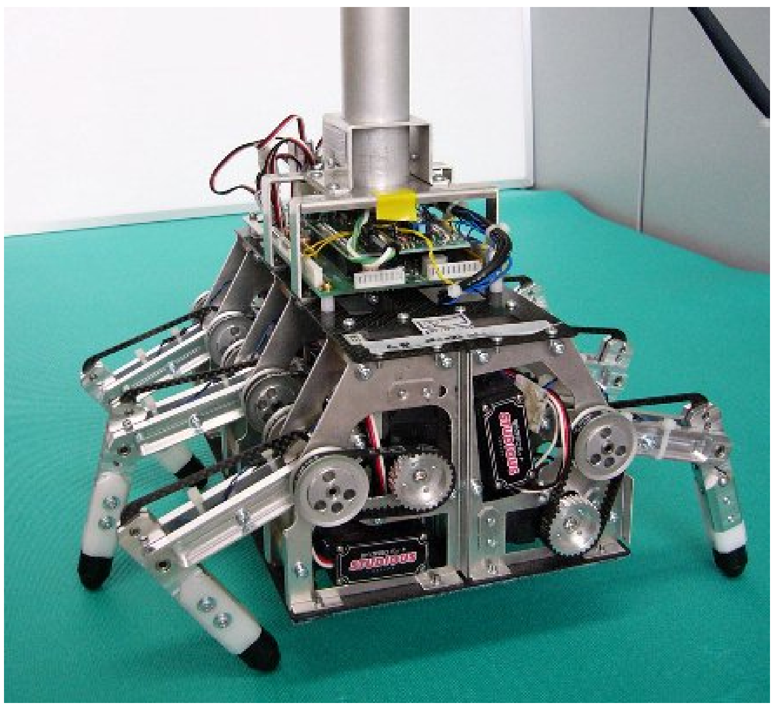
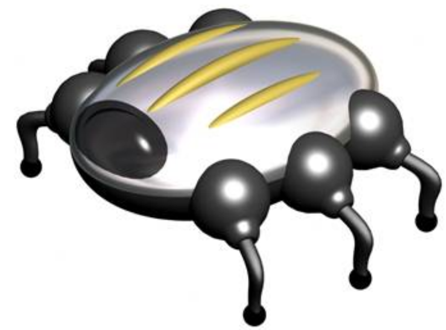
ランダムタイリングを用いた多次元状態-行動の強化学習

－Gibbs-Samplingを用いた行動選択方法の提案－

木村　元
九州大学　大学院工学研究院海洋システム工学部門





http://sysplan.nams.kyushu-u.ac.jp/gen/index.html

ロボットの知能化

→ 強化学習: 試行錯誤を通じて制御規則を獲得

学習アルゴリズムに対する要請:

1）とにかく単純な計算処理
- ●プログラムコードが短いこと
- ●自然言語で簡潔に表現できるのが望ましい

アルゴリズム中に「微分記号」など論外！

2）安定した学習動作
- ●問題に依存せずに実装可能であること
- ●パラメータ設定にセンシティブではないこと

⇒ Actor-Critic　Q-learning

ロボットの知能化

→　強化学習：試行錯誤を通じて制御規則を獲得

学習アルゴリズムに対する要請：

1）とにかく**単純**な計算処理
- プログラムコードが短いこと
- 自然言語で簡潔に表現できるのが望ましい

アルゴリズム中に「微分記号」など論外！

2）**安定**した学習動作
- 問題に依存せずに実装可能であること
- パラメータ設定にセンシティブではないこと

⇨　Actor-Critic　　Q-learning

ロボットの知能化

→　強化学習：試行錯誤を通じて制御規則を獲得

学習アルゴリズムに対する要請：

1）とにかく**単純**な計算処理
- プログラムコードが短いこと
- 自然言語で簡潔に表現できるのが望ましい

アルゴリズム中に「微分記号」など論外！

2）**安定**した学習動作
- 問題に依存せずに実装可能であること
- パラメータ設定にセンシティブではないこと

⇒　~~Actor-Critic~~　　**Q-learning**

本研究の目的

高次元連続状態-行動空間のロボット強化学習

Q-learning	連続空間 関数近似	ボルツマ 行動選択

・これら基礎的な技術と知見をそのまま継承した実装
・数万ステップで学習させる（Actor-Criticの数倍程度以

解決すべき課題:
1）状態-行動の評価値（Q値）の表現 ←膨大な空間の汎化
2）高次元連続行動空間における行動選択は？

本研究の目的

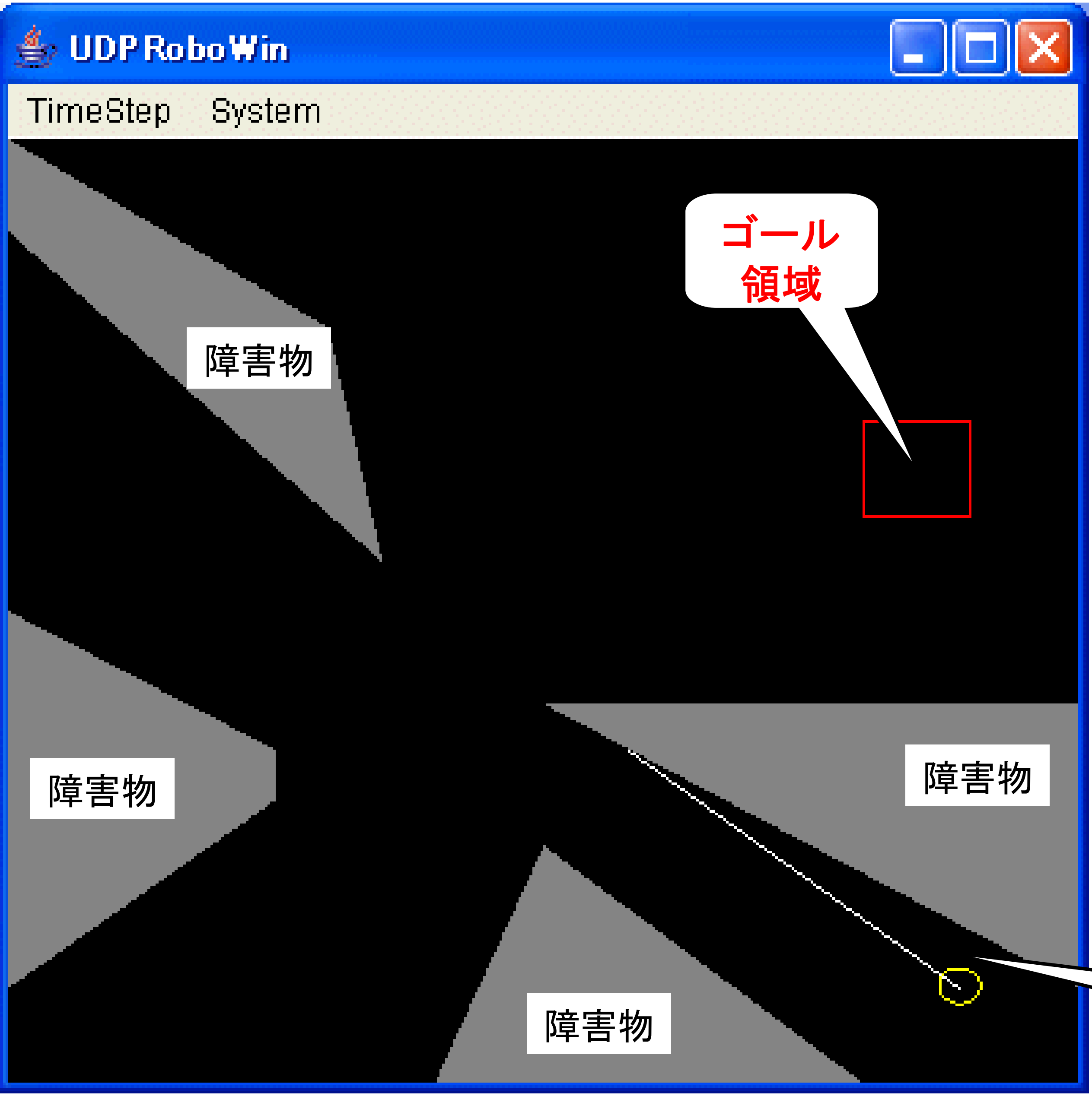
高次元連続状態-行動空間のロボット強化学習

Q-learning	連続空間 関数近似	ボルツマ 行動選択

・これら基礎的な技術と知見をそのまま継承した実装
・数万ステップで学習させる（Actor-Criticの数倍程度以

解決すべき課題:
1）状態-行動の評価値（Q値）の表現 ←膨大な空間の汎化
2）高次元連続行動空間における行動選択は？

対象とする学習問題1：Rod in Maze (Moore 1995)



2次元平面中でスタート状態からゴール領域まで棒を移動

領域中に固定障害物

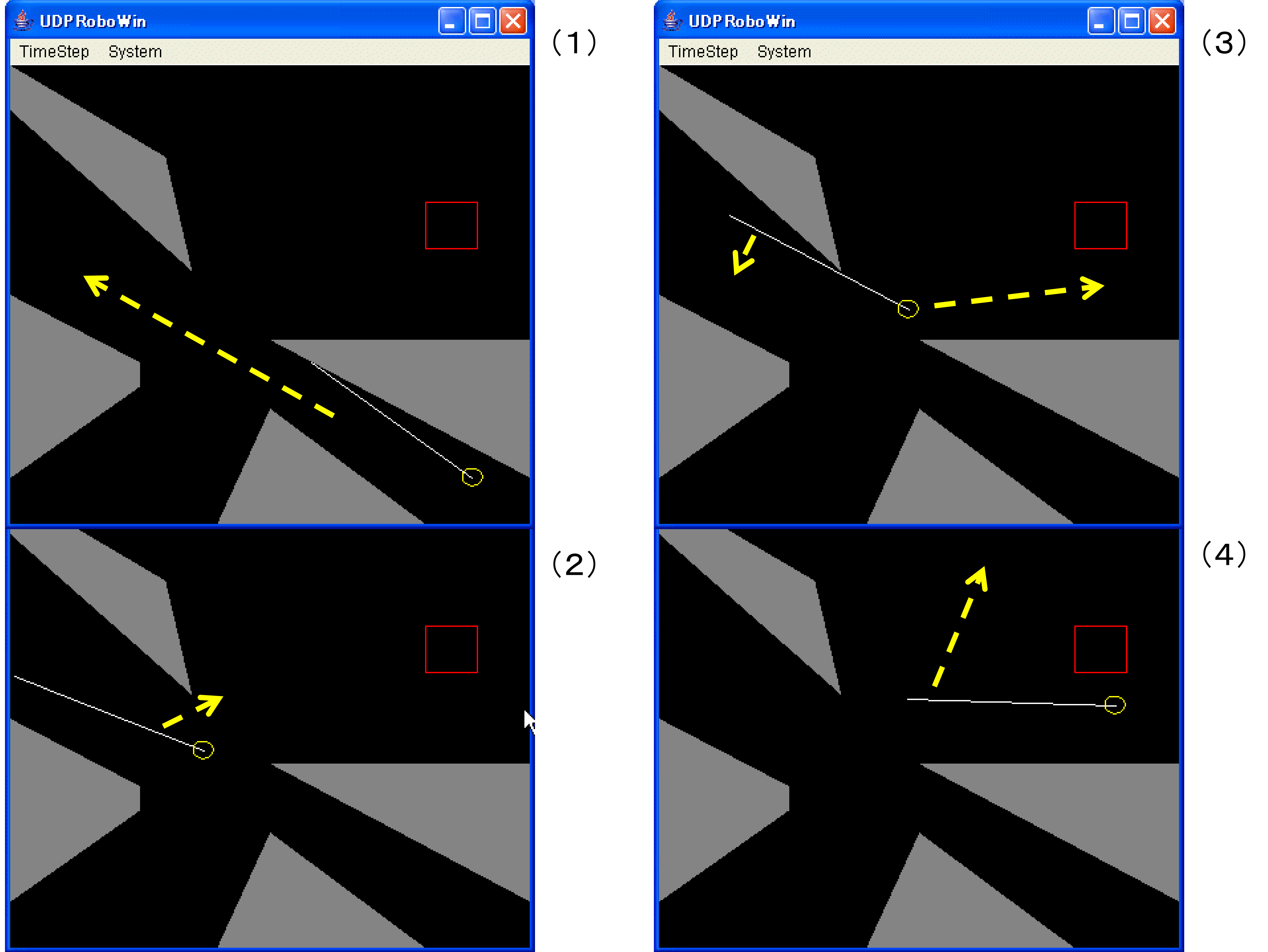
棒の傾き　0～180度のみ

状態：　X, Y座標と角度 θ

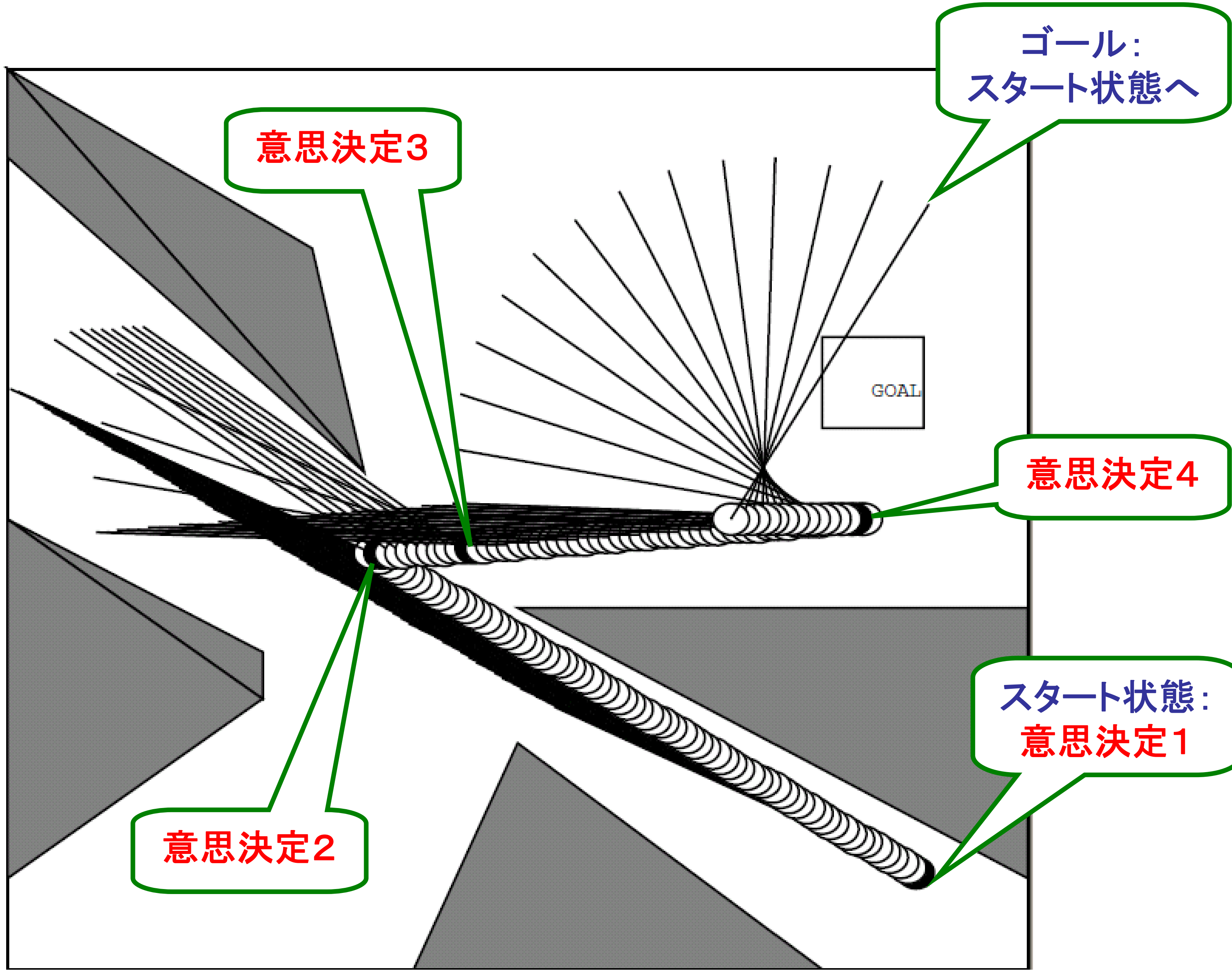
行動：　動作目標X,Y, θ 座標

棒が目標座標へ到達するか、途中で障害物へぶつかると意思決定のイベント発生

ゴールへ到達すると報酬／再び初期状態へもどる

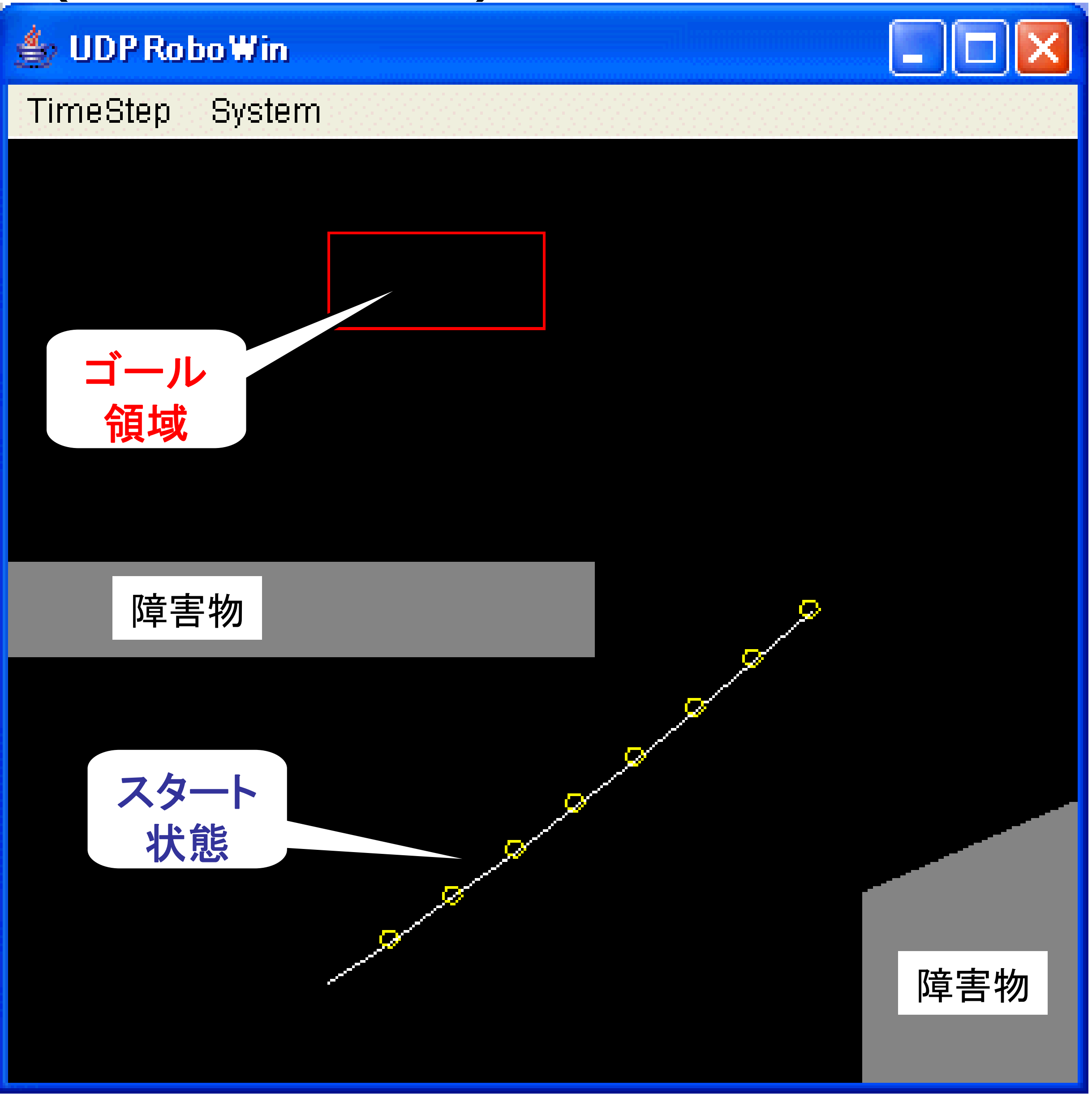

UDP RoboWin
TimeStep System
(1)
(2)
UDP RoboWin
TimeStep System
(3)
(4)

動作例



3次元状態×3次元行動の強化学習問題

対象とする学習問題2：Multi-Joint Arm (Moore 1995)



