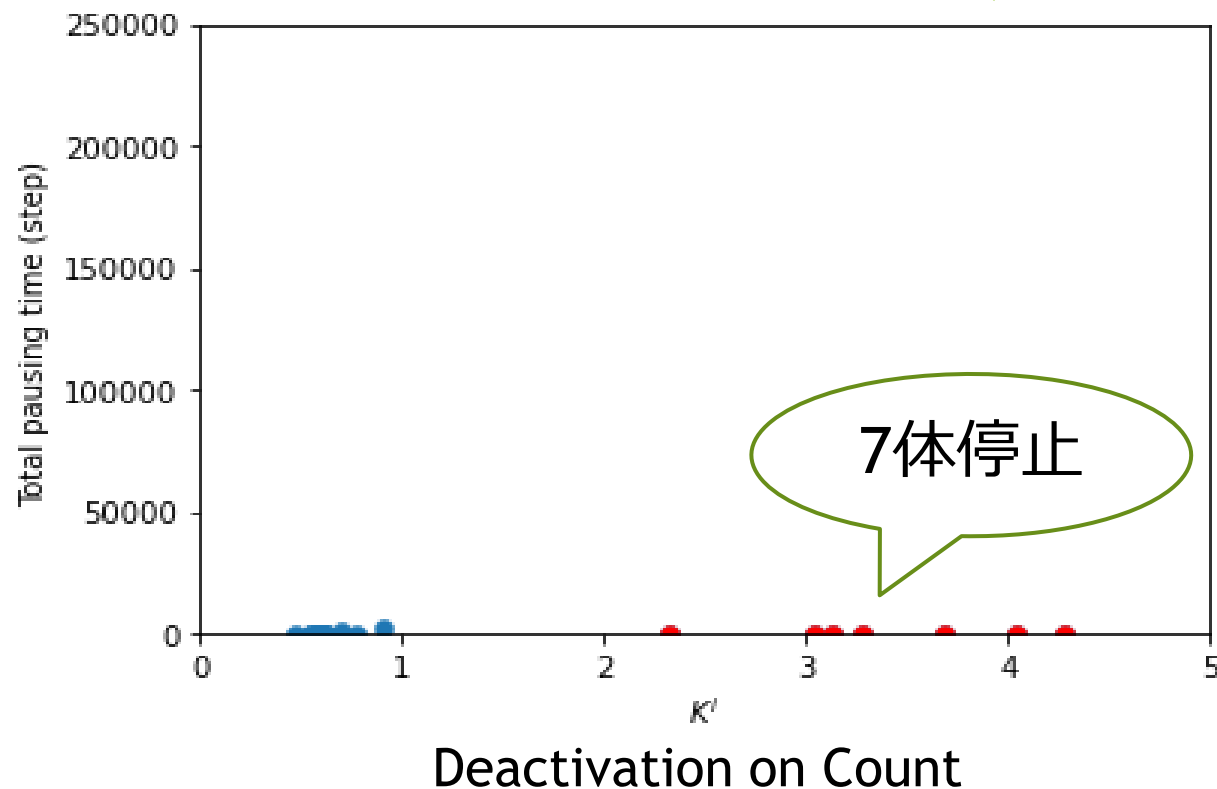


総エネルギー消費量 $C(s)$

要求品質を超えずにエージェント数を減らせた

実験結果(実験5)

赤：停止エージェント
青：非停止エージェント



待機時間が**多い**エージェントを停止できた

AMTDS/ERC

- AMTDS/EDCとAMTDS/ERを合わせた手法
- AMTDS/ERにイベント発生確率の学習と,
エージェント間の交渉を追加

学習パラメータ C_{rate}^i の導入

- エージェントのイベント発生確率の学習が
どの程度ずれているかを示すパラメータ
- 充電基地出発～充電基地帰還までのノード v の期待値
と実際のイベント処理数の合計を S_{est}^i , S_{real}^i とする

$$C_{rate}^i \leftarrow \alpha_{rate} \times \frac{S_{real}^i}{S_{est}^i} + (1 - \alpha_{rate}) \times C_{rate}^i$$

K^i の更新方法の変更

$$K^i \leftarrow (1 - \alpha) \times K^i + \alpha \times \frac{D_{req}}{E^i \left(D_t(V_R^i) \right) \times \frac{|V_{access}|}{|V_R^i|} \times C_{rate}^i}$$

