F-029

Contextual Multi-armed Bandit の運転管理問題への適用性評価
Evaluation of Contextual Multi-armed Bandit for Automatic Operation Management

秋山 高行† Ｐｈｏｎｇ Ｎｇｕｙｅｎ† 大橋 洋輝†
Takayuki Akiyama Phong Nguyen Hiroki Ohashi

1. はじめに

人口減少，少子高齢化に伴う環境変化が進展する中、社会インフラのサービス品質を維持しながら，効率性を重視して再構築することは，先進国において喫緊かつ重要な課題である．とりわけ，熟練作業員の退職によるノウハウ断絶は，差し迫った課題である．他方，近年の IoT(Internet of Things)の進展，計算機性能向上，機械学習技術の発展と共に，いわゆる人工知能による効率向上・自動化が可能となってきている．中でも，従来では熟練者の経験と勘に基づく診断や運転管理を，人工知能で代替・支援することで，労働力不足解消，社会インフラの持続性に貢献することが可能であると考えている．

本研究では，運転管理の人工知能による自動化に主眼を置く．本稿で取り扱う運転管理は，変動する需要を満足する供給を維持することを示す．ここで，人工知能は，状況を考慮して，適切な運転をするようにパラメータを制御する行動を行う必要がある．例えば，ＡＴＭ(Automatic Teller Machine)の運転管理を考えると，利用者の入出金の需要を満足するようにＡＴＭ内に装填されている現金量を維持するタスクとなる．現金量は，管理者側でコントロールできない変化する需要に応じて，最適な現金量と最適な装填時期を決定する必要がある．このとき，最適性は，現金管理に関わるコストと供給による利用者満足とのバランスによって評価される．これは，他に，工場の生産量を需要に応じて管理するといった問題や，上水道における貯水量管理などとも類似する問題であり，一般的な問題の一つと考えている．

最適行動を自動的に獲得する手法として，強化学習の手法が知られている[1]．強化学習では，意思決定を行うエージェントと，エージェントが行動することで変化する環境が定義される．エージェントは選択可能な行動のセットから，与えられた目標値（報酬と呼ぶ）を最大化する行動を学習する．これまでは，ロボットの自律制御やゲームプレイなどの研究開発が行われていたが，オンライン上でユーザの行動結果を取得できるようになると，オンライン広告の最適化や，情報推薦に適用されるようになってきている．また，近年の IoT の普及により，実環境からの報酬信号をセンサなどにより観測できるようになったことから，活発に研究が実施されている．

実世界の運転管理問題では，ゲームでのスコアやオンライン広告のように CTR(Click Through Rate)のような一次元の目標値の最大化では表現しきれず，例えば，社会的信用や安全性といった経済価値に一意に変換できないような要素も考慮する必要がある．

本論文では，実世界の運転管理問題への Contextual Multi-armed Bandit Algorithm の適用性を評価した結果を報告する．Multi-armed Bandit Algorithm は，強化学習の一つであり，逐次観測された報酬信号に基づいて，複数の行動選択肢の中から最適な行動を学習する手法である．まず，対象とする運転管理問題を Contextual Multi-armed Bandit として定式化し，複数の行動価値の評価指標を持つ報酬信号の設計を実施する．そして，行動選択手法に未知の行動を確率的に選択する UCB(Upper Confidence Bound)アルゴリズムを導入，さらに現在の状況(context)を考慮することで，現状に応じた最適な行動を探索的に学習するようにし，状況判断に基づく意思決定を実現する．検証には，実際のＡＴＭの入出金需要を利用し，現金管理問題への適用性を評価する．

本論文は，下記のように構成される．2 章にて，これまでの従来研究を紹介し，3 章にて，対象とする問題の一般的な定式化と，Contextual Multi-armed bandit による方式について述べ，4 章で評価結果について述べる．

2. 関連研究

人工知能の目標の一つは，複雑な行動を人間のように知的に実施することである．これらの研究の中心的な位置に強化学習がある．これはある環境内のエージェント（行動する主体）が，現在の状態を観測し，取るべき行動を決定する．行動選択後に環境から得られる報酬により，行動と行動価値の関数を推定することで，報酬を最も多く得られるような最適な行動を獲得する[1]．強化学習においては，古くはバックギャモンでの成功が知られているが(TD-gammon) [4]，高次の状態空間の探索による計算時間の増大が問題である．近年では，Q 学習と非線形関数近似を組み合わせることで，このような高次の問題へのアプローチが実施されている．例えば，Google DeepMind が開発したエージェントが，単純なゲームプレイにおいてトッププレイヤーよりも高いスコアを獲得した，さらには，長年の難問とされてきた囲碁で人間のトップ棋士に勝利，などの成果が報告されている[2][8]．