2次元平面中でスタート状態からゴール領域までアームを移動

領域中に固定障害物

アームは冗長自由度（8関節）

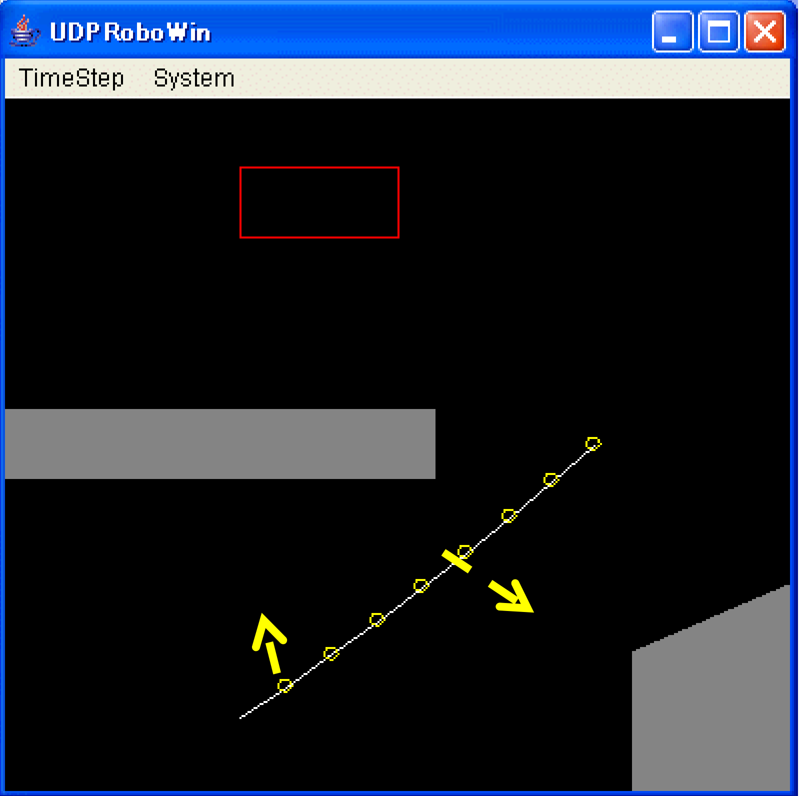
状態：　関節角度θ（8次元）

行動：　目標角度θ（8次元）

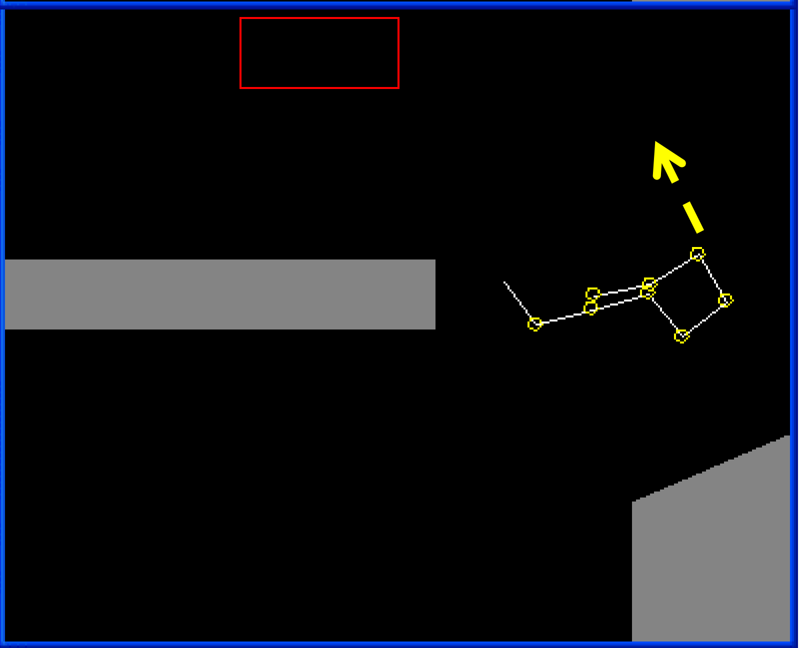
アームが目標角度へ到達するか、途中で障害物へぶつかると意思決定のイベント発生

ゴールへ到達すると報酬／再び初期状態へもどる

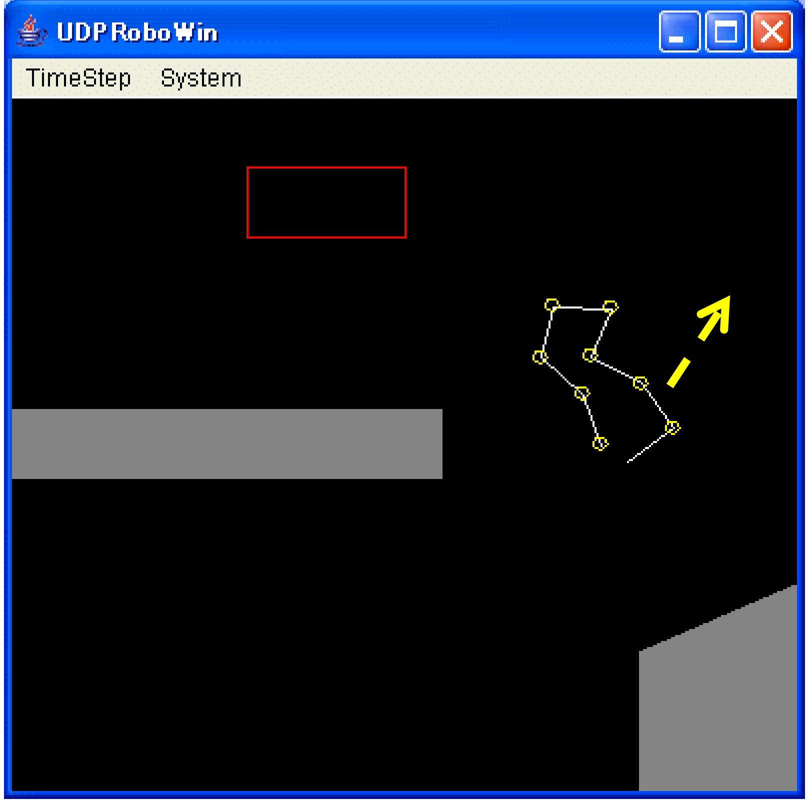
(1)



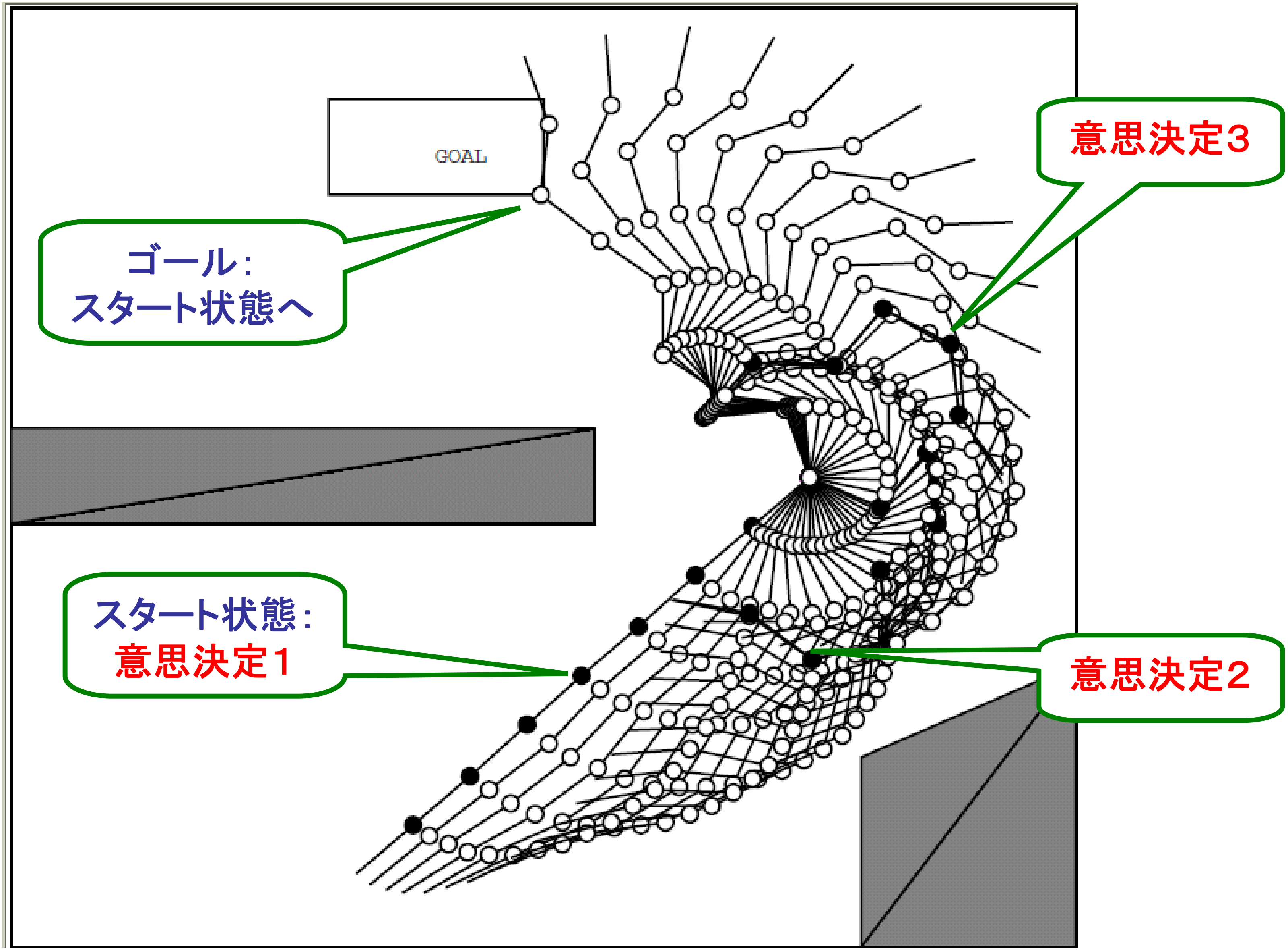
(2)



(3)



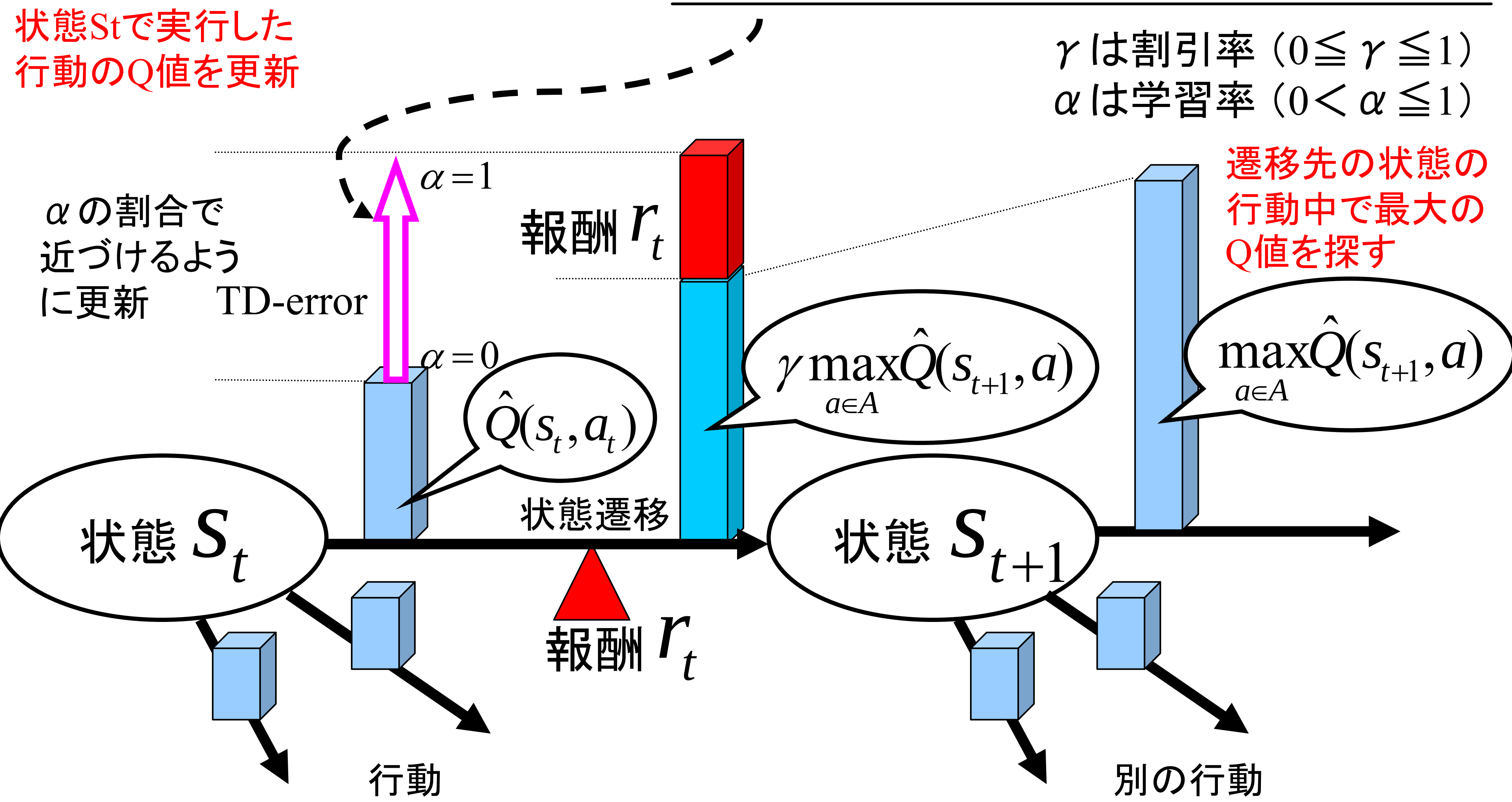
動作例



8次元状態×8次元行動の強化学習問題

Q-learning | 連続空間の関数近似 | ボルツマン行動選択

$$\hat{Q}(s_t,a_t) \leftarrow \hat{Q}(s_t,a_t) + \alpha\left[r_t + \gamma \max_{a\in A}\hat{Q}(s_{t+1},a) - \hat{Q}(s_t,a_t)\right]$$



Q-learning | 連続空間の関数近似 | ボルツマン行動選択

Q-learningの収束定理（Watkins92）

行動選択において全行動を十分な回数選択し，かつ学習率αが

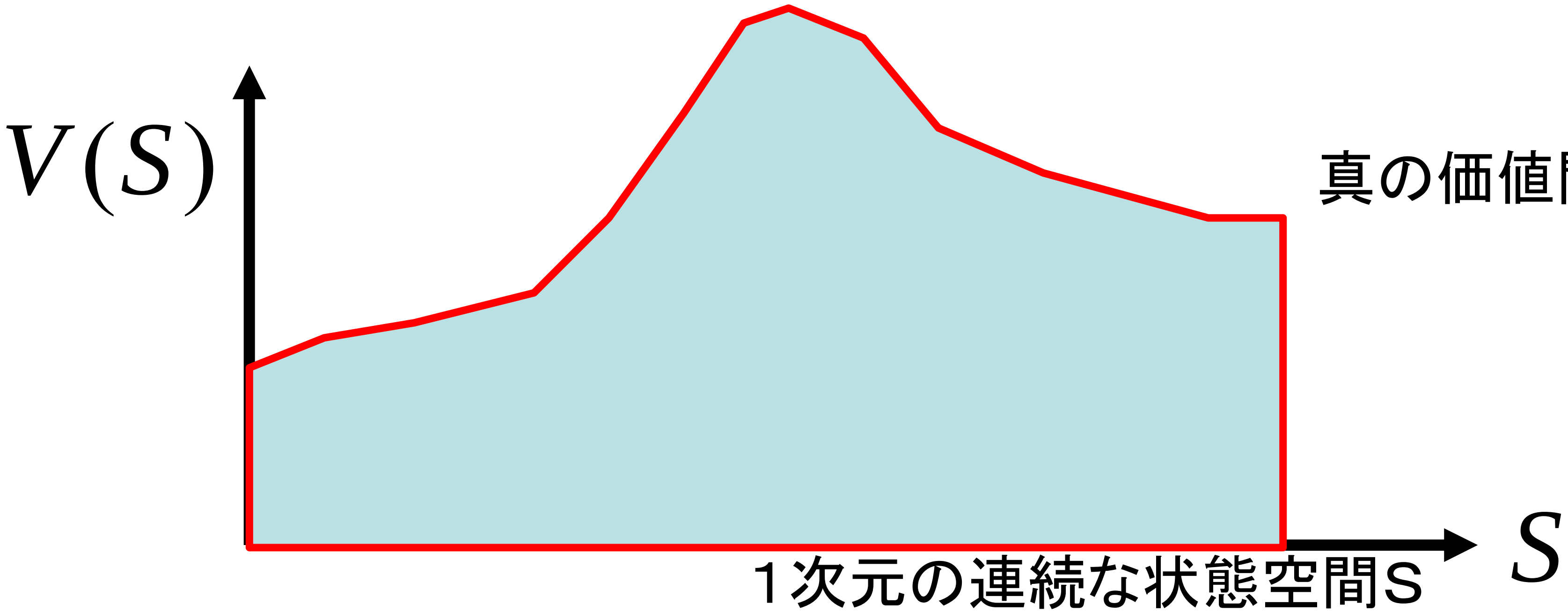
$$\sum_{t=0}^{\infty} \alpha'(t) \to \infty \qquad \text{and} \qquad \sum_{t=0}^{\infty} \alpha(t)^2 < \infty$$