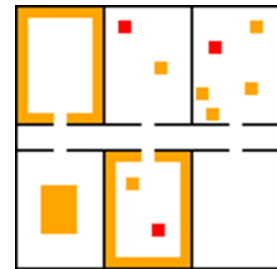
V_R^i : i の責任ノード集合
 V_{access} : 環境全体から障害物を除いたノード集合
 $E^i \left(D_t(V_R^i) \right)$: i の t における V_R^i 内の総イベント量の推定値

未来のイベント予測量と 要求条件の比較の変更

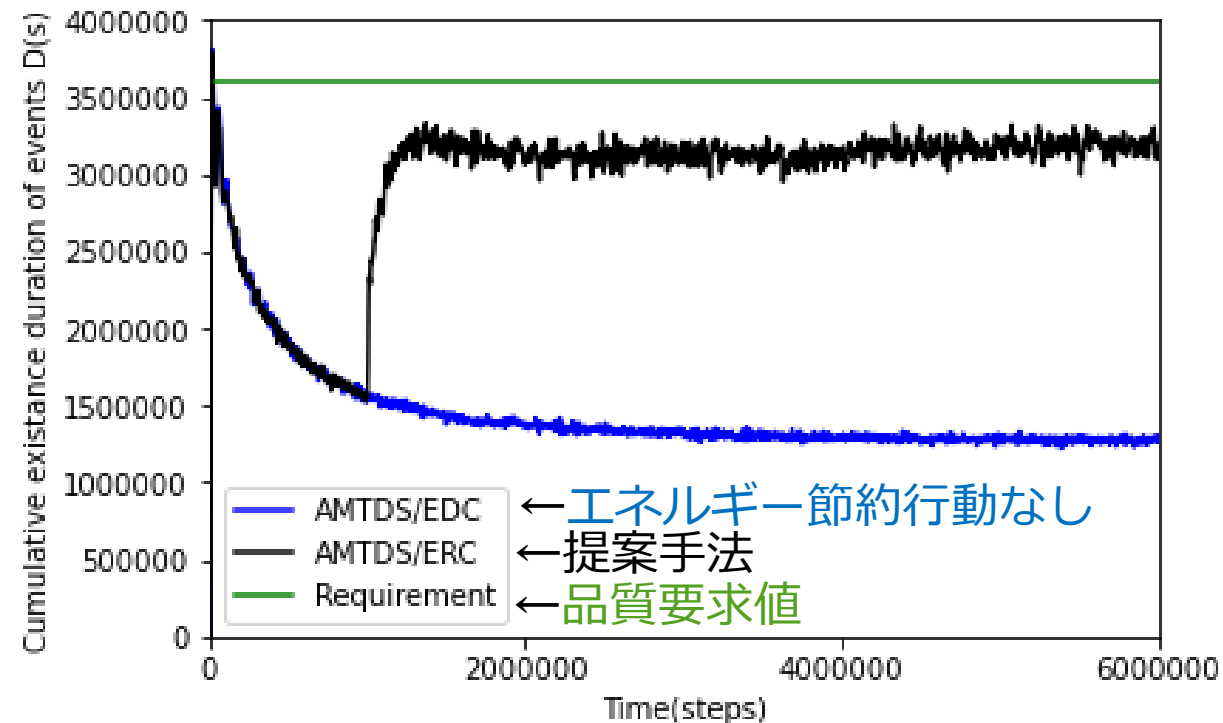
以下の式を満たす最大の T を待機時間とする

$$E^i \left(D_{t+T} (V_R^i) \right) \times \frac{|V_{access}|}{|V_R^i|} \times C_{rate}^i \div K^i \leq D_{req}$$

実験結果(AMTDS/ERC)



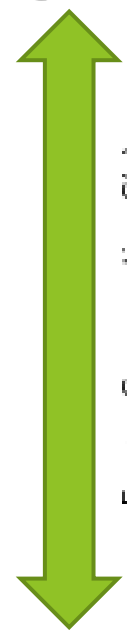
品質要求値: $(1,000/\text{step} \times 3,600) = 3,600,000$