他方，エージェントがユーザに最適な情報を選択する情報推薦や広告最適化においては，強化学習問題の一つの Multi-armed Bandit 問題によるモデル化が活発に行われている[9]．これは，k 本の行動選択肢(k-arm)の中から，最適な選択肢を知る，という問題である．行動価値を推定するために，未知の arm を選択する必要があるため，探索が必要だが，探索は最適行動ではないため，探索と活用のバランスを取ることが基本的な問題となる[10]．これまで，探索手法として，e-greedy, UCB, softmax, Thompson sampling などの探索手法が提案されている．また，行動価値が状況によって異なる場合，例えば，情報推薦の場合，対象ユーザの属性によって提示すべき情報が異なるため，ユーザの属性(context)に応じて arm の行動価値を推定する手法も提案されている[11] ．さらには，ベイズの定理を利用して事前情報を活用する手法も提案されている[12] ．

従来では，上記したように，行動とその結果が一意に決まることを仮定した静的な環境で活発に研究開発が行われている．また，web やゲーム分野では，機械による試

†（株）日立製作所 研究開発グループ, Hitachi, Ltd., Research & Development Group

行錯誤が比較的実施しやすいために，多くの手法が実用化され始めている．しかしながら，行動の結果が変動し，エージェントの行動がミッションクリティカルとなるような実環境での適用は未だ研究途上であり，実用化された例は少ない．また，ゲームのようにスコアが 1 次元で表されるものではなく，実際の環境では，金銭的な収益性のみならず，社会的信用や安全性などの経済価値として表現できないような要素を同時に考慮する必要がある．本研究では，そうした実環境での問題として運転管理問題を取り上げ，Contextual Multi-armed Bandit の適用方法を検討し，実際のデータにより評価した結果を報告する．

3. 提案方式

3.1 Multi-armed Bandit によるモデル化

まず，問題設定を一般化するため，環境は容量の設定された容器であるとみなす．この容器に離散的な時間間隔(t=1, 2, 3, …, T，ラウンドと呼ぶ)で，エージェントは容器へ注入する供給量を決定する．ラウンド t で発生する需要と供給量に応じて，容器内の残量が増減する．エージェントは，各ラウンドで満たした需要に応じた報酬を得る．一方で，需要を超過する供給を実施した場合には，超過分に応じたペナルティが発生する．さらに，残量が容量を超過，もしくは，0 を下回る場合には，供給失敗としてペナルティが発生する．また，供給にはコストが発生するものとし，供給 1 回あたりのコストが発生する．上記をまとめると，エージェントの目標は，コストを低減しつつ，報酬を最大化することであり，次式で表される収益を最大化することとなる．

$$\mathrm{Profit} = \sum_{t}^{T} (SD_t - \beta L_t - C_t - F_t)$$

$$SD_t = \begin{cases} D_t r & \text{if } D_t \le L_{t-1} + S_t \\ (L_{t-1} + S_t) r & \text{if } D_t > L_{t-1} + S_t \\ 0 & \text{otherwise} \end{cases}$$

$$L_t = \begin{cases} Capacity & \text{if } L_t \ge Capacity \\ 0 & \text{if } L_t \le 0 \\ L_{t-1} + S_t - D_t & \text{otherwise} \end{cases}$$

$$C_t = \begin{cases} \mathrm{Supply\,cost} & \text{if } S_t \ne 0 \\ 0 & \text{otherwise} \end{cases}$$

$$F_t = \begin{cases} \mathrm{fail\,cost} & \text{if } Capacity < L_t \text{ or } L_t < 0 \\ 0 & \text{otherwise} \end{cases}$$

ここで，Profit は全ラウンド T における全収益を示し，SDt は満たされた需要量に応じた報酬値， Ct は，ラウンド t で発生した供給コストである．Ft は，容量超過・供給不足が発生したコストである．Dt はラウンド t における需要量， St はラウンド t における供給量を表す．Lt は各ラウンド t 後の残量，Capacity は容量である．r は需要量から報酬値へ換算するための変換パラメータ， β は，残量からコストへ換算するための変換パラメータである．