を満たす時間 t の関数になっているとき，アルゴリズムで得るQ値は確率１で最適政策の評価値に概収束する。
ただし環境はエルゴート性を有するMDP

Q-learning

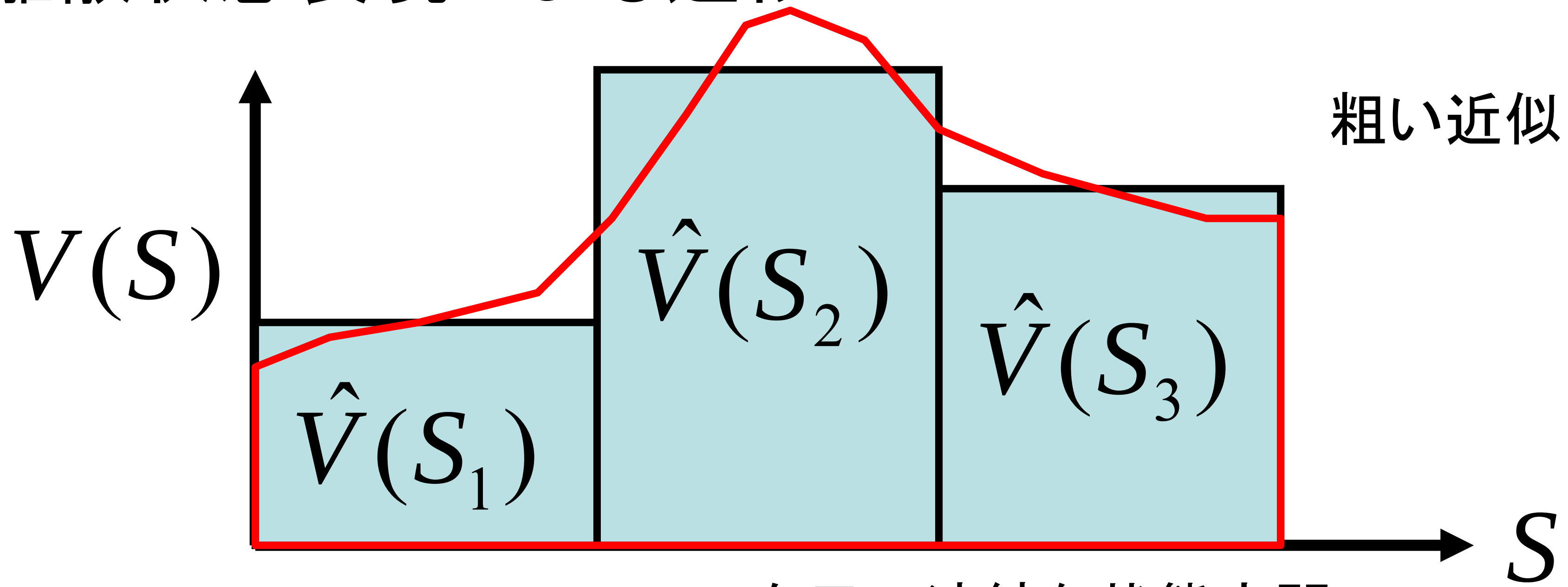
連続空間の関数近似

ボルツマン行動選択

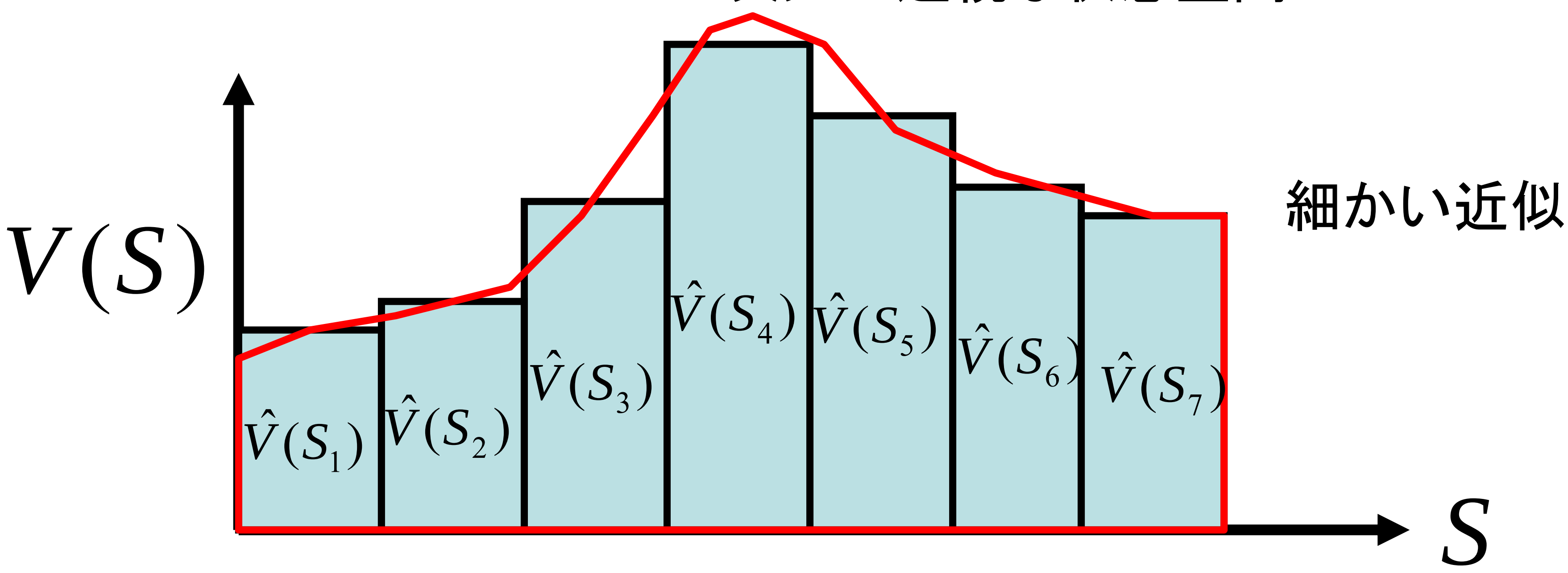


このような離散ではない状態空間のvalueを学習するには？

離散状態表現による近似



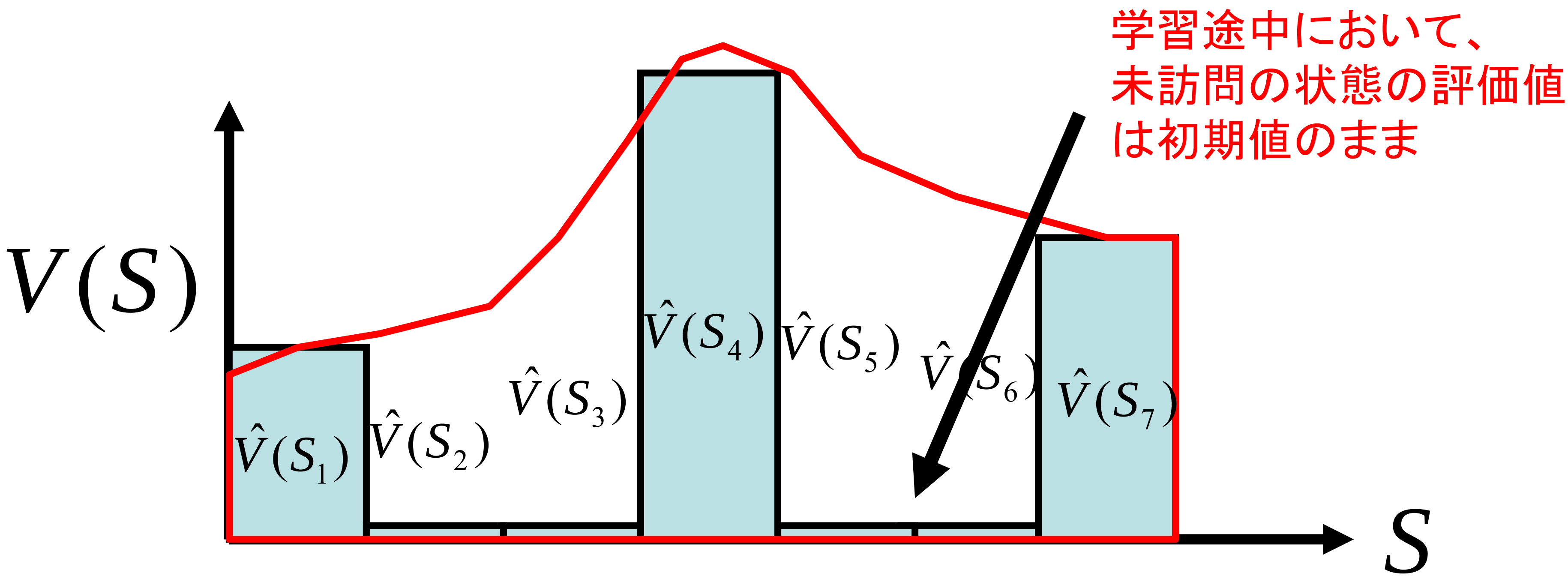
１次元の連続な状態空間S



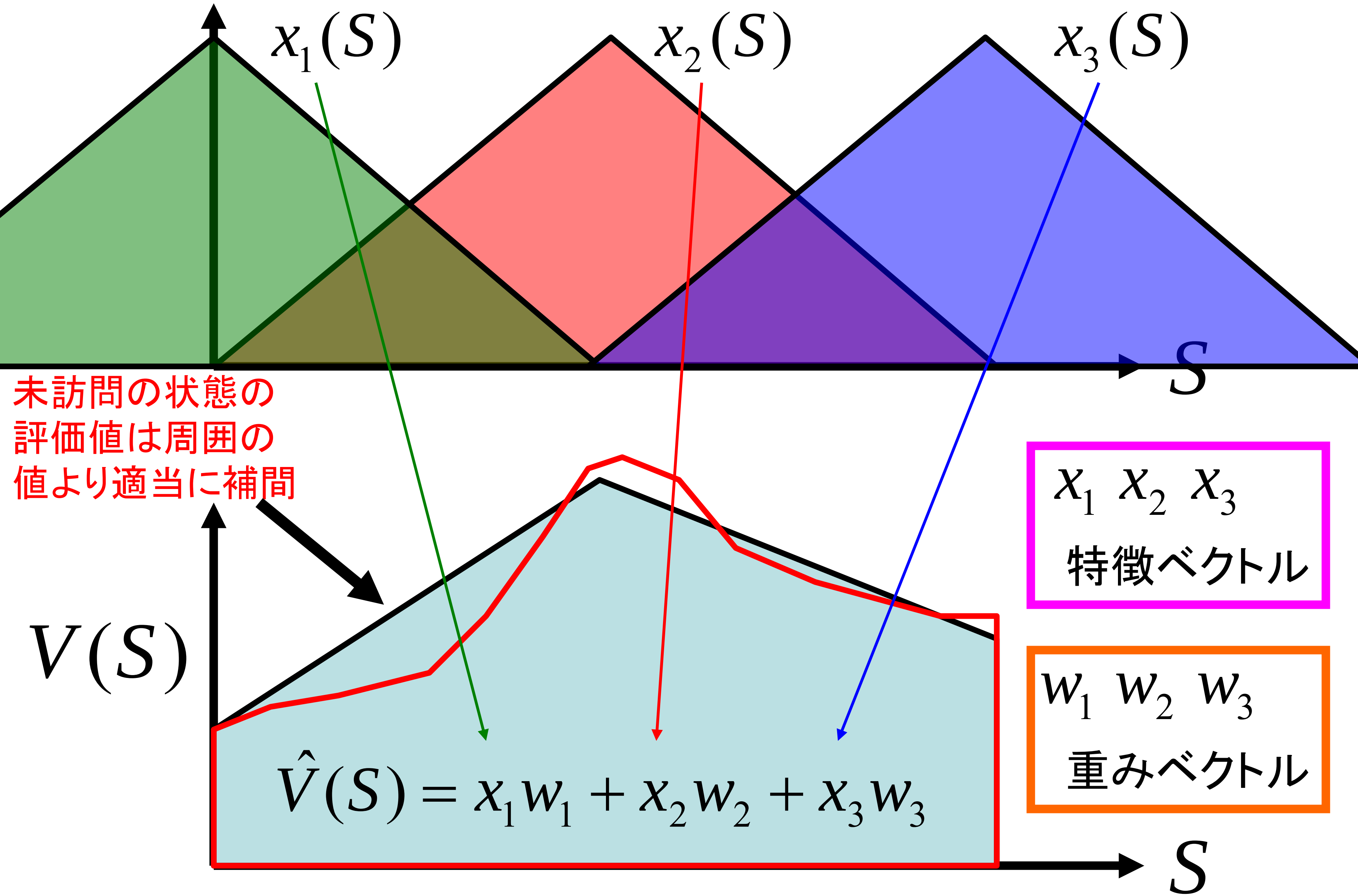
離散状態表現による近似の問題点

粗い近似：
- ●価値関数の学習は早い
- ●非マルコフ性が生じる　→　最適政策が得られない

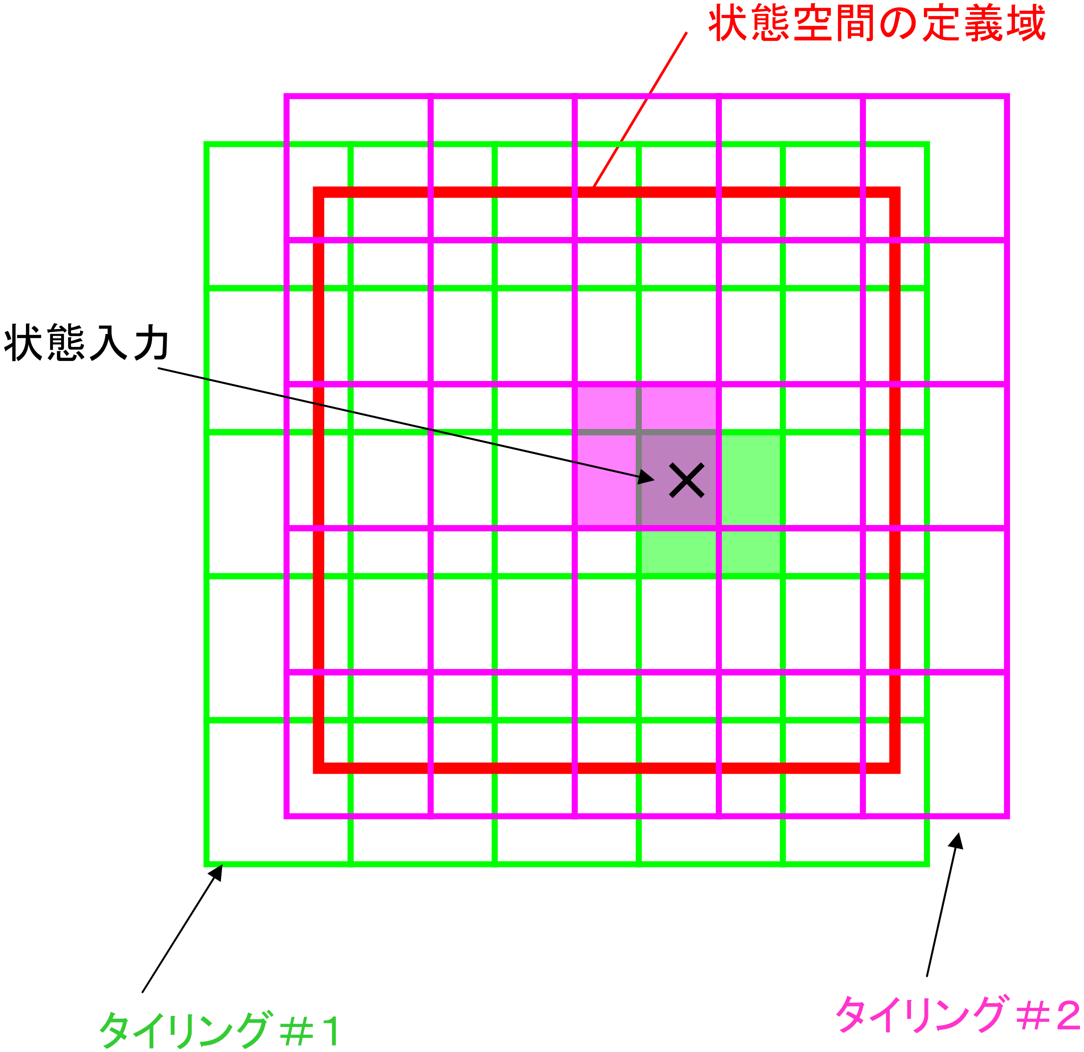
細かい近似
- ●価値関数の学習は指数関数的に遅くなる
- ●特に高次元環境において膨大なメモリーが必要



線形アーキテクチャによる汎化と関数近似



複数タイリングの重ねあわせ(CMAC)による特徴ベクトル生成



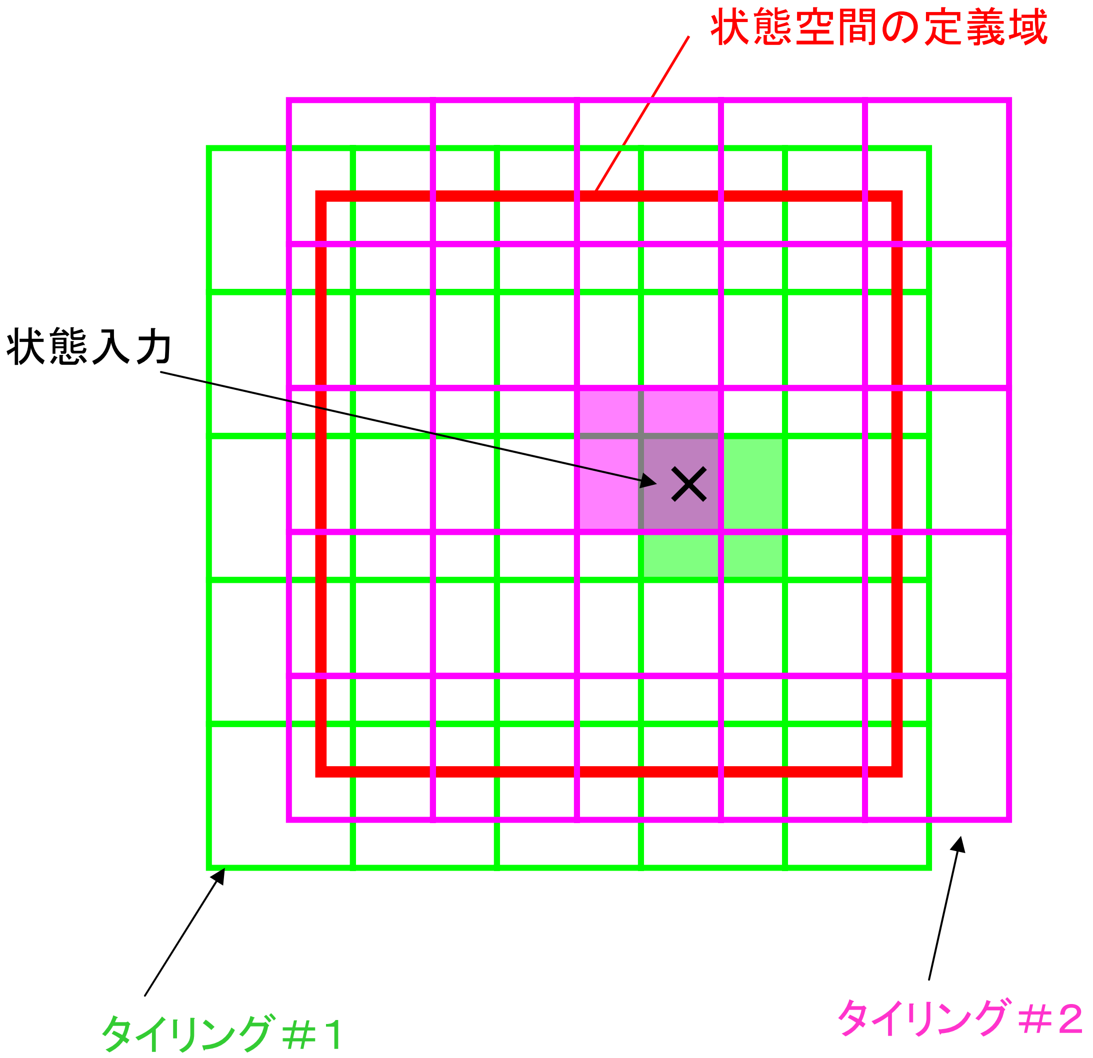
複数のタイリングを
ランダムにずらせながら
複数重ね合わせる

ずらせた時に状態空間の
定義域をカバーしない
領域ができないよう注意

各タイリングが粗くても、
多数重ねることで精度良く
近似表現できる

高次元空間では
メモリー空間が爆発

複数タイリングの重ねあわせ（CMAC）による特徴ベクトル生成



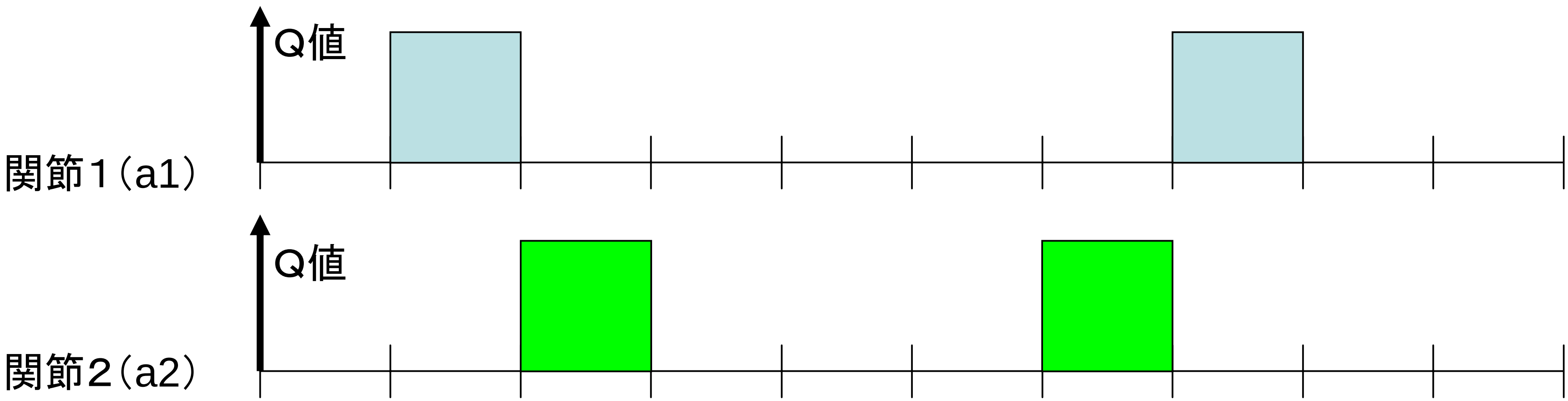
複数のタイリングを
ランダムにずらせながら
複数重ね合わせる

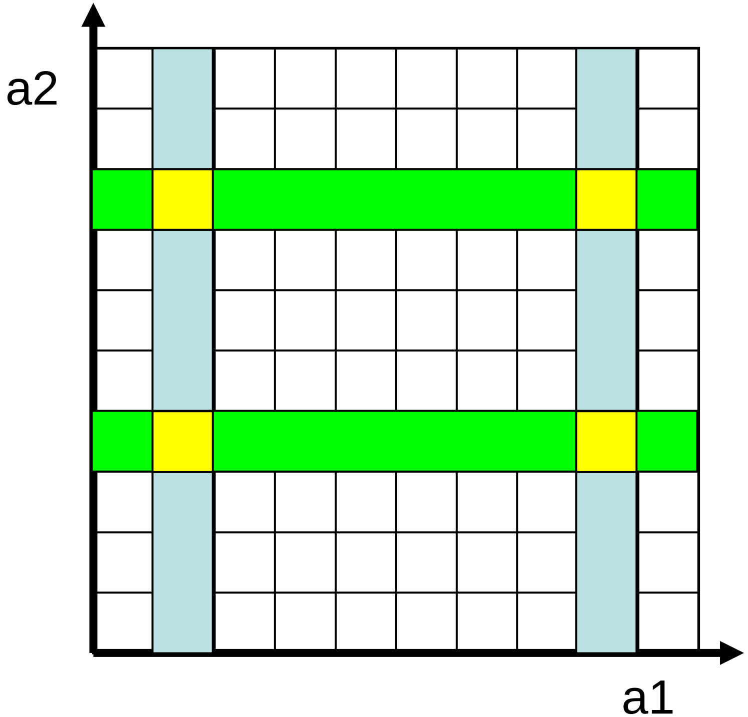
ずらせた時に状態空間の
定義域をカバーしない
領域ができないよう注意

各タイリングが粗くても、
多数重ねることで精度良く
近似表現できる

高次元空間では
メモリー空間が爆発

高次元行動空間の扱いの難しさ:空間の汎化



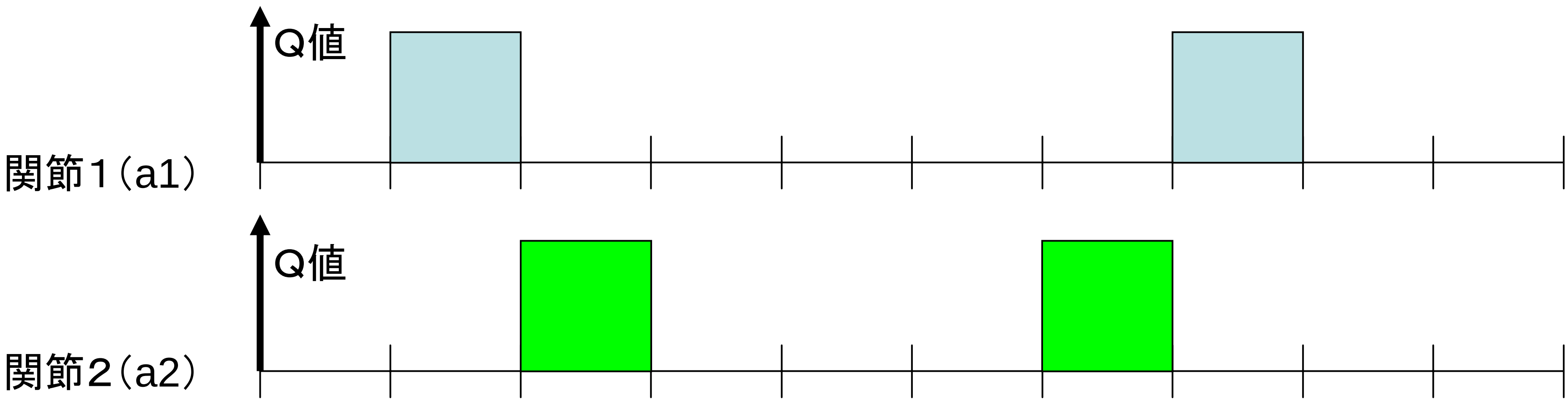


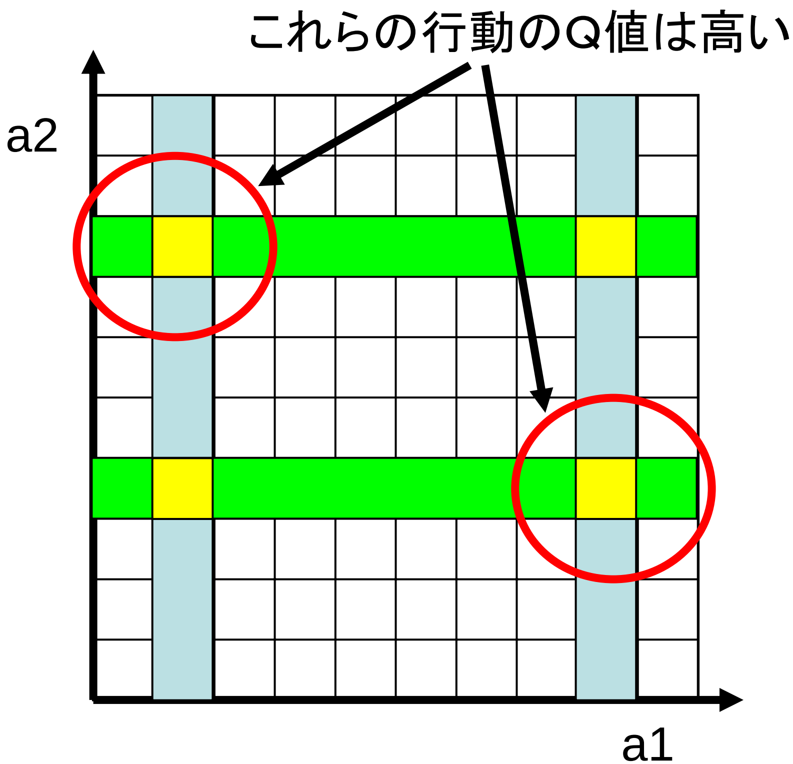
●行動空間の各次元毎にQ値設定して合計したら?

●行動空間の各次元毎に行動選択したら?