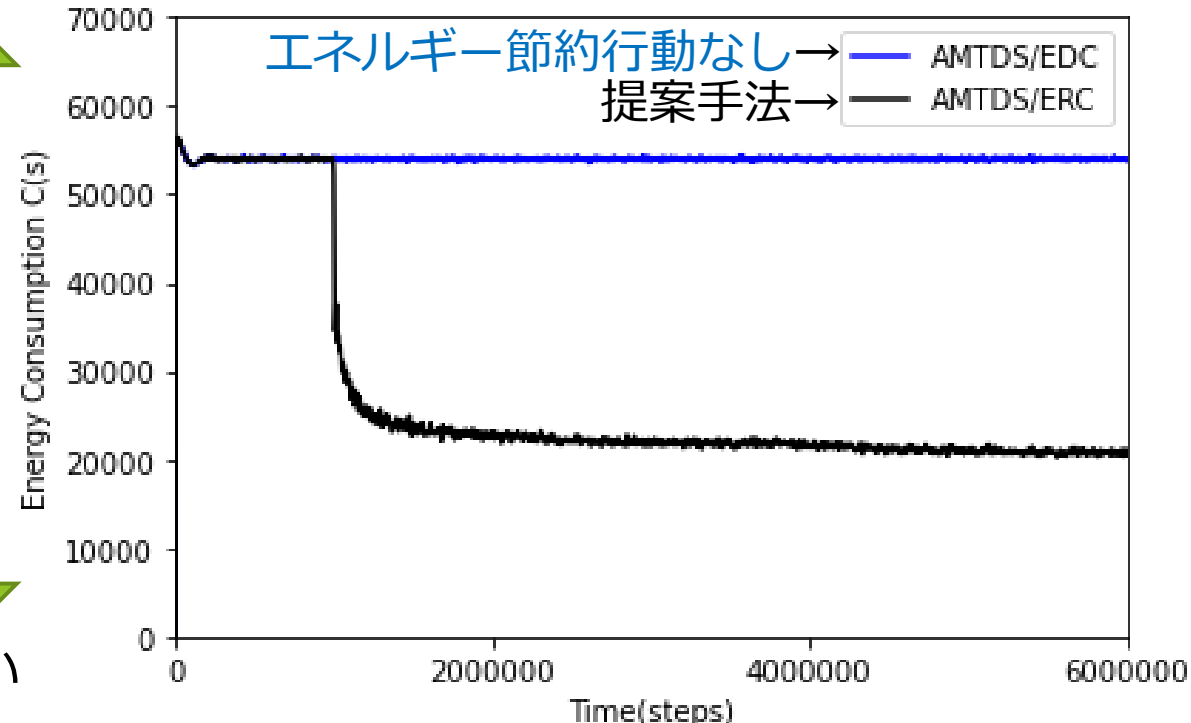
イベント残存時間の総和 D(s)

1275539.7
3186541.5
+149.8%

悪い



良い

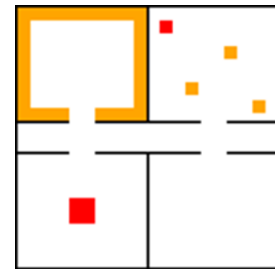


総エネルギー消費量 C(s)