本研究では，各ラウンドで，エージェントは供給量 St を[-Capacity, Capacity]の範囲で決定する。St=0 は供給しないことと等しい．

供給コストやペナルティは，発生したラウンドで報酬として観測されるが，実際には，前ラウンドまでの供給量の適切性に関連するものと思われる．直感的には，低頻度で大量の供給を行うことと，高頻度で少量の供給を行うことのどちらが選択するべきか，を決定する問題と考えられる．大量の供給を長い間隔で実施する場合は，供給回数が少なくなり供給コストは低くなるが，残量に応じたコストは高くなる．他方，少量を高頻度で供給する場合には，供給コストは高くなるものの，残量に応じたコストは低く抑えることができる．これらは，実際の需要変動に応じて調整されるべきものである．そこで，行動の価値は，その後のラウンドでの報酬にも影響するものとして，各ラウンドでの報酬を次の供給が実施されたタイミングで，次式のように計算する．つまり，次の装填が発生するまでのトータルの報酬から算出することにする．このため，行動の報酬の観測には遅れが発生することになる．

$$\mathrm{Reward}_t = \frac{\sum_{t}^{tn} (SD_t - F_t) - \mathrm{Supply\,cost}}{tn - t}$$

ここで，tn は，ラウンド t において次の Stn≠0 となるラウンドである．ここまでのシステムの処理フローを図１に示す．

現実の問題としては，エージェントはＡＴＭの管理者であり，ＡＴＭ内の現金量を入出金の需要量を満たすように保持しておく．現金輸送は次の日に実施できるものとすると，装填量の決定は，1 日ごとに行われる．現金量が需要に対して超過している場合には，残金に対して金利コストが発生することが知られている．また，現金輸送には警送が必要であるため，装填時には警送コストが発生する．一方で，現金量は容量を上回る，もしくは，現金が無くなってしまった場合には，サービスが停止するため，機会損失ならびに罰金や信頼の減少などのコストが発生する．これは，生産管理のような在庫を少なく保ちつつ生産量を最大化する問題や，上水道管理における貯水量を一定に保ちながら需要を満たす量の水を生産・供給する，といった問題とも同様と考えられ，一般性を持った問題設定と考えられる．

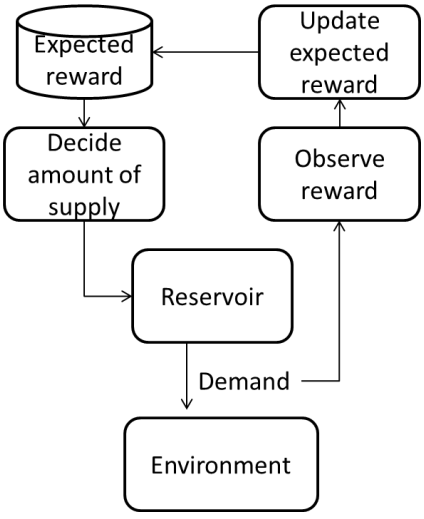

図１．システムの処理フロー
（環境と相互作用するのは Reservoir であり，システムでは供給量を決定する）

3.2 アルゴリズム

次の装填が行われる，つまり，報酬が確定した段階で，各 arm の価値を下記のように更新する．

$$R(a,\text{context}) = (1-\alpha)R(a,\text{context}) + \alpha\text{Reward}_t$$

α は learning rate であり，選択回数が増加すると小さくなるようになる．また，本方式では，容量過多・現金不足を回避するために，残量に応じた行動方策決定が適していると考え，context を残量で定義し，contextual bandit 方式を導入した．

このとき，UCB アルゴリズムに基づき，下記の評価値最大の arm を選択する．

$$E(a,\text{context}) = R(a,\text{context}) + \frac{\sqrt{2\log(\text{totalCount})}}{count(a)}$$

ここで，装填間隔を制御するために，経過日数も含めて，報酬を計測する．

以上の手続きを下記で表す．

Algorithm 1 UCB with delayed reward

Input: α, β, Capacity, r, Supply cost, fail cost $\in$R
for t = 1, 2, 3, … T **do**
  Observe context, current amount loaded
  Choose arm a = argmax E(a,context)
  Observe the reward
  if supply $\neq 0$
    Update R(a,context) based on accumulated reward
    Accumulated reward is initialized
  end if
end for