依存関係のある行動を学習できない

高次元行動空間の扱いの難しさ：空間の汎化

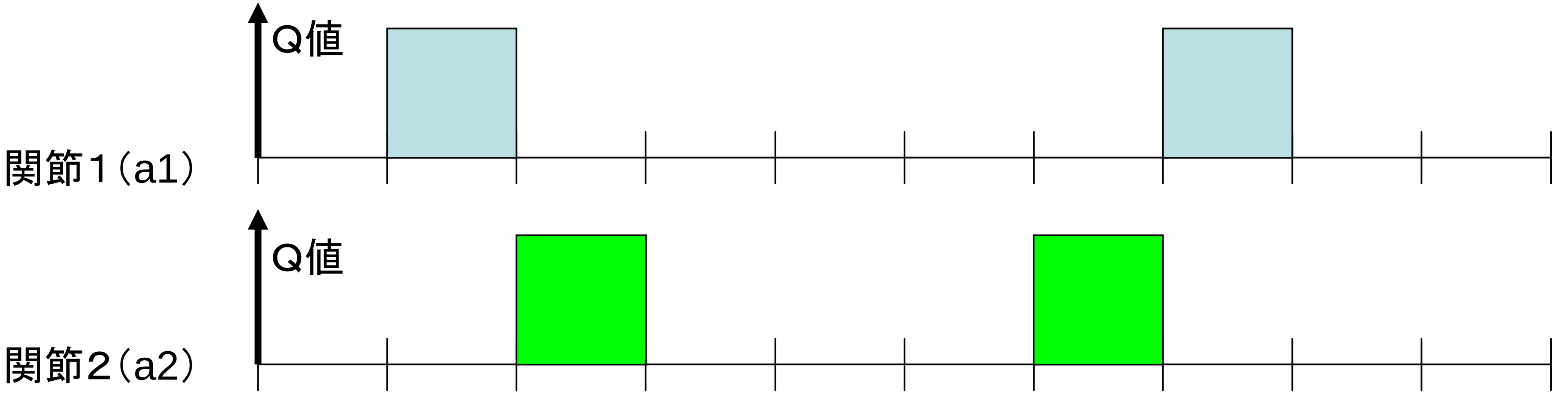


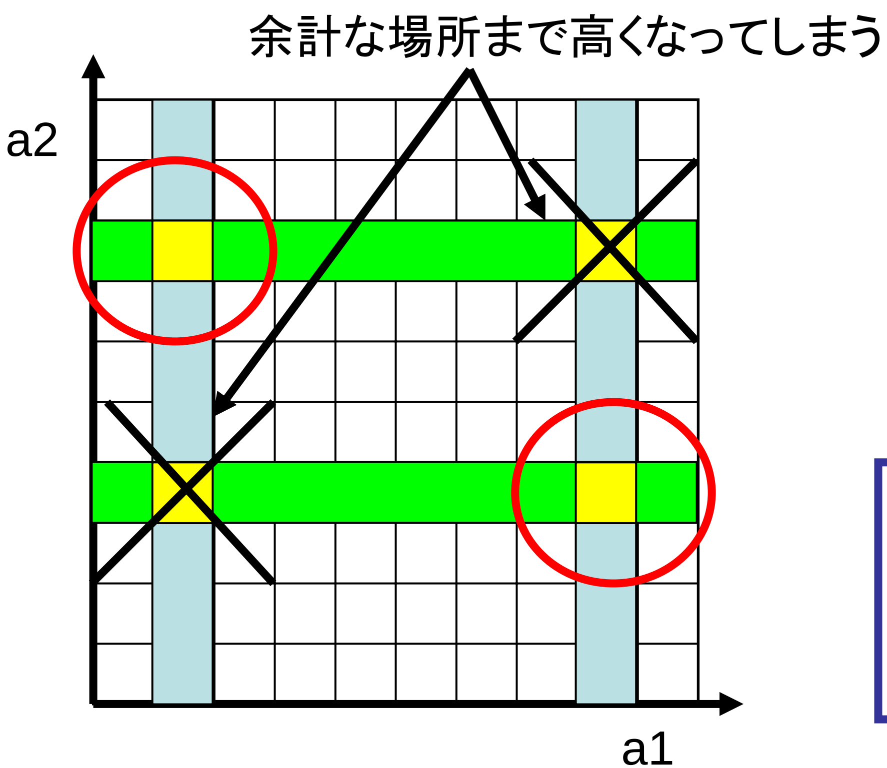


- **行動空間の各次元毎にQ値設定して合計したら？**
- **行動空間の各次元毎に行動選択したら？**

依存関係のある行動を学習できない

高次元行動空間の扱いの難しさ:空間の汎化





- ●行動空間の各次元毎にQ値設定して合計したら?
- ●行動空間の各次元毎に行動選択したら?

依存関係のある行動を学習できない

高次元状態・行動空間の汎化：
ランダムタイリングによるQ関数表現

状態入力
（センサ入力）

固定変換

高次元の
特徴量ベクトル

$$(f_1, f_2, \cdots f_n)$$

線形アーキテクチャ $Q(s,a) = \sum_{i=1}^{n} f_i\, w_i$

各特徴量に対応する重み変数

高次元状態･行動空間の汎化：
ランダムタイリングによるQ関数表現

状態入力
（センサ入力）

固定変換

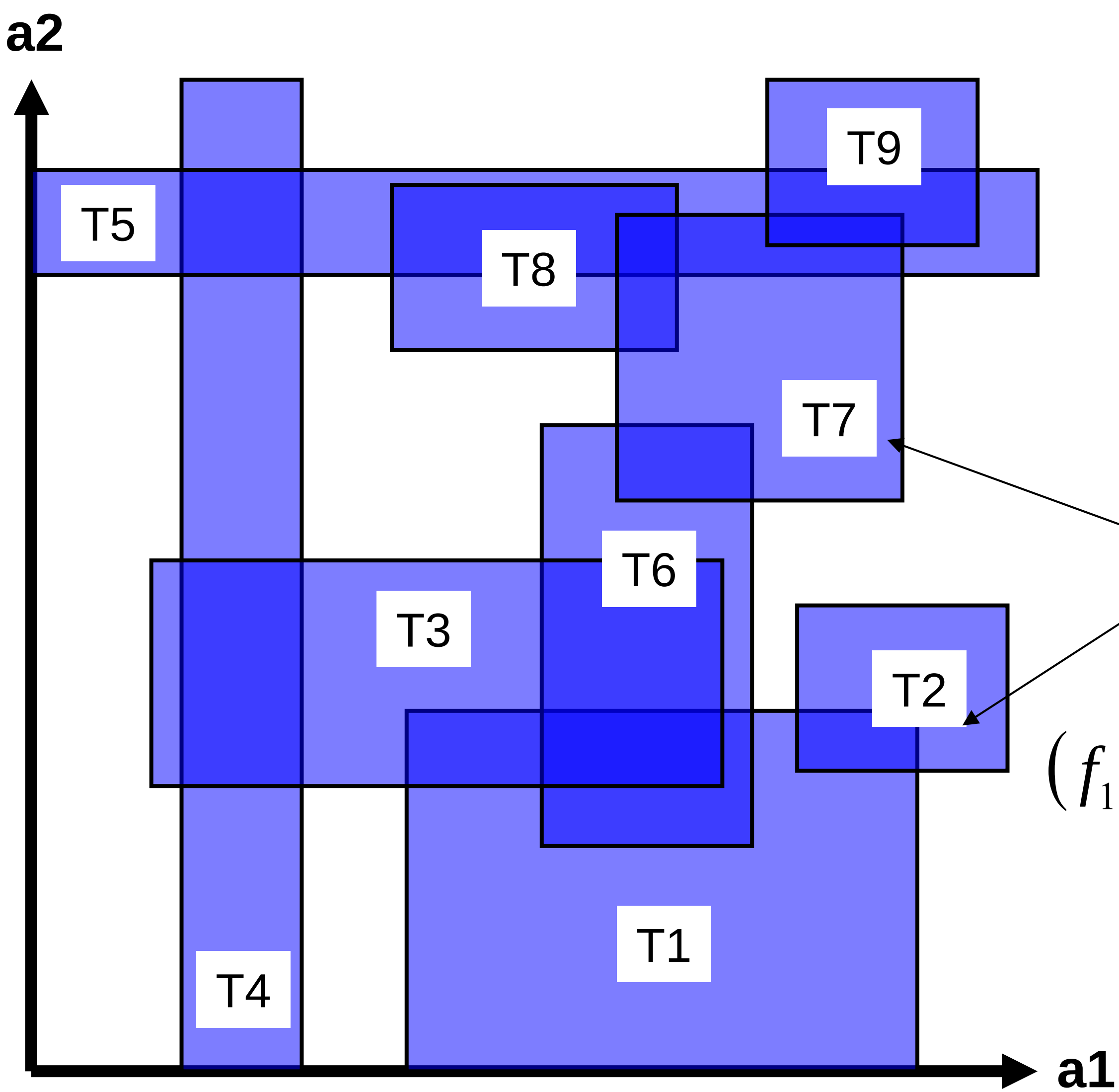
高次元の
特徴量ベクトル
$\left(f_1, f_2, \cdots f_n\right)$

線形アーキテクチャ $Q(s,a) = \sum_{i=1}^{n} f_i \, w_i$

各特徴量に対応する重み変数

多様な非線形変換を用いて
膨大な特徴量ベクトルを生成すると良い
例） サポートベクターマシン

高次元状態･行動空間の汎化：
ランダムタイリングによるQ関数表現



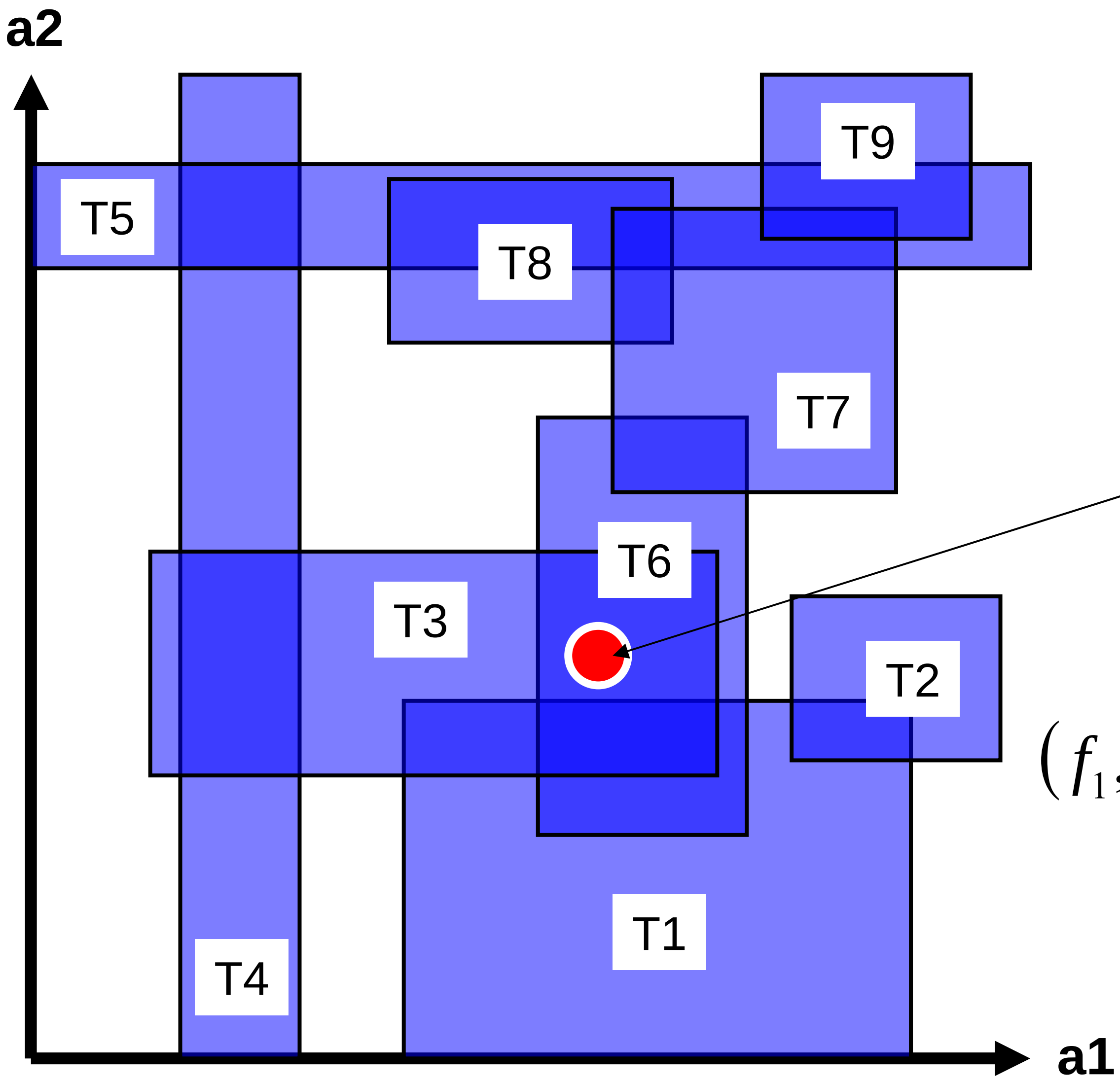
様々な大きさ･次元の超矩形タイルをランダムに配置して特徴量ベクトルを生成

各タイルが特徴量ベクトルの各要素に対応

$$(f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9)$$

実験では、各軸あたり
0.3 の確率で区域を設定

高次元状態・行動空間の汎化：
ランダムタイリングによるQ関数表現

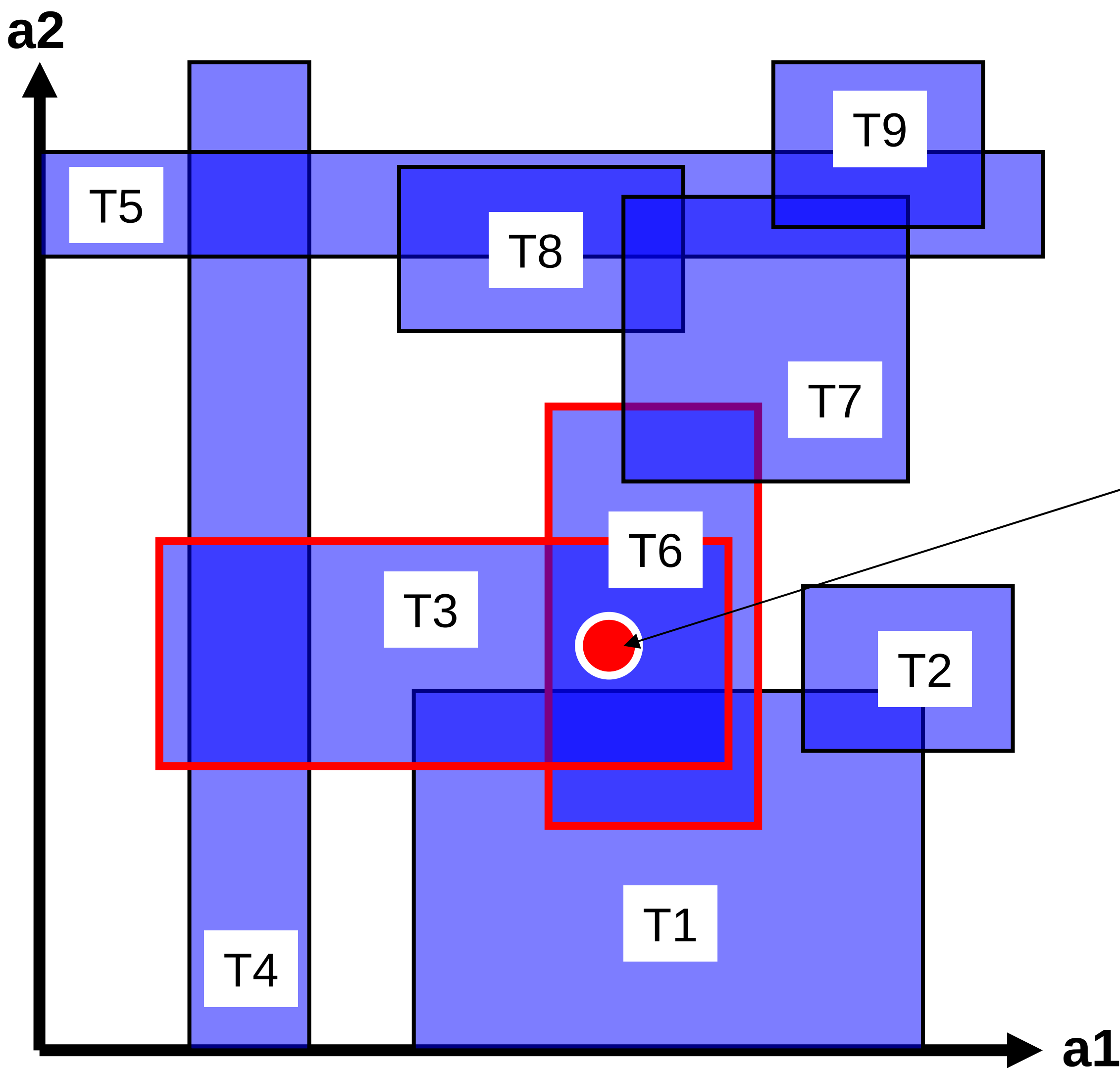


様々な大きさ・次元の超矩形タイルをランダムに配置して特徴量ベクトルを生成

ある値が入力されたとき

$$\left(f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9\right)$$

高次元状態・行動空間の汎化：
ランダムタイリングによるQ関数表現



様々な大きさ・次元の超矩形タイルをランダムに配置して特徴量ベクトルを生成

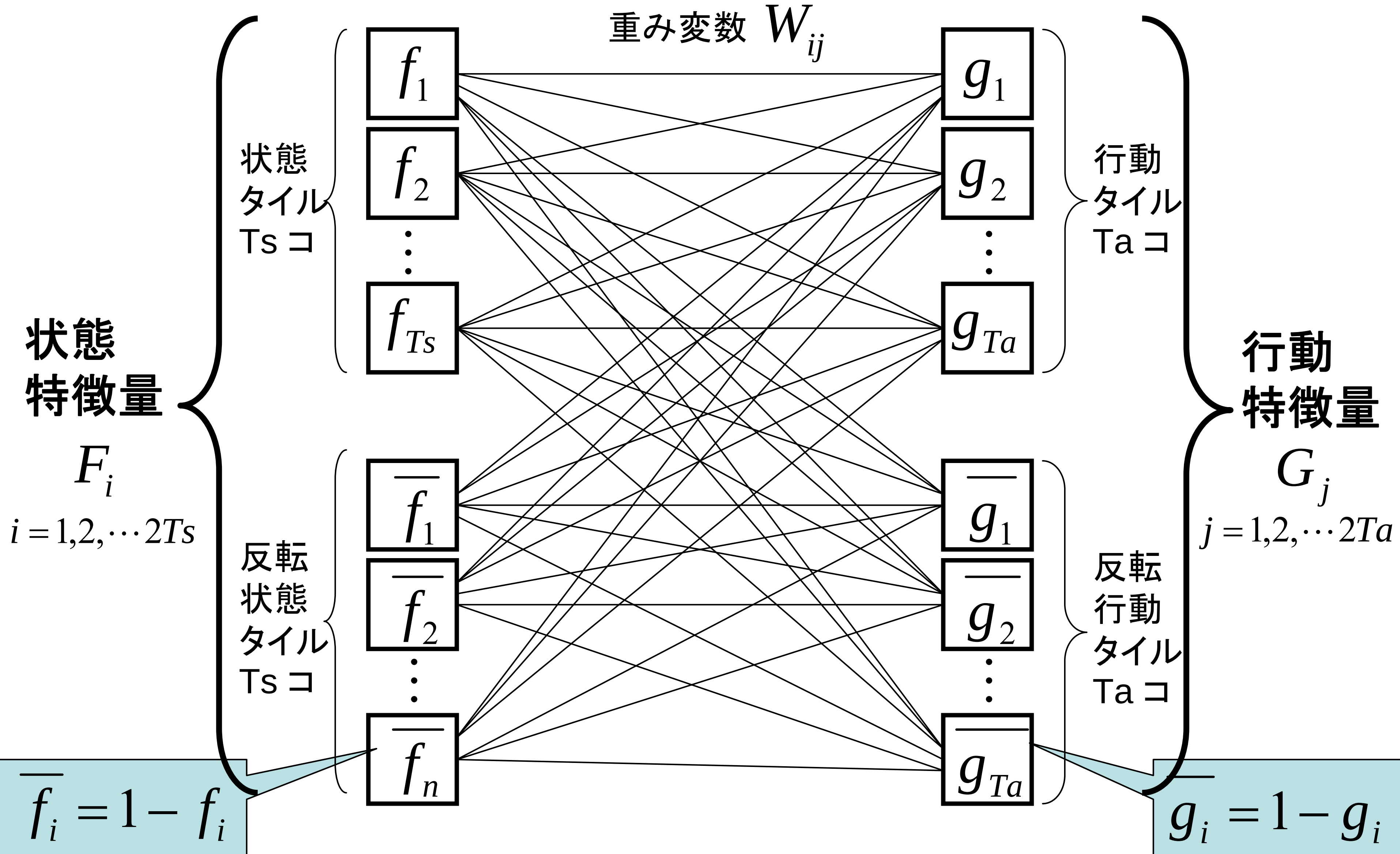
ある値が入力されたとき

$$\left(0, 0, 1, 0, 0, 1, 0, 0, 0\right)$$

該当するタイル
特徴量のみ1

ある状態 s, 行動 a
におけるQ値の求め方

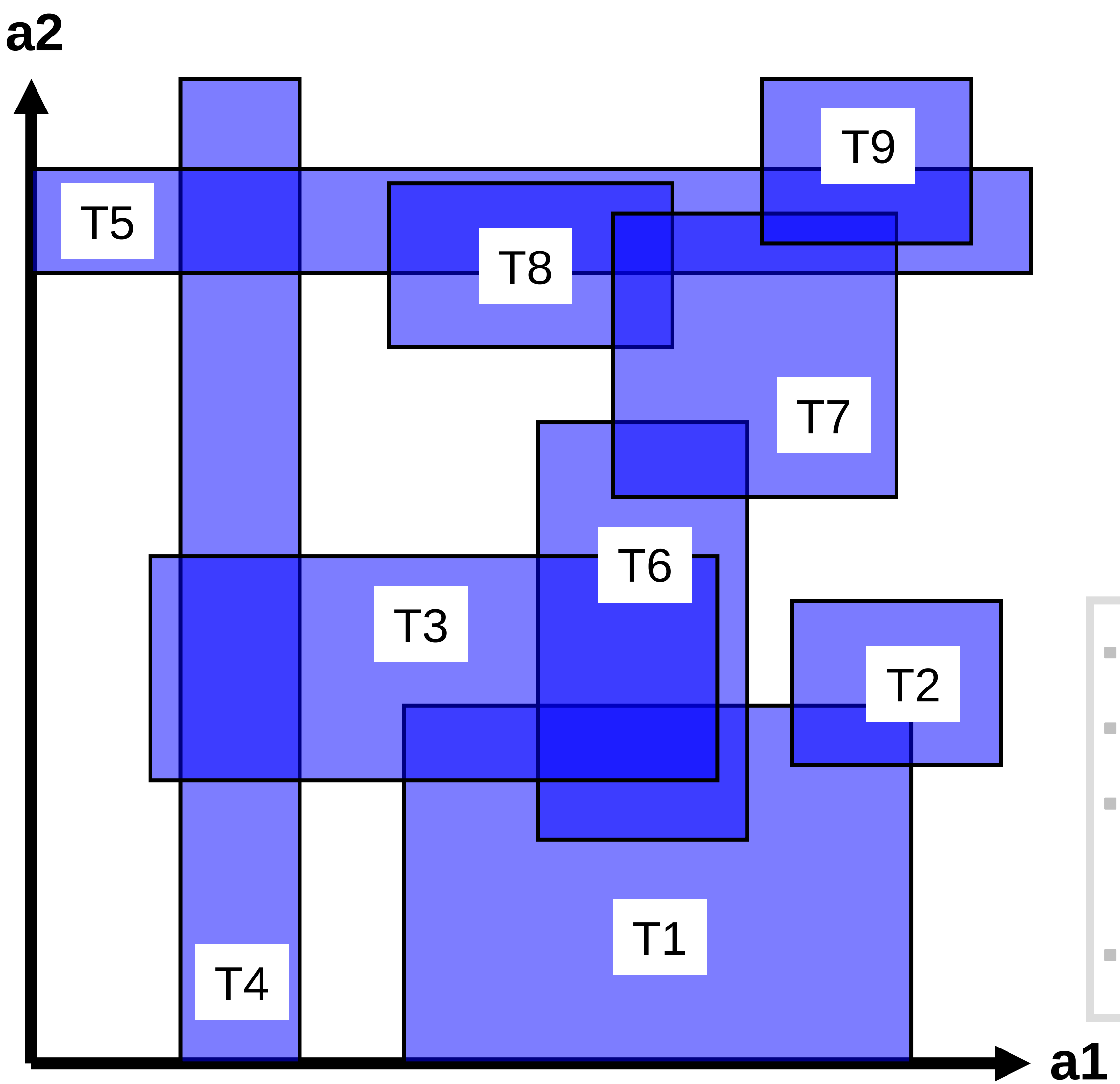
$$Q(s,a)=\frac{1}{Ts\,Ta}\sum_{j=1}^{2Ta}\sum_{i=1}^{2Ts}F_i\ G_j\ W_{ij}$$