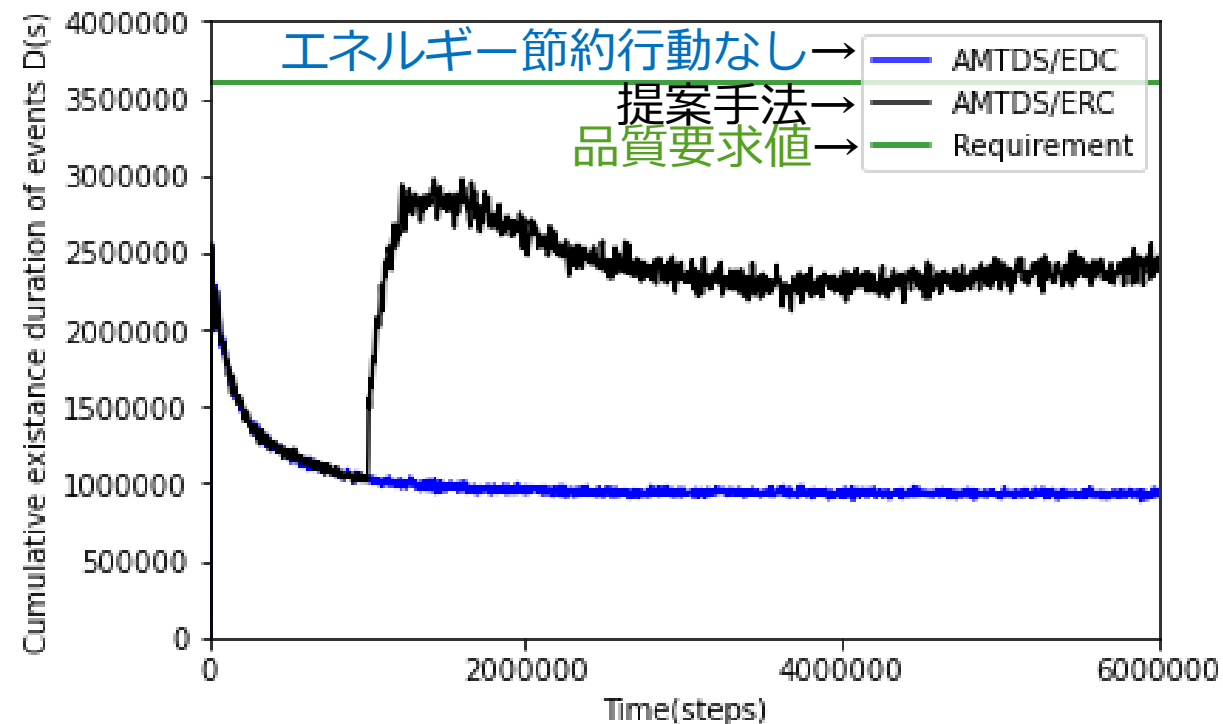
54002.0
20849.4
-61.4%

提案手法は品質要求値に近く, エネルギー消費量が小さい 57

実験結果(AMTDS/ERC)



品質要求値: $(1,000/\text{step} \times 3,600) = 3,600,000$



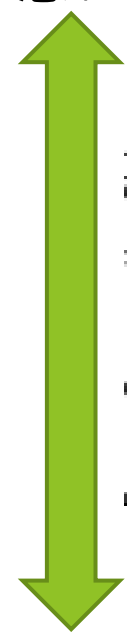
イベント残存時間の総和 D(s)

933158.7

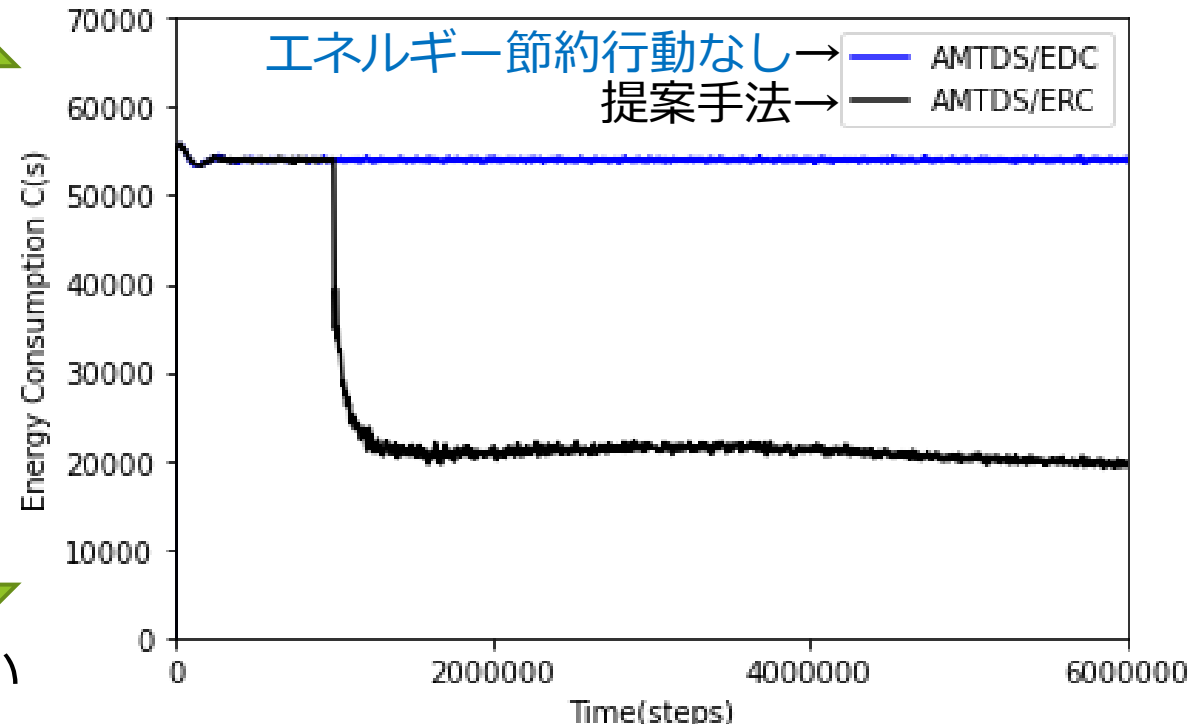
2380238.1

+155.1%

悪い



良い



総エネルギー消費量 C(s)