4. 評価実験

4.1 データセットと実験方法

評価用のデータとして，中国に設置されている９台の実際のＡＴＭにおける約 1 年間の入出金需要データを利用した．本実験で想定するＡＴＭでの取引は入金・出金が可能であり，各ＡＴＭの取引量と日数を表１に示し，需要データの一例を図２に示す．図２に示す需要量は出金量から入金量を引いたものとしている．評価では，それぞれのデータを 10 倍に拡張し 10 年分のデータとし，長期間での学習の挙動を確認する．

実験では，各ＡＴＭに対して，提案手法により装填量を決定した．ＡＴＭの容量は 1360000 とし，arm は，－1360000～1360000 の間を 13600 の間隔で離散値として生成した(arm 数 201)．ここで arm が負の値を取る場合は，ＡＴＭから現金を引き出すということに相当する．装填量は日々決定し，context を当日の残量として，0～200000，200000～500000，500000～700000，700000～1000000，1000000～1360000 の５つの context を設定した．報酬における各設定値は，r=1.0，α=0.1，β=0.05，Supply cost=280，fail cost=1000 とした．

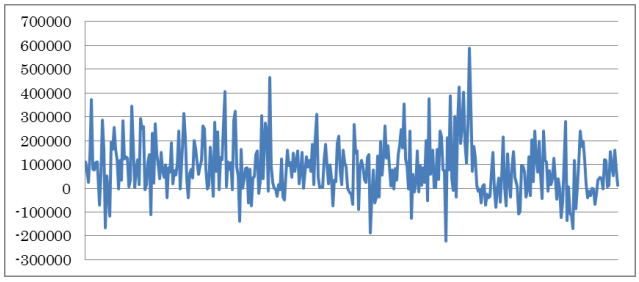

図２．ATM の需要変動の一例
（縦軸：需要量，横軸：日数）

表１．Statistical data of ATMs

ID	NUMBER OF TRANSACTIONS	NUMBER OF DAYS
20134001	29276	365
20134002	10352	363
20134009	96811	360
20134010	101935	360
20134011	105336	359
20134012	48713	351
20134013	57942	352
20134016	58476	340
20134017	91238	351

4.2 評価結果

まず，全ＡＴＭでの学習結果を示す．図３は，全ＡＴＭの平均収益（総収益をラウンド数で割ったもの）をラウンドに対してプロットし，図４は全ＡＴＭの平均の失敗回数（総失敗回数をラウンド数で割ったもの）をラウンドに対してプロットしたものである．

図３より，初期ラウンドでは探索が頻繁に行われるために多数の失敗が発生しコストが大きくなっている．失敗時には報酬が低くなるように設計されているため，次第に失敗しない行動の価値が比較的に高くなり，500 ラウンド付近から収益が安定する行動を獲得していることが分かる．また，適切な行動に収束していくことから，少なくとも局所最適解に達したものと考えられる．図４でも同様に，初期ラウンドでは失敗が大量に発生し，次第に失敗の回数が減少し，失敗の無い行動を獲得していることが分かる．今回の実験条件では arm 数が 201 であるため，201 ラウンドまで失敗回数が増大し，それ以降は失敗しない期待報酬の高い arm が選択されていることが示されていると考えられる．

実際の運転行動の一例を図５，６に示す．初期ラウンド（図５）では探索による多数の装填の発生と失敗が起きており，様々な供給行動が実施されているが，後期のラウンド（図６）では適切な供給の頻度と量が学習され，残量を安定的に維持し，失敗の起こらない行動を実施していることが分かる．後期のラウンドを見ると，残量が 200000 になると供給が実施されるようになっており，こ

れは context を残量として導入したことにより，残量の少ない状況において供給を実施し，残量が多い状況では，供給しないことが収益に対して適した行動であることを学習したものと考えられる．また，UCB の収束性により，探索的な行動が実施されなくなっていると考えられるが，残量が 200000 以上の context では，探索的な行動が時折発生していることが分かる．これは，残量が多い状況での行動回数が，残量の少ない状況での行動回数よりも比較的に少ないために，UCB が未だ収束していないとも考えられる．