Q-learning

連続空間の関数近似

ボルツマン行動選択



ランダムタイリングによる高次元連続空間の汎化

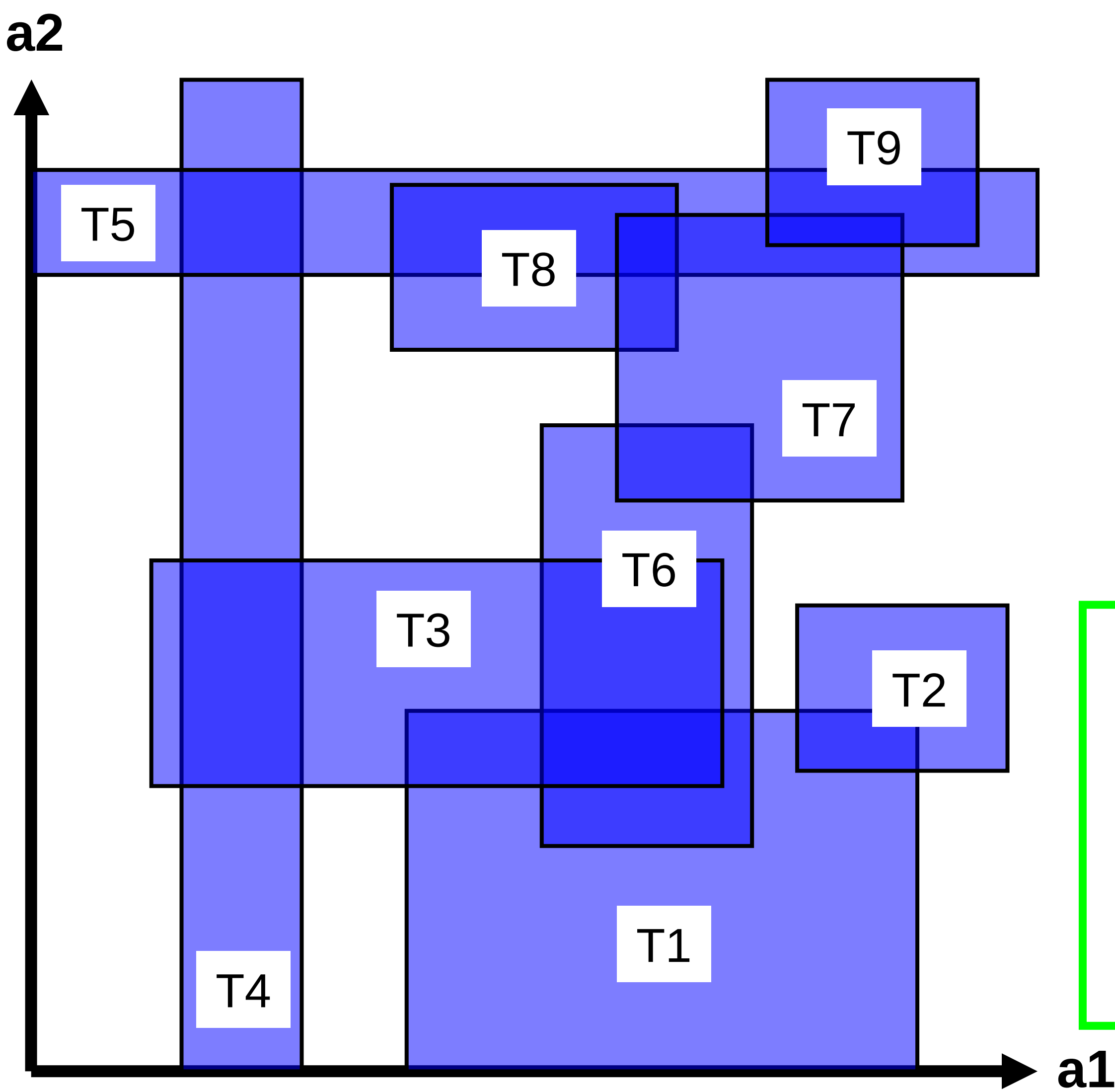
特長

- 実装が簡単
- 空間爆発を回避
- 足りなければタイルを増やすだけ
- 実装が問題に依存しない

Q-learning

連続空間の関数近似

ボルツマン行動選択



ランダムタイリングによる高次元連続空間の汎化

特長

- 実装が簡単
- 空間爆発を回避
- 足りなければタイルを増やすだけ
- 実装が問題に依存しない

Q-learning	連続空間の関数近似	**ボルツマン行動選択**

ある状態SにおいてQ値のボルツマン分布に従って
確率的に行動aを選ぶ

行動選択確率:

$$P(a_i) = \frac{\exp\left(\frac{Q(s,a_i)}{T}\right)}{\sum_{j=1}^{A} \exp\left(\frac{Q(s,a_j)}{T}\right)}$$

しかし！
全行動空間の exp(Q) の合計が必要！

Q-learning

連続空間の関数近似

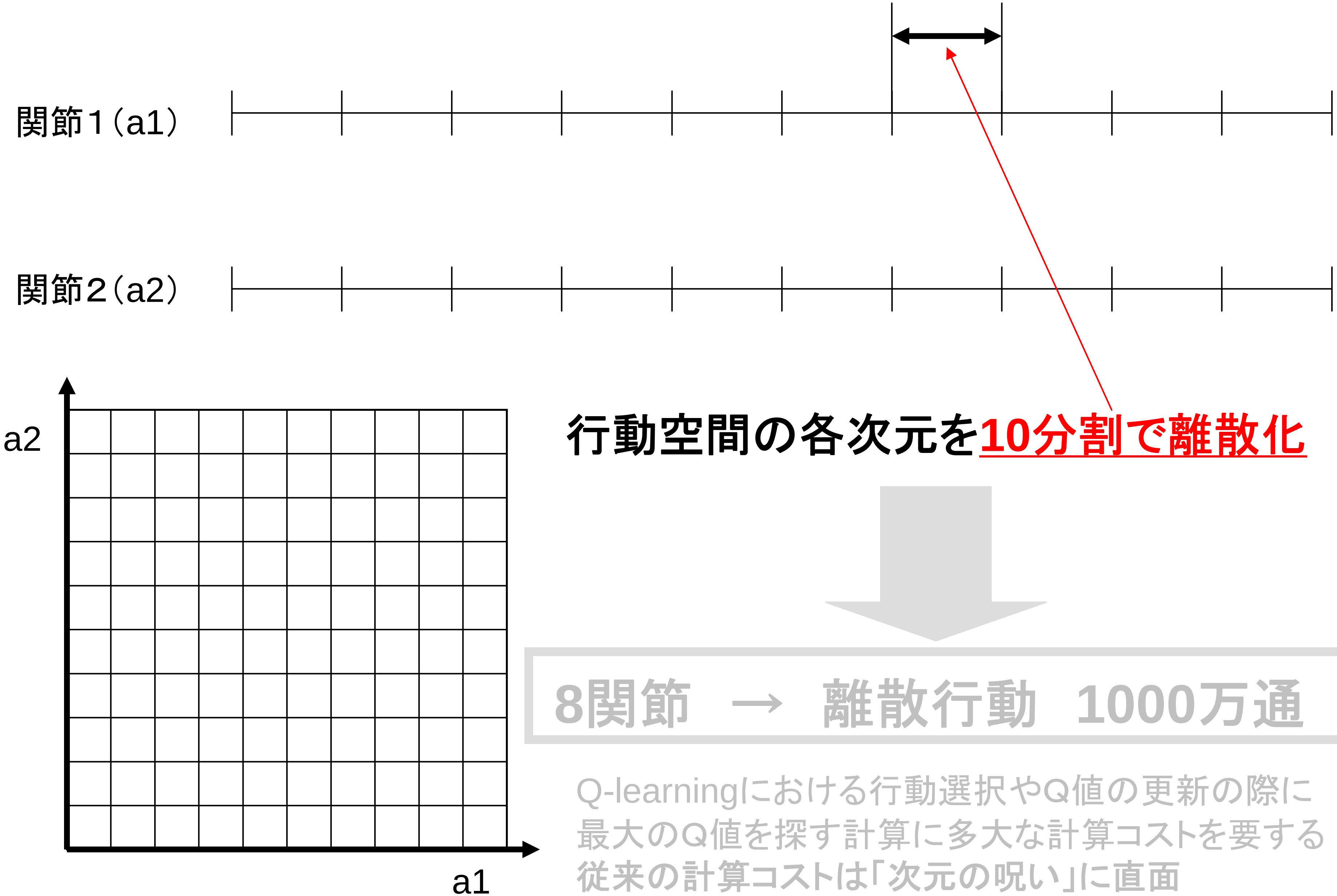
ボルツマン行動選択

ある状態SにおいてQ値のボルツマン分布に従って
確率的に行動aを選ぶ

行動選択確率:
$$P(a_i) = \frac{\exp\left(\frac{Q(s,a_i)}{T}\right)}{\sum_{j=1}^{A} \exp\left(\frac{Q(s,a_j)}{T}\right)}$$

しかし！
全行動空間の exp(Q) の合計が必要！

高次元行動空間の扱いの難しさ: 行動選択

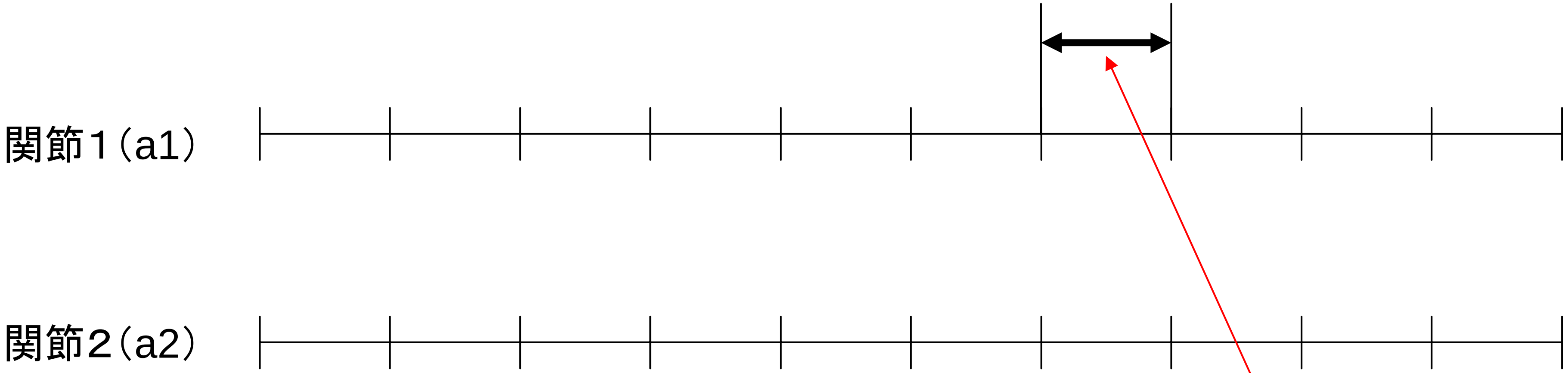


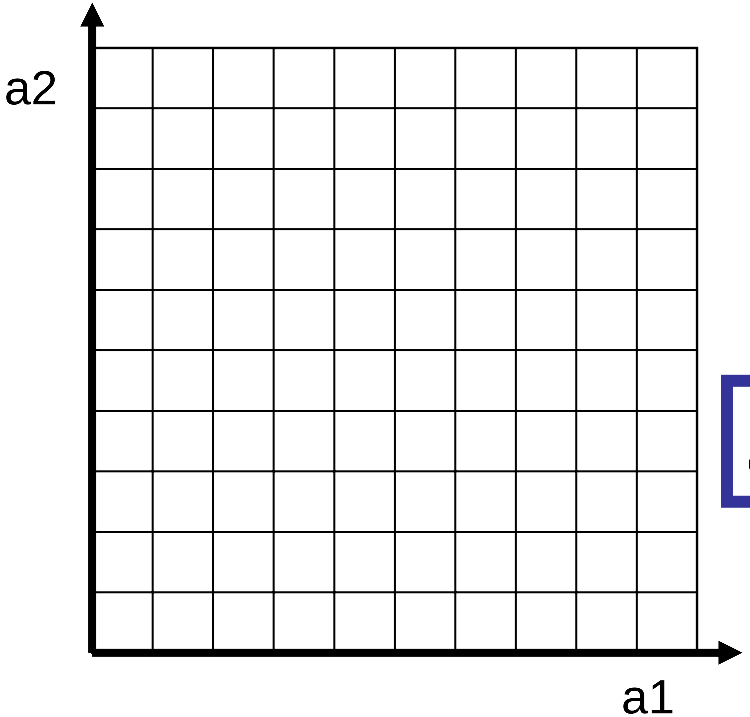
行動空間の各次元を<u>10分割で離散化</u>

8関節 → 離散行動 1000万通

Q-learningにおける行動選択やQ値の更新の際に最大のQ値を探す計算に多大な計算コストを要する
従来の計算コストは「次元の呪い」に直面

高次元行動空間の扱いの難しさ: 行動選択



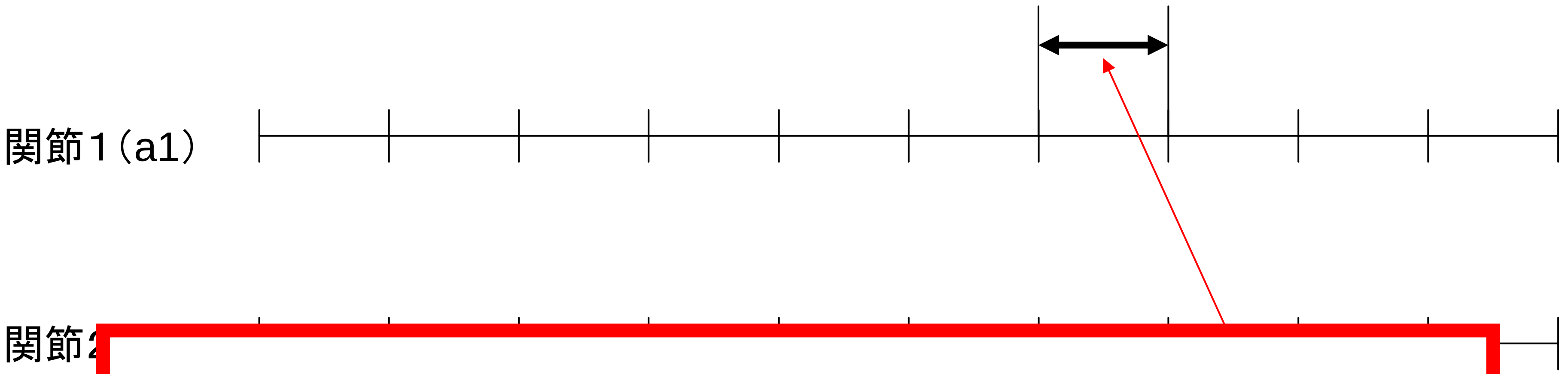


行動空間の各次元を10分割で離散化

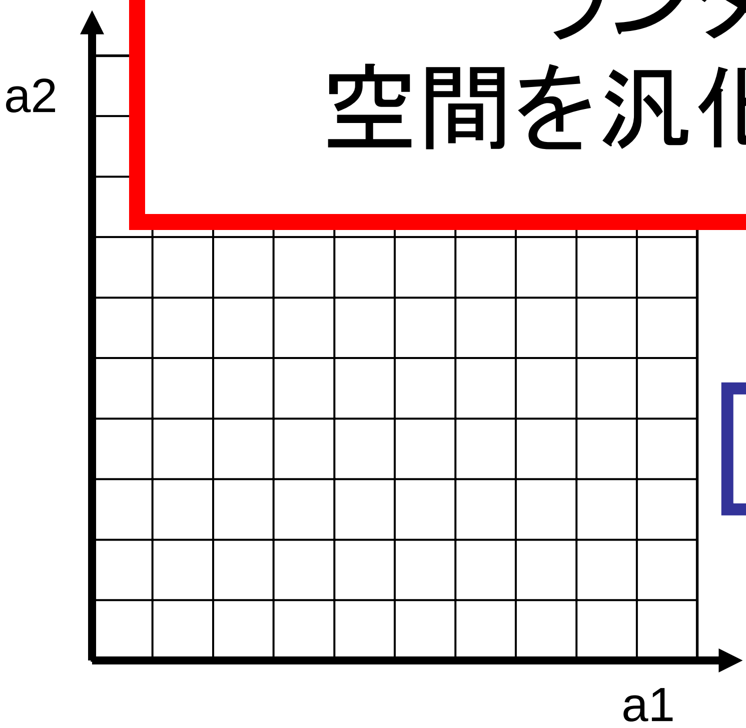
8関節 → 離散行動 1000万通り

Q-learningにおける行動選択やQ値の更新の際に
最大のQ値を探す計算に多大な計算コストを要する
従来の計算コストは「次元の呪い」に直面

高次元行動空間の扱いの難しさ: 行動選択



ランダムタイリングで
空間を汎化しても意味がない!?



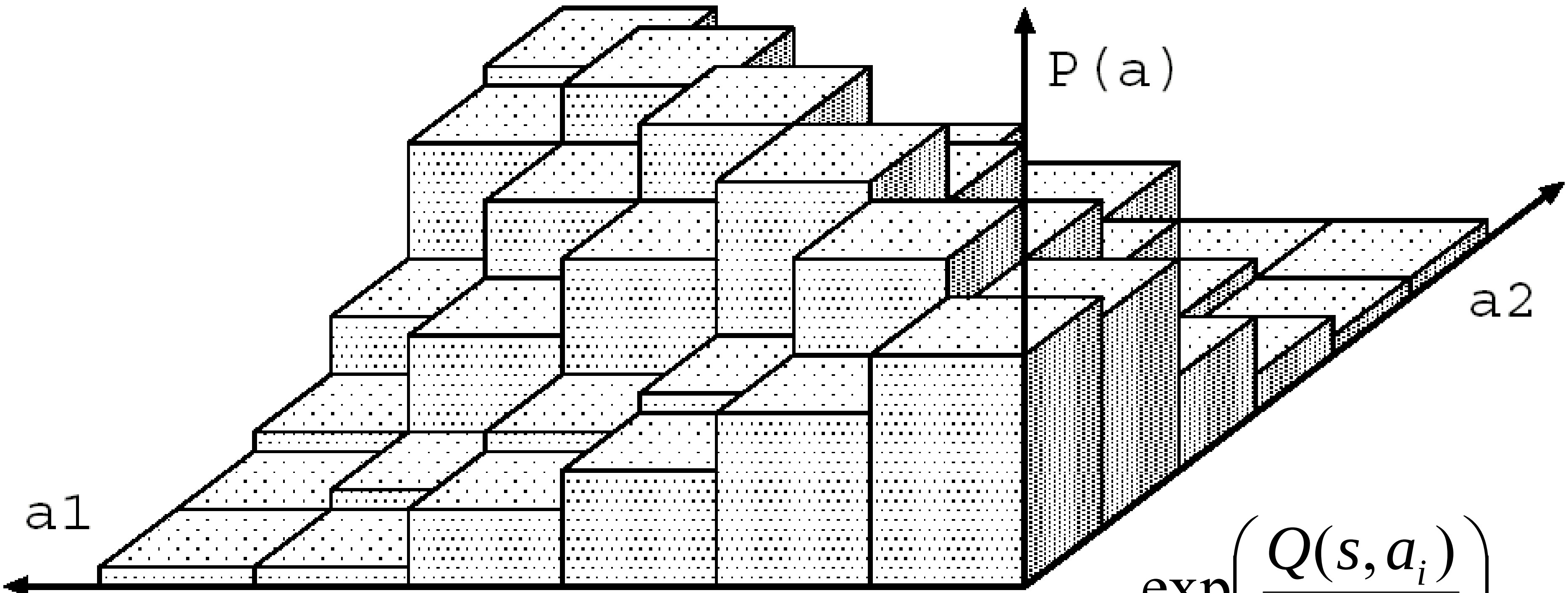
8関節 → 離散行動 1000万通り

Q-learningにおける行動選択やQ値の更新の際に
最大のQ値を探す計算に多大な計算コストを要する
従来の計算コストは「次元の呪い」に直面

Q-learning

連続空間の関数近似

ボルツマン行動選択

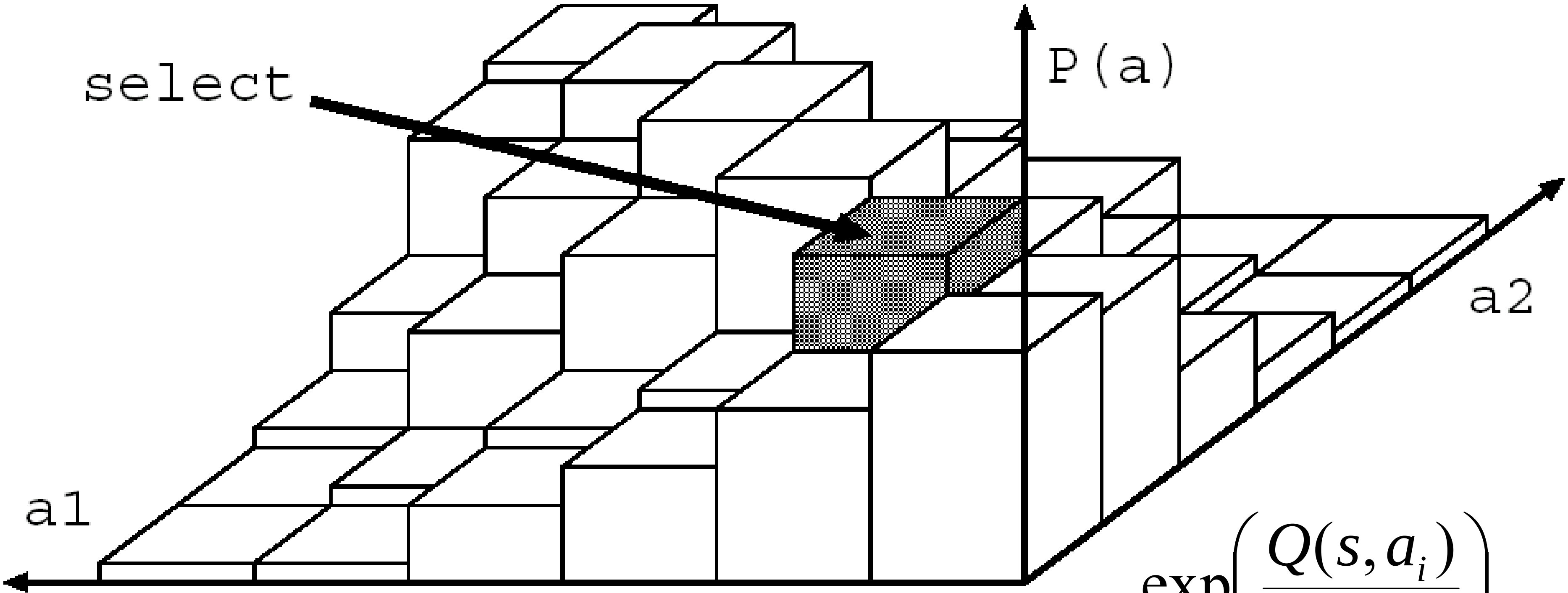


$$P(a_i) = \frac{\exp\left(\dfrac{Q(s,a_i)}{T}\right)}{\sum_{j=1}^{A} \exp\left(\dfrac{Q(s,a_j)}{T}\right)}$$