54001.2

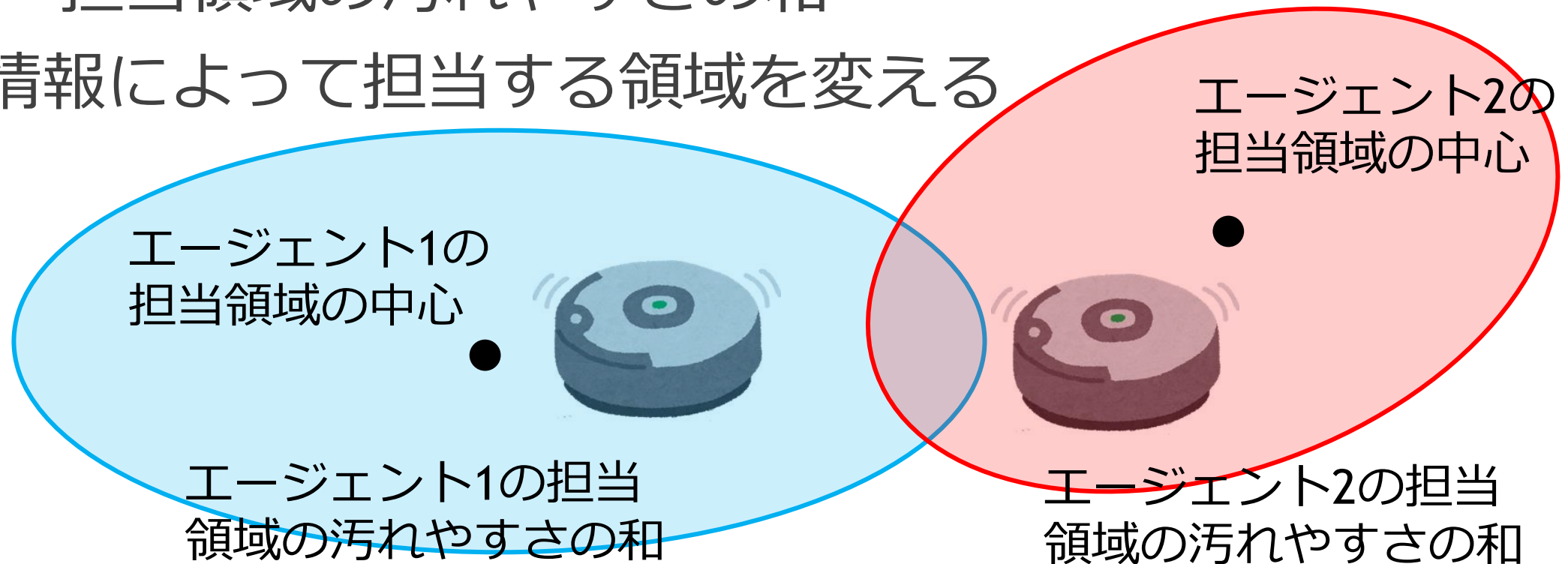
19945.2

-63.1%

エネルギー消費量を最大限削減できたとはいえない

AMTDS/EDC (先行研究)