Q-learning

連続空間の関数近似

ボルツマン行動選択



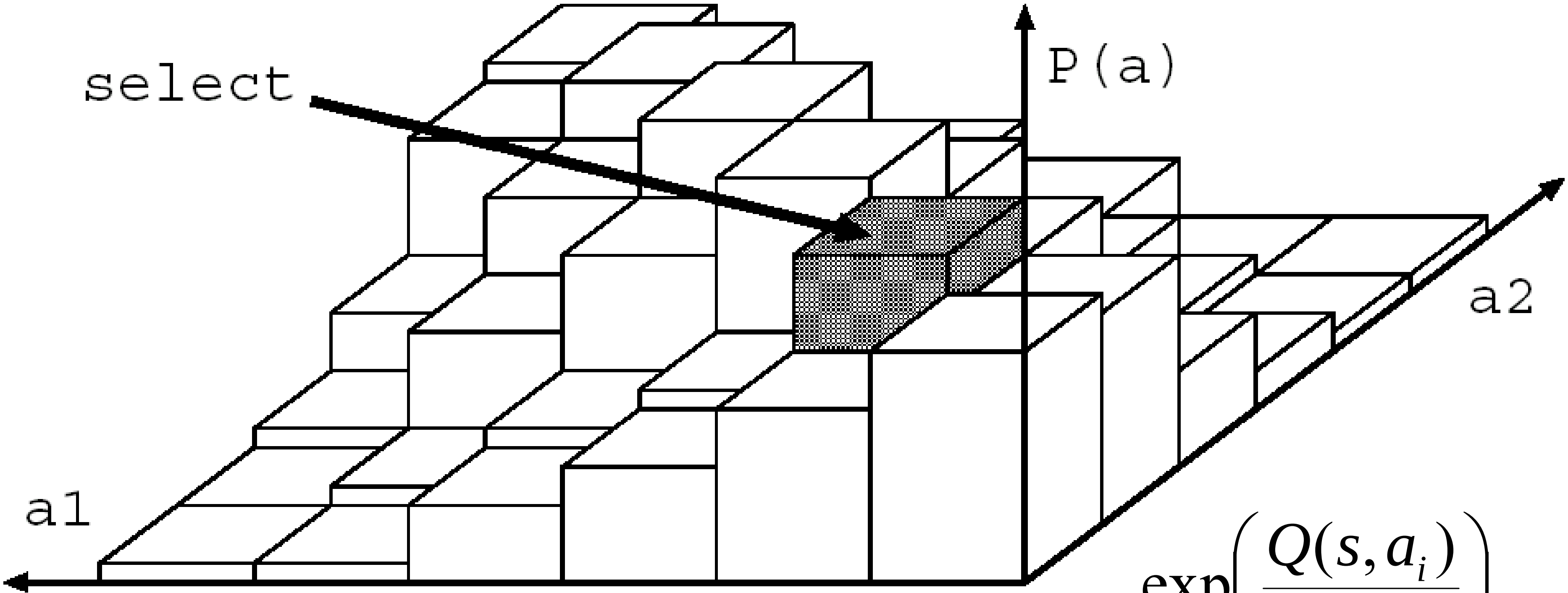
$$P(a_i) = \frac{\exp\left(\dfrac{Q(s,a_i)}{T}\right)}{\sum_{j=1}^{A} \exp\left(\dfrac{Q(s,a_j)}{T}\right)}$$

全行動空間の exp(Q)
の合計が必要！

Q-learning

連続空間の関数近似

ボルツマン行動選択



$$P(a_i) = \frac{\exp\left(\frac{Q(s,a_i)}{T}\right)}{\sum_{j=1}^{A} \exp\left(\frac{Q(s,a_j)}{T}\right)}$$

全行動空間の exp(Q)
の合計が必要！

Q-learning	連続空間の関数近似	ボルツマン行動選択

高次元空間における確率分布に従ったサンプルを効率よく得るには？　→　統計学的な一般問題

Markov chain Monte-Carlo（MCMC）法の一種

Gibbsサンプリング

Q-learning | 連続空間の関数近似 | **ボルツマン行動選択**

高次元空間における確率分布に従ったサンプルを効率よく得るには？ → 統計学的な一般問題

Markov chain Monte-Carlo（MCMC）法の一種

Gibbsサンプリング

計算が簡単な1次元の条件付確率分布

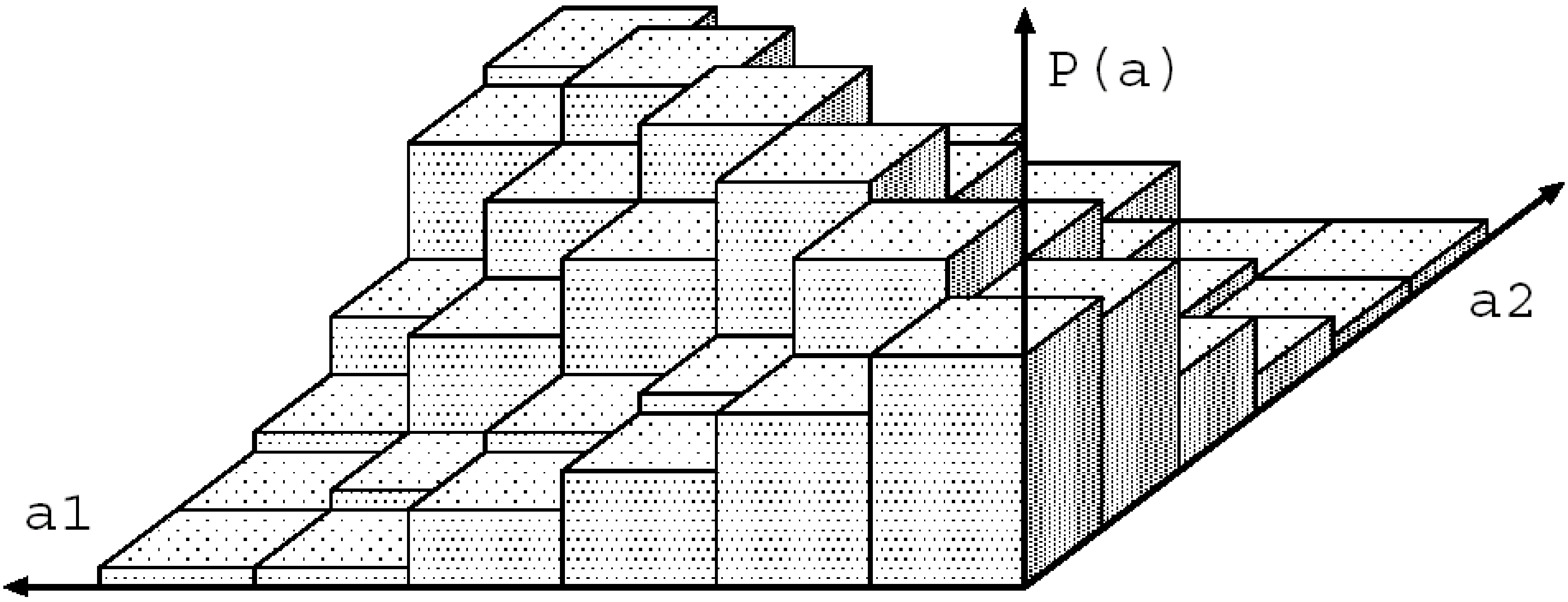
$$
\left.
\begin{aligned}
a^1(t+1) &\approx P(a^1 \mid a^2(t), a^3(t), \cdots a^N(t)) \\
a^2(t+1) &\approx P(a^2 \mid a^1(t), a^3(t), \cdots a^N(t)) \\
&\vdots \\
a^N(t+1) &\approx P(a^N \mid a^1(t), a^2(t), \cdots a^{N-1}(t))
\end{aligned}
\right\}
$$

このような反復 t を十分な回数繰返し、最終的に得たa(t)をサンプルとする

Q-learning

連続空間の関数近似

ボルツマン行動選択

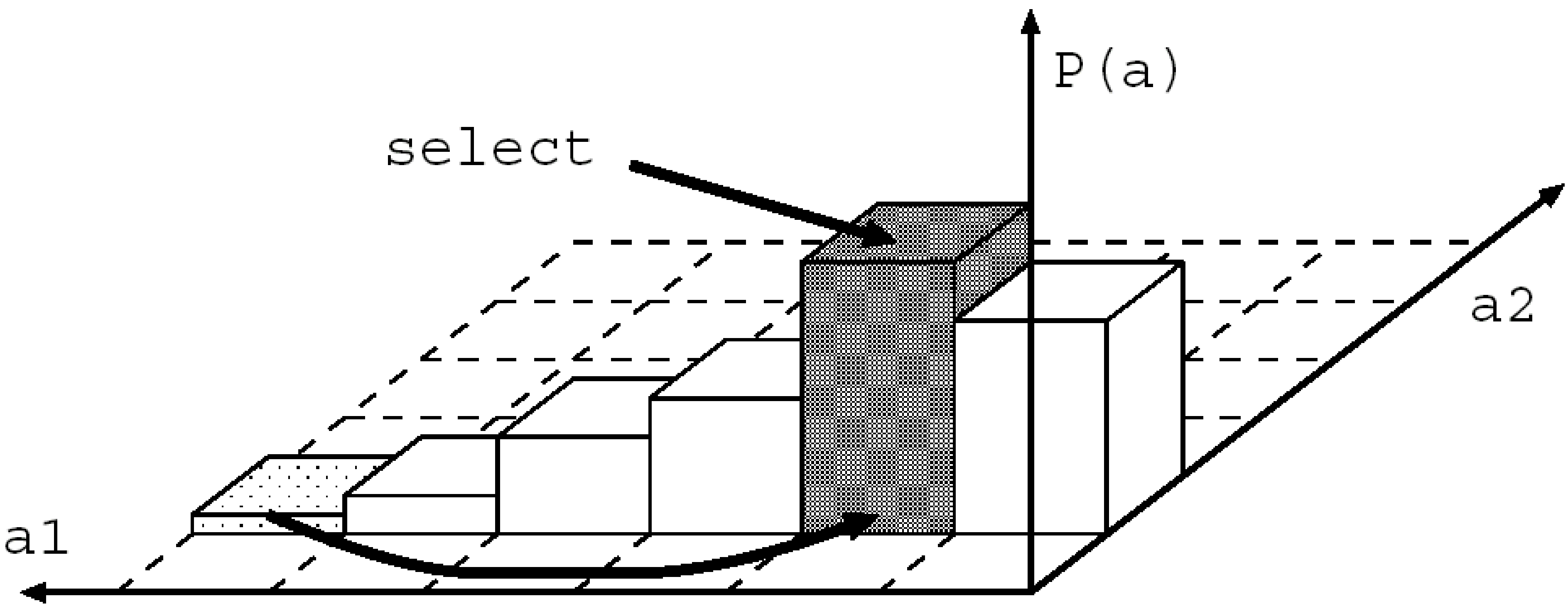


先ほどのボルツマン選択を Gibbsサンプリングしてみる

Q-learning

連続空間の関数近似

ボルツマン行動選択

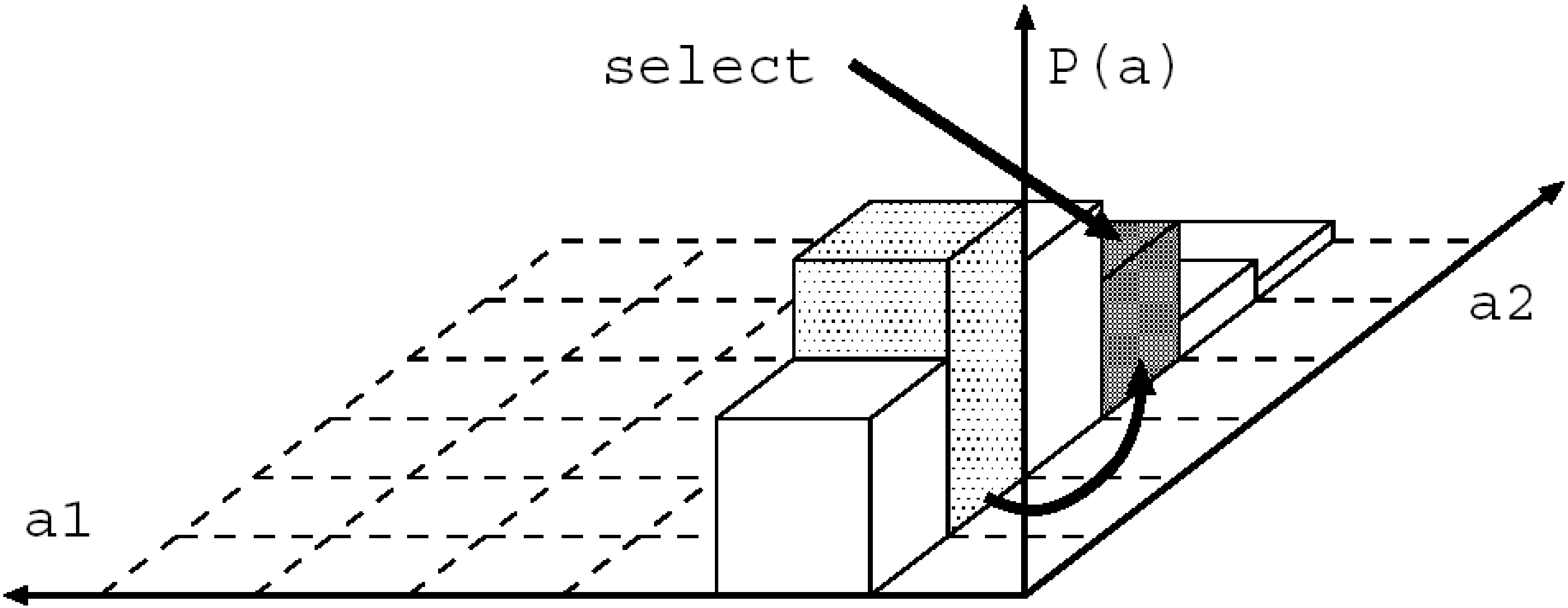


a2軸を固定し、a1軸についてのみボルツマン選択

Q-learning

連続空間の関数近似

ボルツマン行動選択

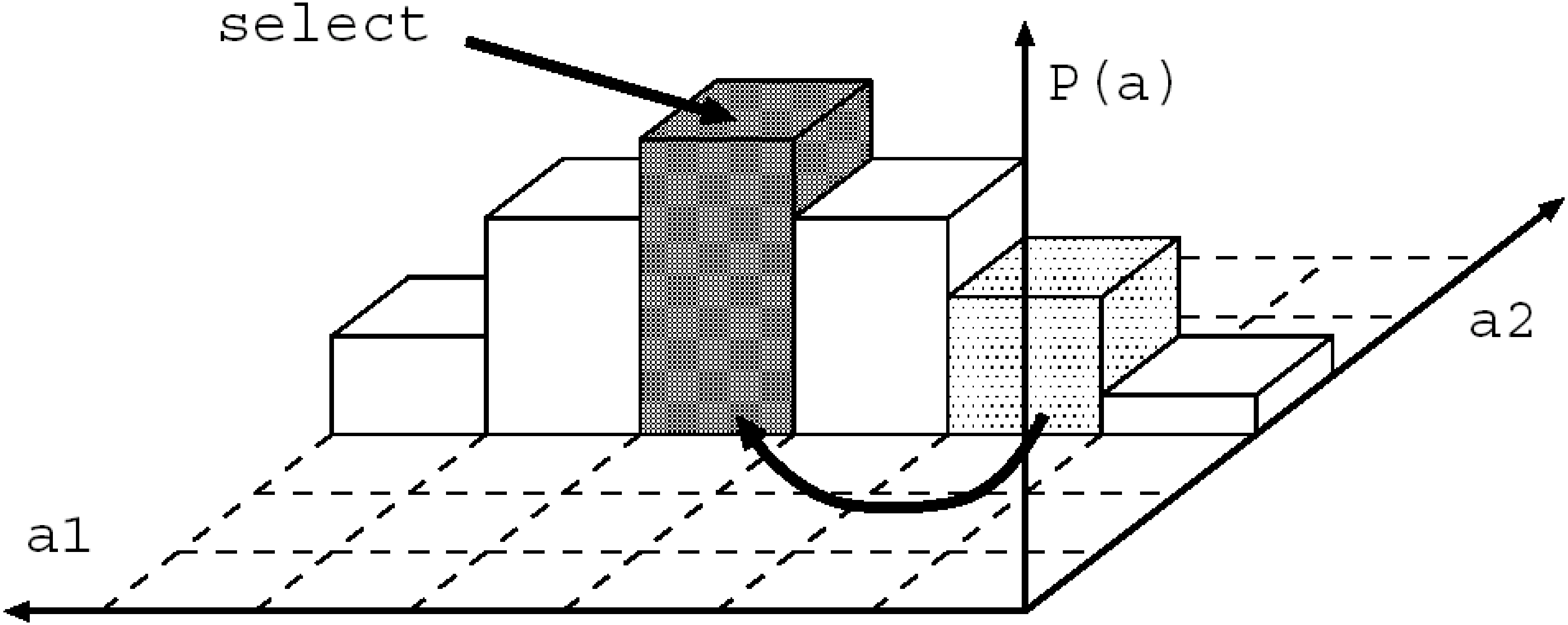


a1軸を先に選択された値に固定し、
今度はa2軸についてのみボルツマン選択

Q-learning

連続空間の関数近似

ボルツマン行動選択

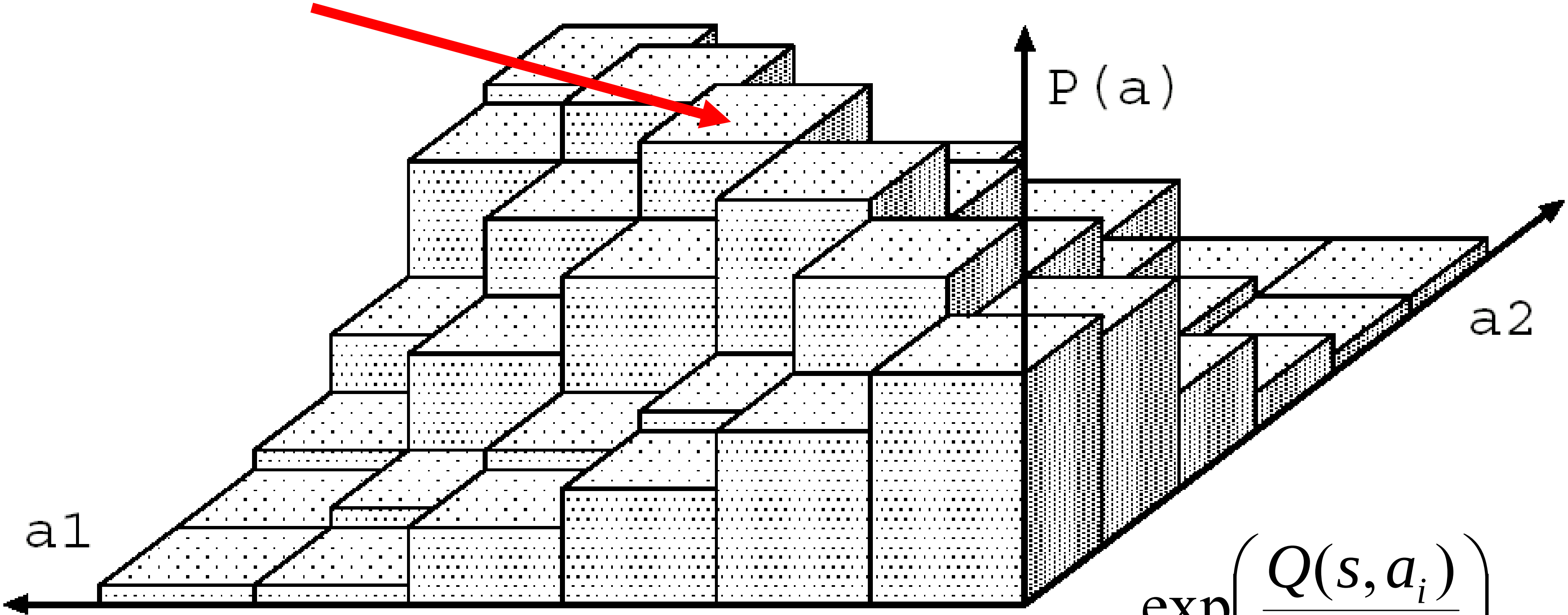


a2軸を先に選択された値に固定し、
再びa1軸についてのみボルツマン選択　これを繰返す

Q-learning

連続空間の関数近似

ボルツマン行動選択



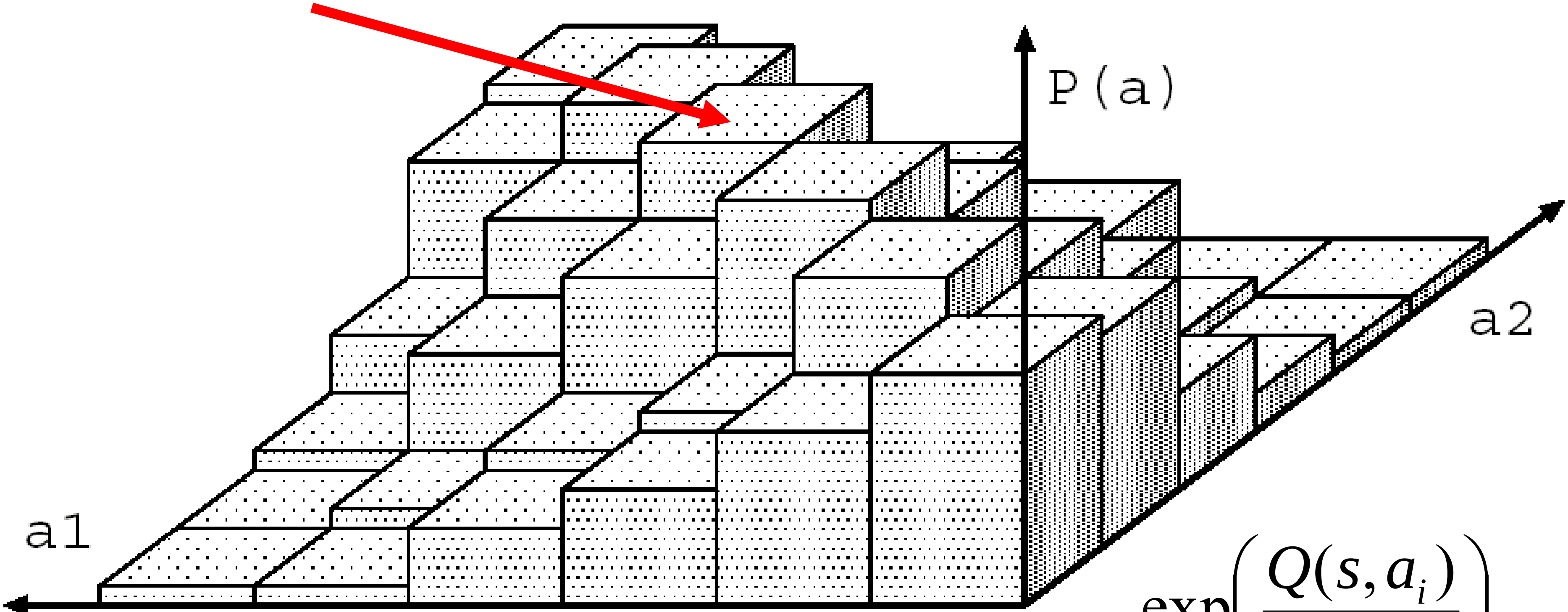
$$P(a_i) = \frac{\exp\left(\frac{Q(s,a_i)}{T}\right)}{\sum_{j=1}^{A} \exp\left(\frac{Q(s,a_j)}{T}\right)}$$

Gibbsサンプリング：
全行動空間の exp(Q) の合計を
計算せずに右式に従うサンプルを得る！

Q-learning

連続空間の関数近似

ボルツマン行動選択



$$P(a_i) = \frac{\exp\left(\frac{Q(s,a_i)}{T}\right)}{\sum_{j=1}^{A} \exp\left(\frac{Q(s,a_j)}{T}\right)}$$

Gibbsサンプリング:
全行動空間の exp(Q) の合計を計算せずに右式に従うサンプルを得る！

Q-learning

連続空間の関数近似

ボルツマン行動選択

Gibbsサンプリングによる行動選択処理

ただし、
MaxQ を求める
計算が「近似」に

SARSAアルゴリズムなら問題ない

特長

- 実装が簡単
- 空間爆発を回避
- 足りなければ計算反復の回数を増やすだけ
- 分割を細かくしても、計算はあまり変わらない

Q-learning

連続空間の関数近似

ボルツマン行動選択

Gibbsサンプリングによる行動選択処理

ただし、
MaxQ を求める
計算が「近似」に

SARSAアルゴリズムなら問題ない

特長

- 実装が簡単
- 空間爆発を回避
- 足りなければ計算反復の回数を増やすだけ
- 分割を細かくしても、計算はあまり変わらない

Q-learning

連続空間の関数近似

ボルツマン行動選択

Gibbsサンプリングによる
行動選択処理

ただし、
MaxQ を求める
計算が「近似」に

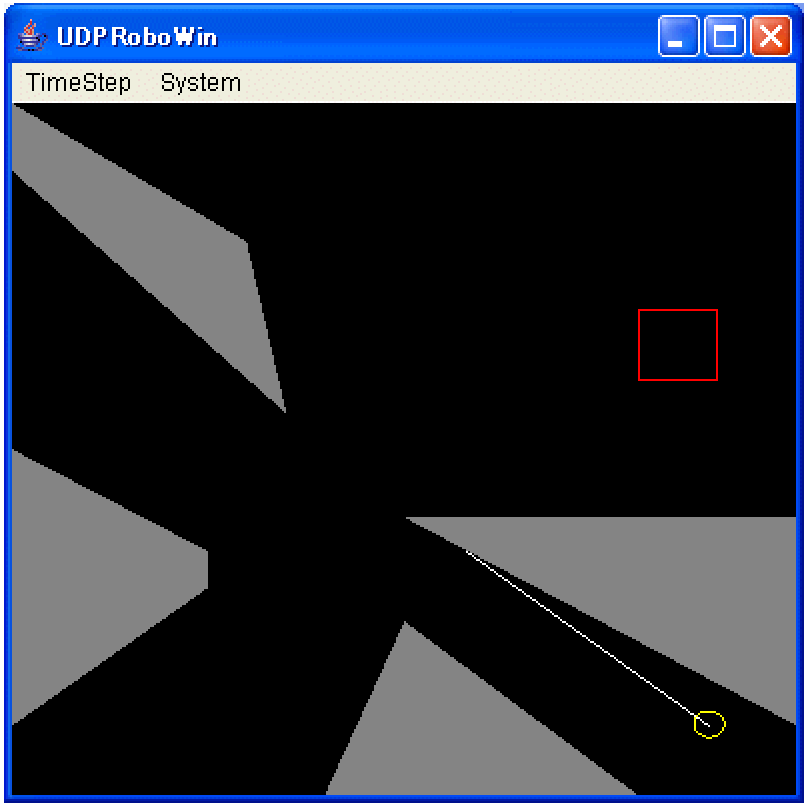
SARSAアルゴリズムなら問題ない

特長

- 実装が簡単
- 空間爆発を回避
- 足りなければ計算反復の回数を増やすだけ
- 分割を細かくしても、計算はあまり変わらない

シミュレーション１： Rod in Maze

状態3次元
行動3次元



特徴ベクトルの設定

１）ランダムタイリング
3次元空間を10x10x10 の格子状とし、
タイルの境界をこの格子に合わせた
ランダムな大きさのタイルを200個生成
各次元要素をタイルに選択する確率=0.6

２）等間隔グリッドタイリング
３次元空間を 5x5x5 の等間隔グリッドで
分割し、5x5x5=125個のタイル生成

３次元空間を 6x6x6 の等間隔グリッドで
分割し、6x6x6=216個のタイル生成

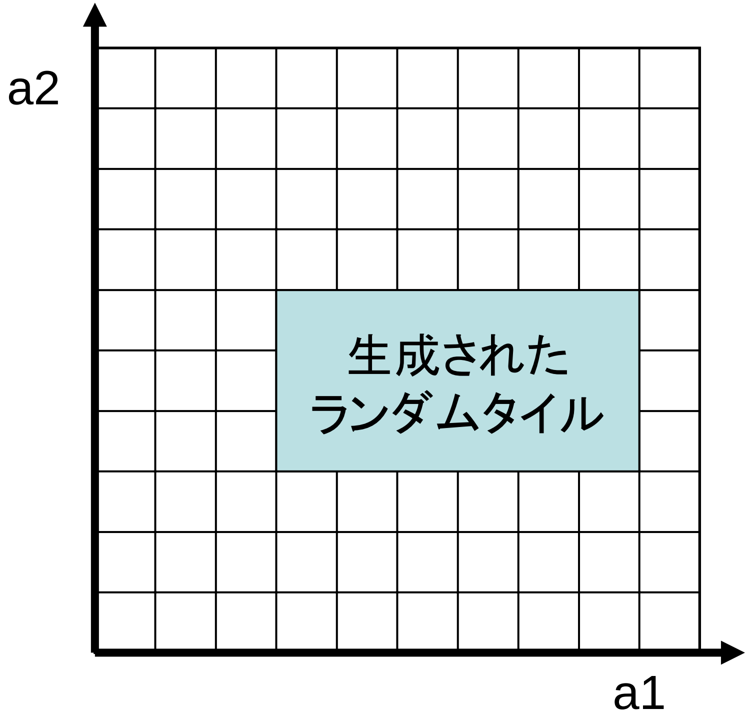
Q-learning の設定

割引率 γ=0.9
温　度　T=0.4
学習率 α=0.5

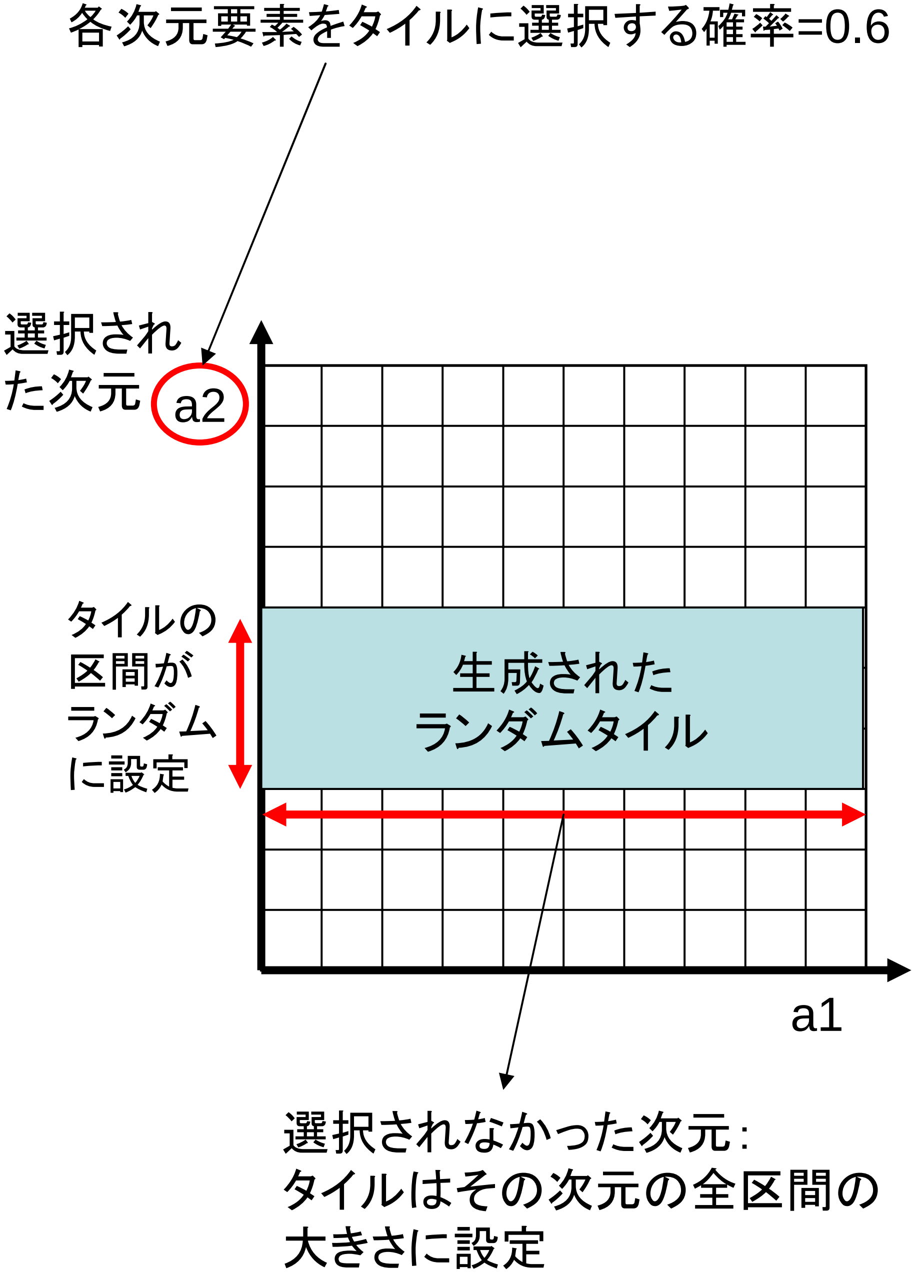
Gibbs-Sampling の設定

反復回数：15回×3次元

ランダムタイルの生成



3次元空間を10x10x10 の格子状とし、
タイルの境界をこの格子に合わせた
ランダムな大きさのタイルを生成

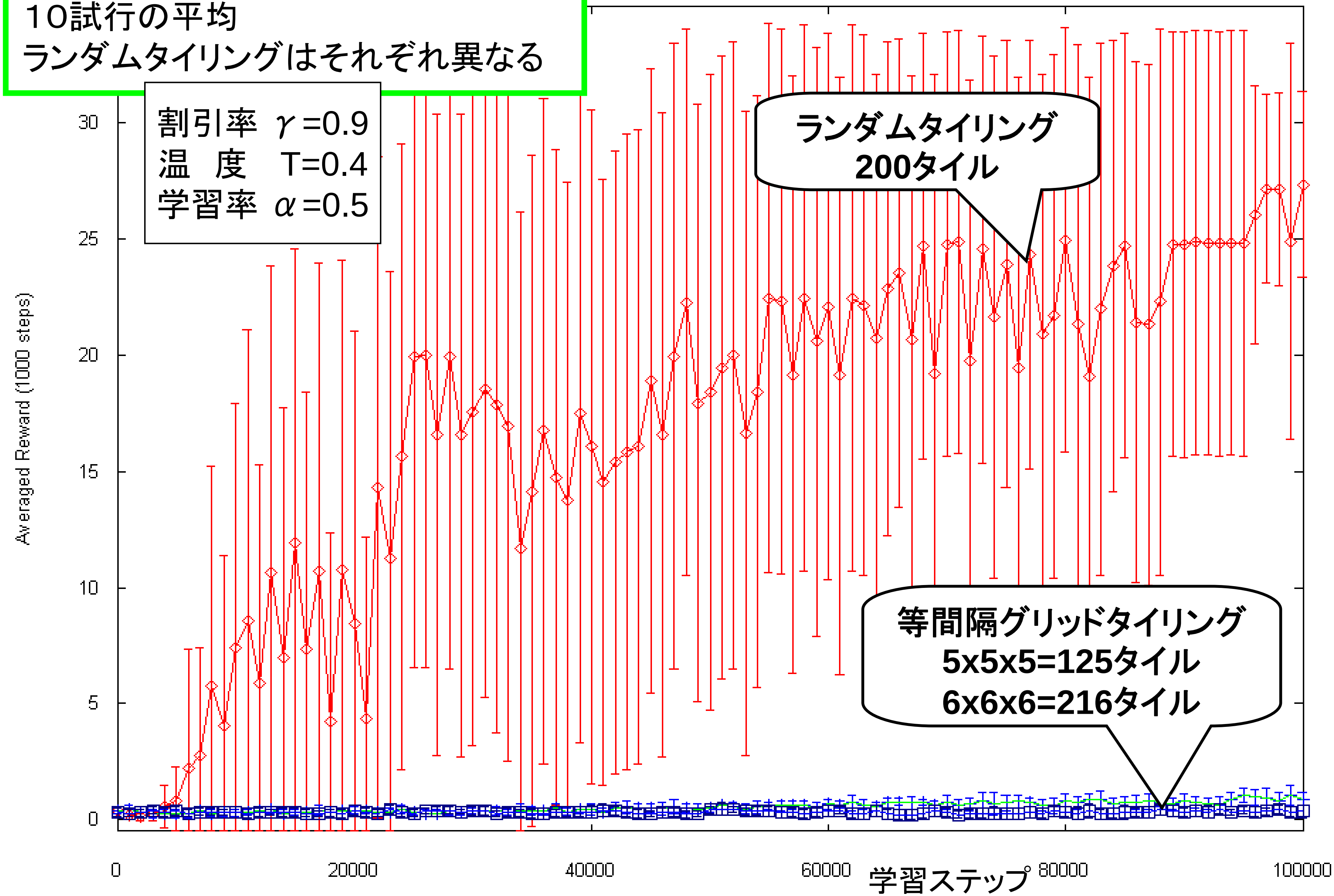


シミュレーション１結果： Rod in Maze

状態3次元
行動3次元

１０試行の平均
ランダムタイリングはそれぞれ異なる

割引率 γ=0.9
温　度　T=0.4
学習率 α=0.5

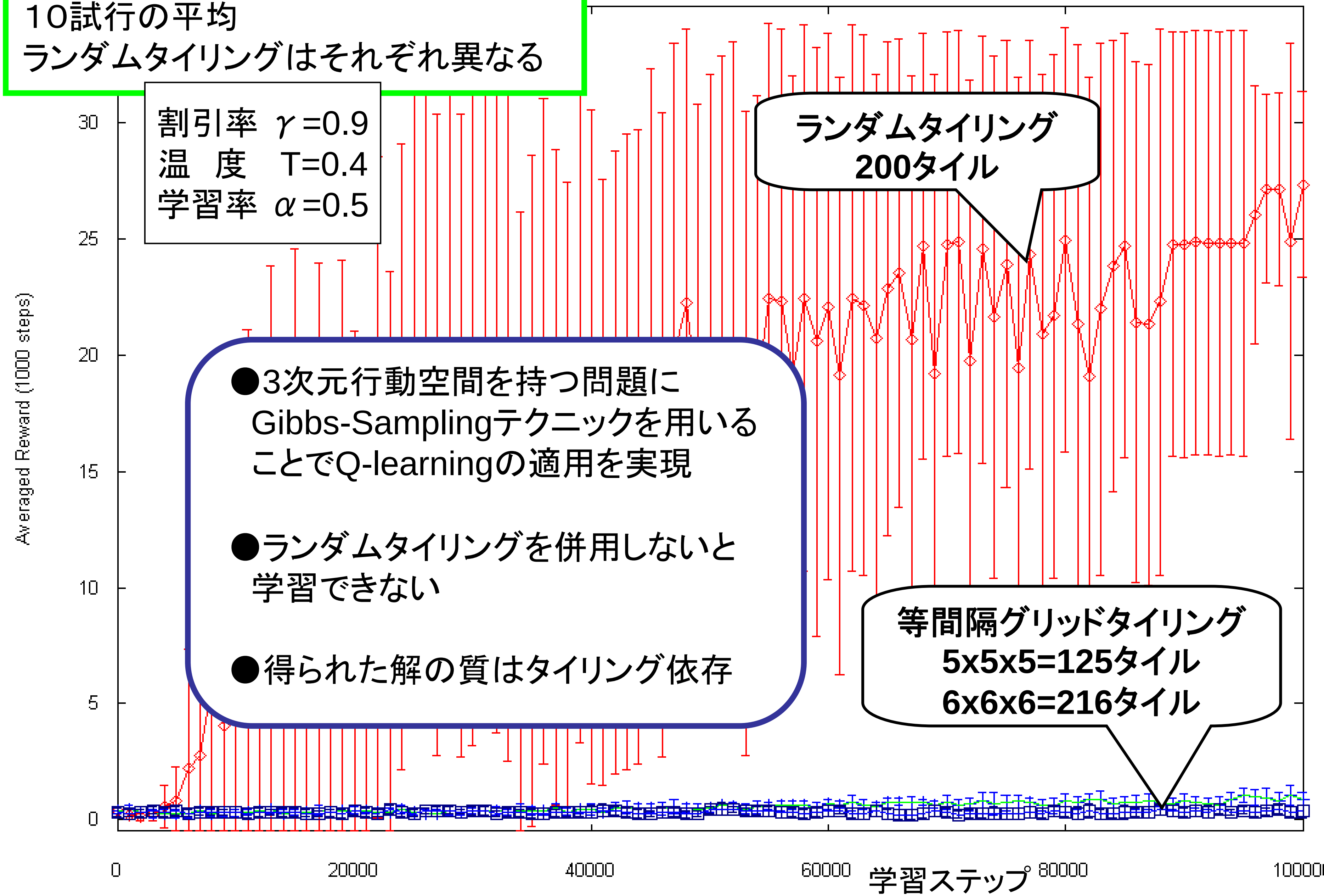


シミュレーション1結果： Rod in Maze

状態3次元
行動3次元

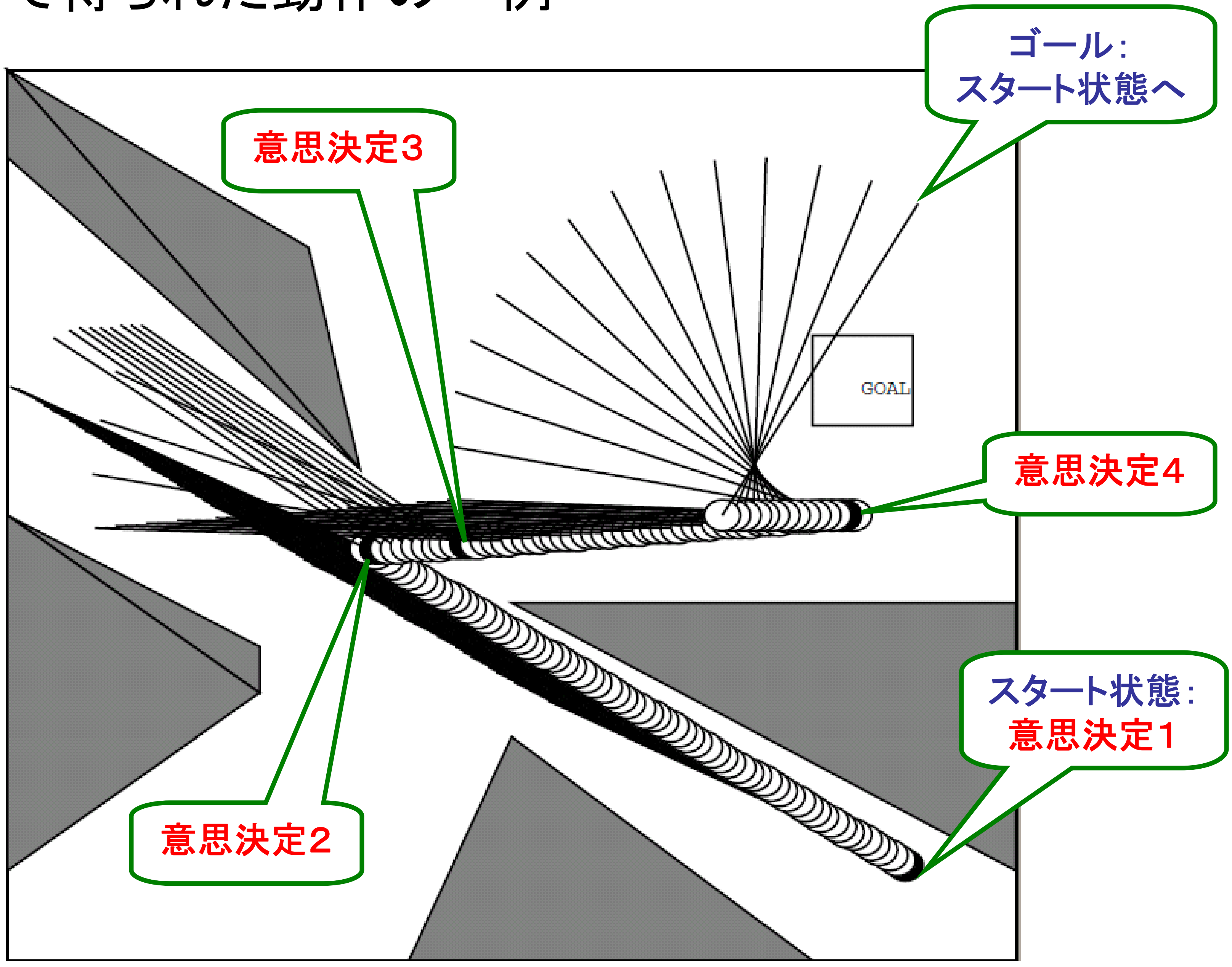
10試行の平均
ランダムタイリングはそれぞれ異なる

割引率 γ=0.9
温　度　T=0.4
学習率 α=0.5



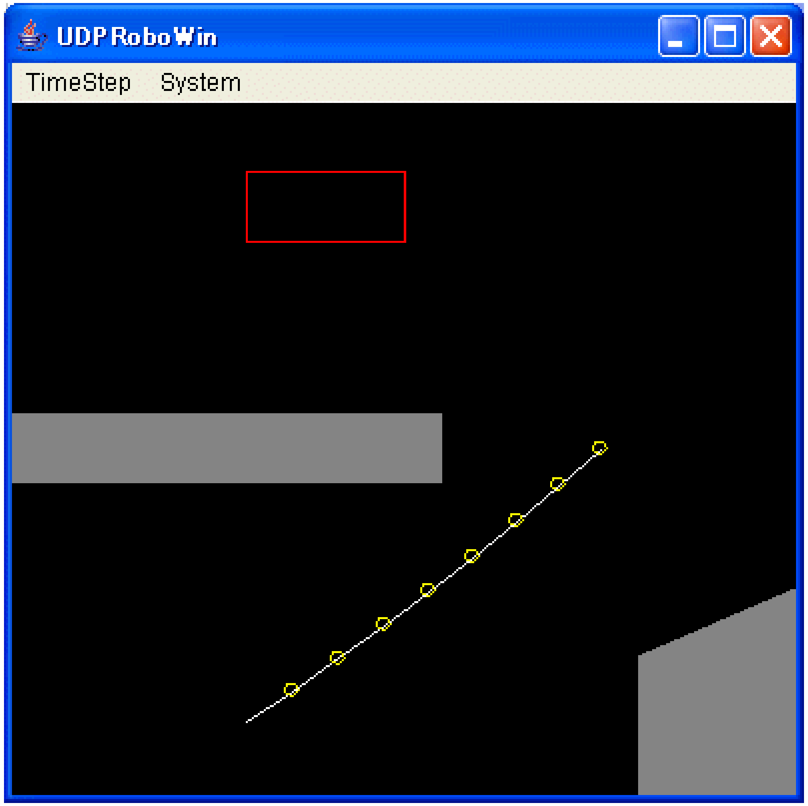
- 3次元行動空間を持つ問題にGibbs-Samplingテクニックを用いることでQ-learningの適用を実現
- ランダムタイリングを併用しないと学習できない
- 得られた解の質はタイリング依存