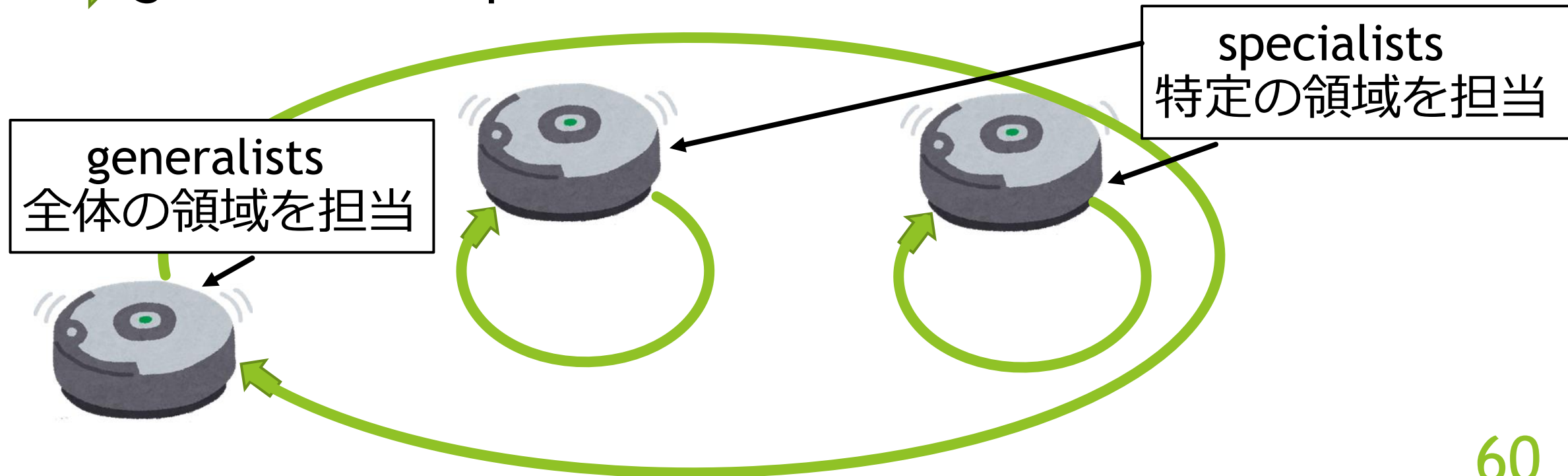
- 交換する情報
 - ・ 担当領域の中心
 - ・ 担当領域の汚れやすさの和
- 情報によって担当する領域を変える



AMTDS/EDC (先行研究)

- この手法のメリット

➡ generalistsとspecialistsに自然に分かれた

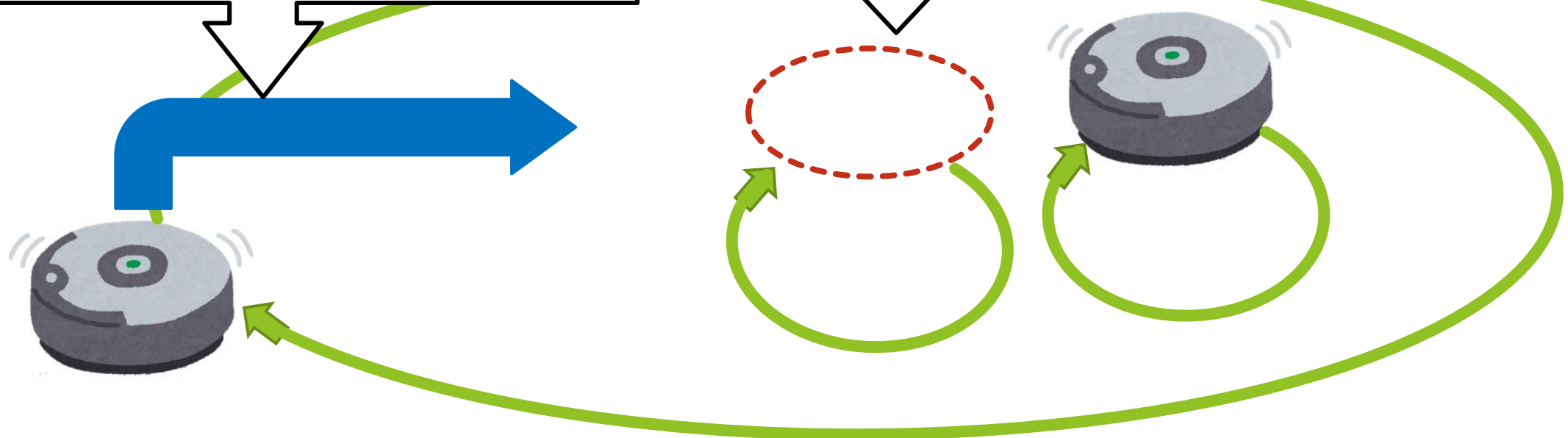


AMTDS/EDC (先行研究)

- generalistsは, specialistsがいなくなるとそこに入る

汚れやすさが増えることによって,
specialistsがいなくなったことに
気付いてそこに入る

specialistsが充電の
ためにいなくなる



先行研究

AMTDS ← 2013年

(Adaptive meta target decision strategy)



2014年 → AMTDS/LD
汚れやすさの学習

AMTDS/ESC ← 2020年
エネルギー抑制



2017年 → AMTDS/EDC
コミュニケーション

①AMTDS/ER

汚れやすさの学習,
コミュニケーション



2022年 → AMTDS/THE
計画停止

②AMTDS/ERC