学習で得られた動作の一例



シミュレーション2： Multi-Joint Arm

状態8次元
行動8次元



特徴ベクトルの設定

1）ランダムタイリング
8次元空間を10^8 の格子状とし、
タイルの境界をこの格子に合わせた
ランダムな大きさのタイルを200個生成
各次元要素をタイルに選択する確率=0.3

2）等間隔グリッドタイリング
8次元空間を 2^8 の等間隔グリッドで
分割し、2^8=256個のタイル生成

Q-learning の設定

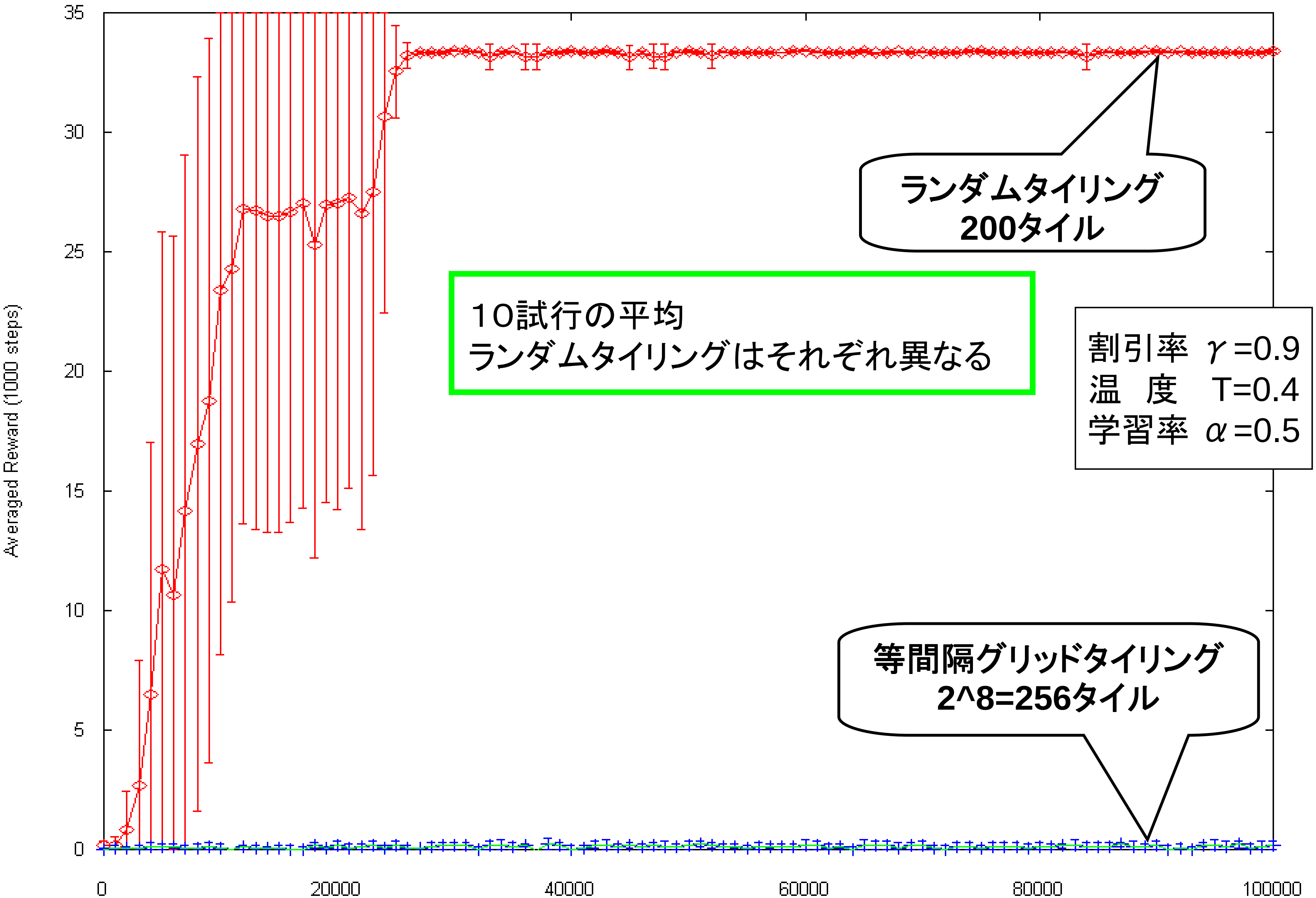
割引率 γ=0.9
温　度　T=0.4
学習率 α=0.5

Gibbs-Sampling の設定

反復回数：40回×8次元

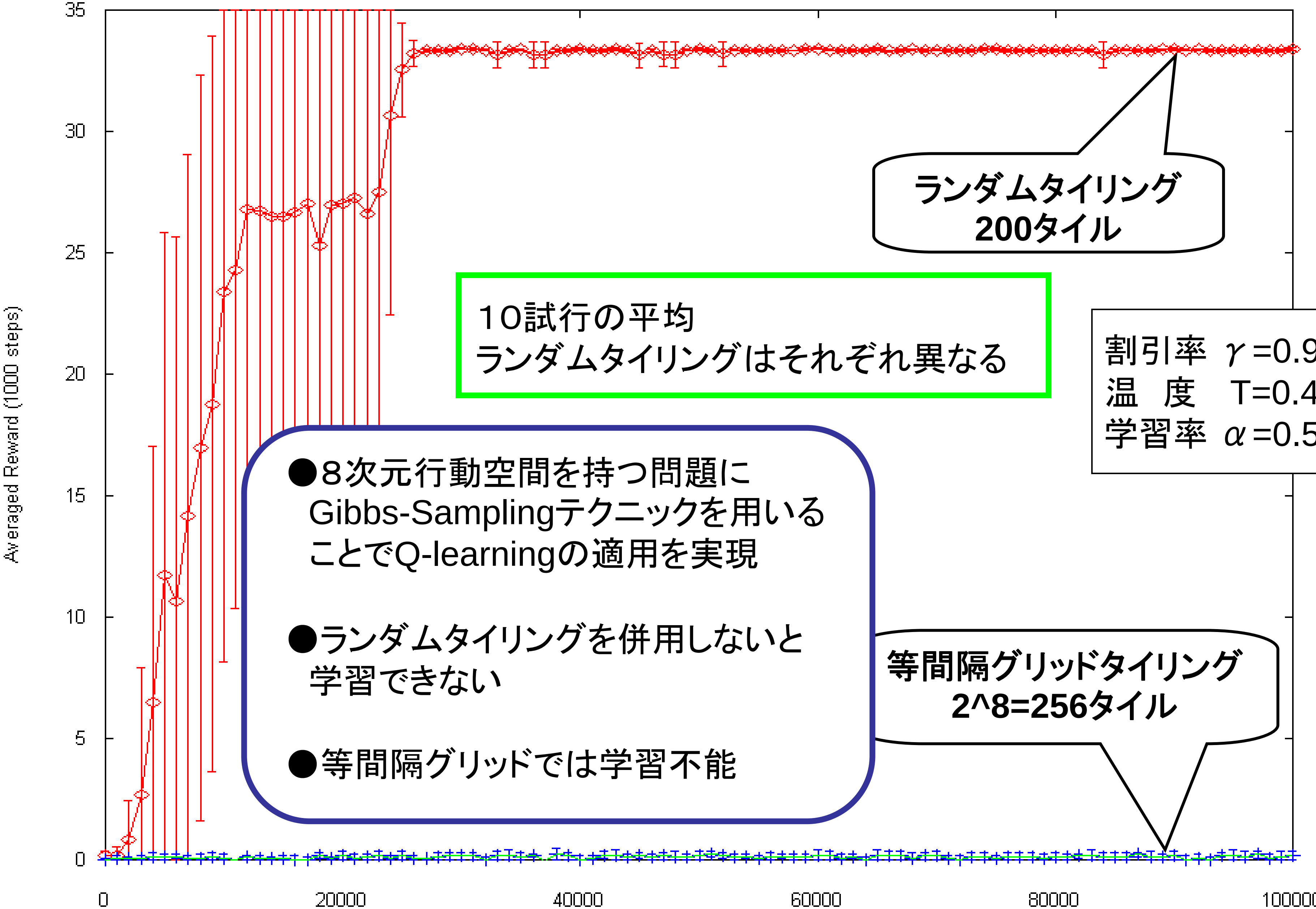
シミュレーション2結果： Multi-Joint Arm

状態8次元
行動8次元



シミュレーション2結果： Multi-Joint Arm

状態8次元
行動8次元

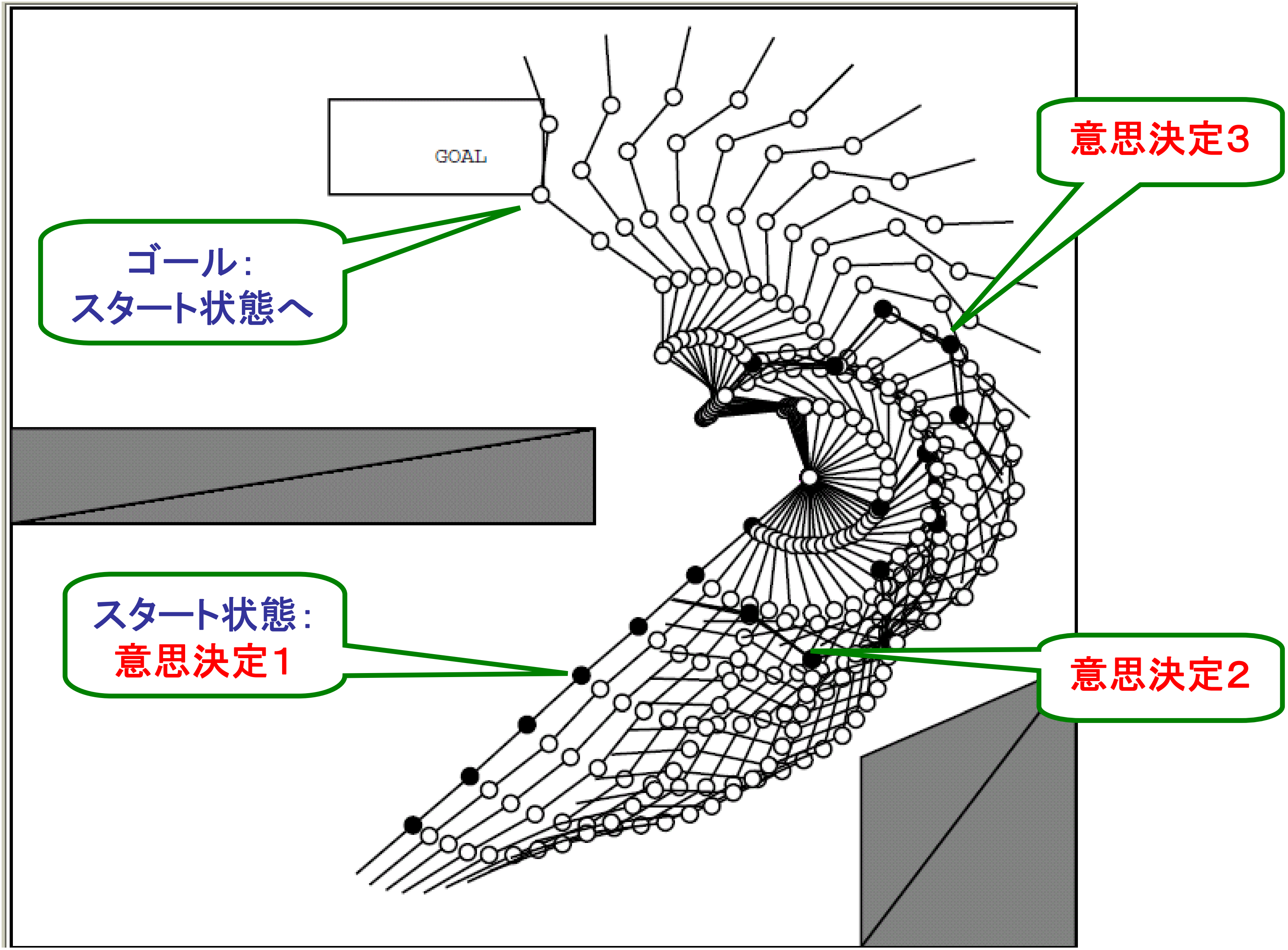


10試行の平均
ランダムタイリングはそれぞれ異なる

割引率 γ=0.9
温　度　T=0.4
学習率 α=0.5

- ●8次元行動空間を持つ問題にGibbs-Samplingテクニックを用いることでQ-learningの適用を実現
- ●ランダムタイリングを併用しないと学習できない
- ●等間隔グリッドでは学習不能

学習で得られた動作の一例



考察とまとめ

Q-learning

連続空間の関数近似

ボルツマン行動選択

MaxQ を求める
計算が近似に

ランダムタイリングによる高次元連続空間の汎化

- 実装が簡単
- 空間爆発を回避
- 足りなければタイルを増やすだけ
- 実装が問題に依存しない

Gibbsサンプリングによる行動選択処理

- 実装が簡単
- 空間爆発を回避
- 足りなければ計算反復の回数を増やすだけ
- 分割を細かくしても、計算はあまり変わらない

高次元連続状態-行動空間の強化学習問題へ実装

考察とまとめ

Q-learning

連続空間の関数近似

ボルツマン行動選択

MaxQ を求める計算が近似に

ランダムタイリングによる高次元連続空間の汎化

Gibbsサンプリングによる行動選択処理

・実装が簡単
・空間爆発を回避
・足りなければタイルを増やすだけ
・実装が問題に依存しない

・実装が簡単
・空間爆発を回避
・足りなければ計算反復の回数を増やすだけ
・分割を細かくしても、計算はあまり変わらない

高次元連続状態-行動空間の強化学習問題へ実装

考察とまとめ

Q-learning

連続空間の関数近似

ボルツマン行動選択

MaxQ を求める
計算が近似に

ランダムタイリングによる
高次元連続空間の汎化

- 実装が簡単
- 空間爆発を回避
- 足りなければタイルを増やすだけ
- 実装が問題に依存しない

Gibbsサンプリング
による行動選択処理

- 実装が簡単
- 空間爆発を回避
- 足りなければ計算反復の回数を増やすだけ
- 分割を細かくしても、計算はあまり変わらない

考察とまとめ

Q-learning

MaxQ を求める
計算が近似に

連続空間の関数近似

ランダムタイリングによる高次元連続空間の汎化

- 実装が簡単
- 空間爆発を回避
- 足りなければタイルを増やすだけ
- 実装が問題に依存しない

ボルツマン行動選択

Gibbsサンプリングによる行動選択処理

- 実装が簡単
- 空間爆発を回避
- 足りなければ計算反復の回数を増やすだけ
- 分割を細かくしても、計算はあまり変わらない

考察とまとめ

Q-learning

MaxQ を求める
計算が近似に

連続空間の関数近似

ランダムタイリングによる
高次元連続空間の汎化

- 実装が簡単
- 空間爆発を回避
- 足りなければタイルを増やすだけ
- 実装が問題に依存しない

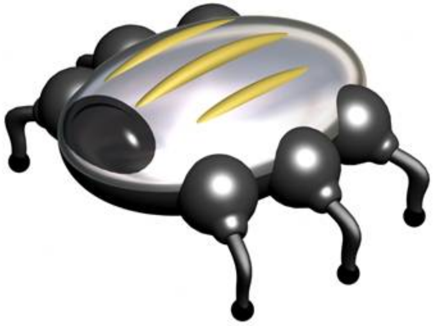
ボルツマン行動選択

Gibbsサンプリング
による行動選択処理

- 実装が簡単
- 空間爆発を回避
- 足りなければ計算反復の回数を増やすだけ
- 分割を細かくしても、計算はあまり変わらない

高次元連続状態-行動空間の強化学習問題へ実装

おわりに



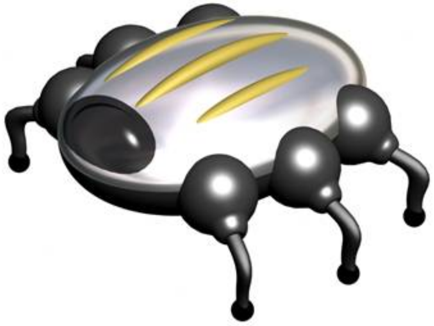
高次元連続な強化学習問題に対して、
誰でも実装できて学習させることが可能な
強化学習の基礎的実装方法を提案・実証

Gibbsサンプリング行動選択／ランダムタイリング特徴量生成

・学習性能はランダムタイリングの配置に依存
・状態空間で生成されたランダムタイルの30%が未使用

・タイルの適応的生成・削除へ拡張
・教示データを利用した効率的タイル生成

おわりに



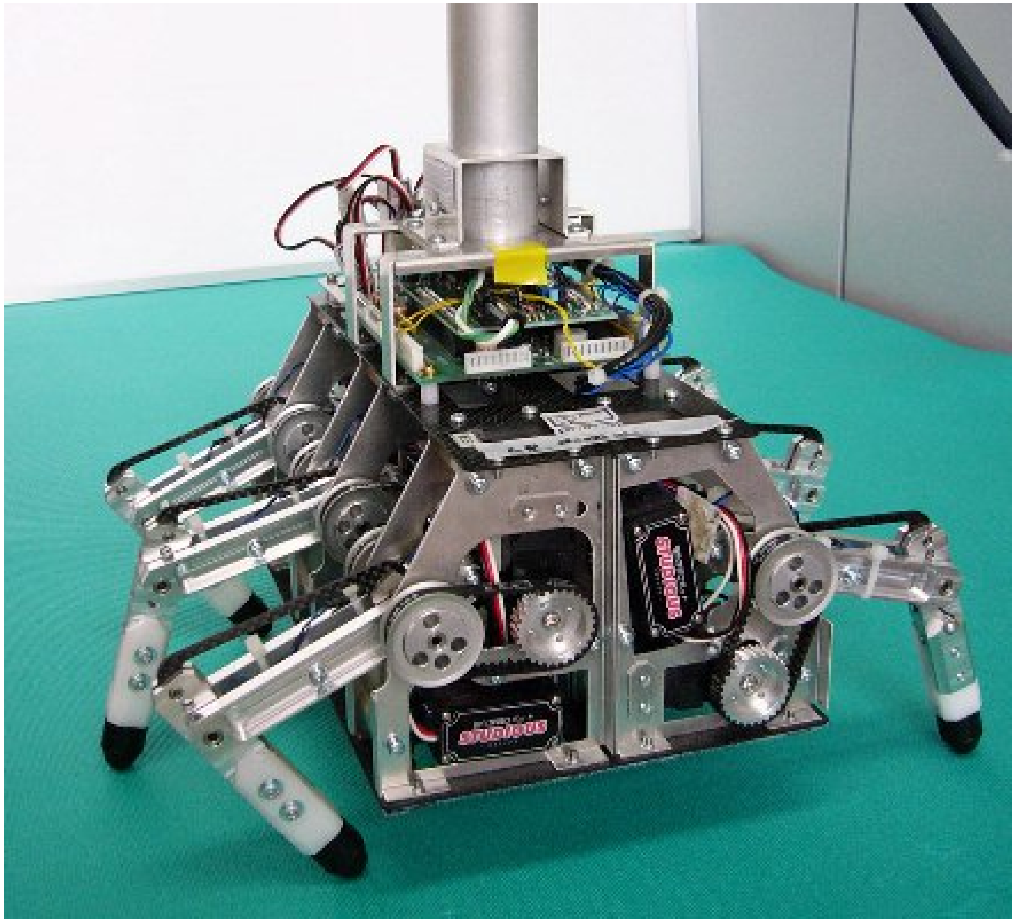
高次元連続な強化学習問題に対して、
誰でも実装できて学習させることが可能な
強化学習の基礎的実装方法を提案・実証

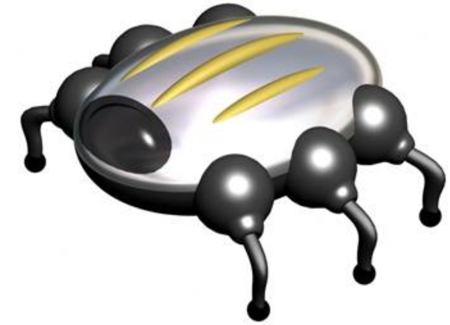
Gibbsサンプリング行動選択／ランダムタイリング特徴量生成

・学習性能はランダムタイリングの配置に依存
・状態空間で生成されたランダムタイルの30%が未使用

・タイルの適応的生成・削除へ拡張
・教示データを利用した効率的タイル生成

http://sysplan.nams.kyushu-u.ac.jp/gen/index.html





「強化学習ロボット：スタディアス」
愛・地球博
NEDOプロトタイプロボット展