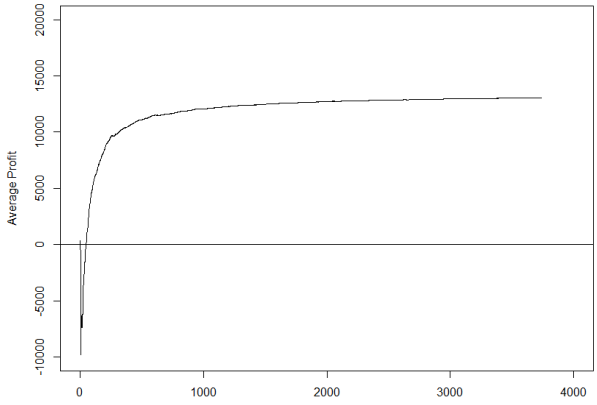

図３．平均収益の収束性
（縦軸：平均収益，横軸：ラウンド数）

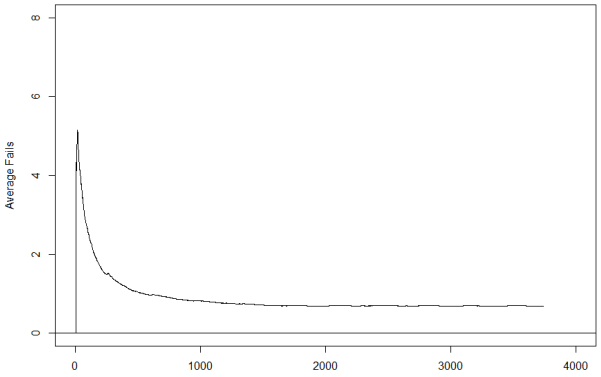

図４．平均失敗回数の収束性
（縦軸：平均失敗回数，横軸：ラウンド数）

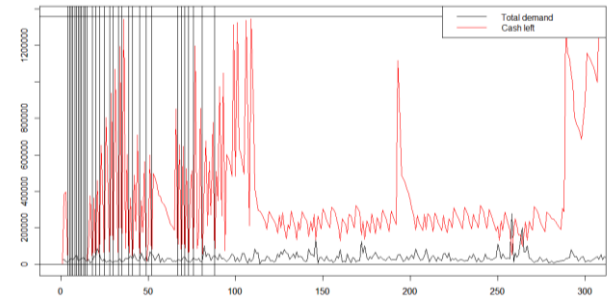

図５．ID20134002 の装填量の初期ラウンドでの推移
（黒線：需要変動，赤線：残量，縦軸：装填量，横軸：ラウンド数，縦棒は失敗時）

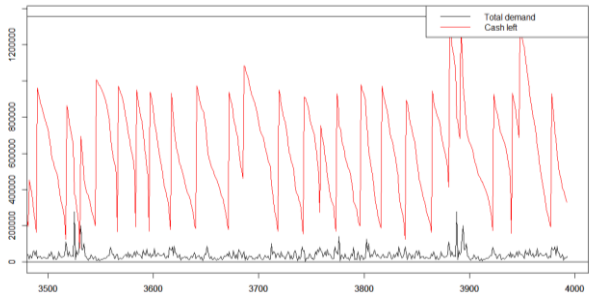

図６．ID20134002 の 3800～4000 ラウンドでの装填量
（黒線：需要変動，赤線：残量，縦軸：装填量，横軸：ラウンド数）

次に，fail cost を変化した場合の平均失敗回数の結果を図７に示す．fail cost 増加に応じて，平均失敗回数が減少することが分かる．これは，fail cost を高く設定すると，失敗時の報酬が低くなるために，より積極的に失敗しない行動を獲得するようになっているものと考えられる．一方で，図８に各 fail cost での平均収益の収束を示す．ここで，各平均収益は，それぞれの fail cost で運転行動を獲得した後に，同一の fail cost=1000 によって収益を計算したものである．fail cost を高く設定する場合に，学習の収束が速くなっている．収益最大化の目標の下では，失敗を回避することが最適行動であるために，失敗に対するペナルティをより高く設定する方がより適した行動を速く獲得しやすいということが考えられる．しかし，最終的な平均収益において，fail cost=1000 と 100000 ではほぼ同一であるが，fail cost=10 の場合の収束は，未だ収束していないと見える．fail cost が小さい場合には，失敗時の収益が他の成功した収益との差異が小さくなっているため，fail cost が収束速度に影響している可能性があると考えている．実際の環境では，失敗というのは社会的な信頼に関わるものであり，これを収益と言った金額換算に対してどの程度の重みを置くか，という問題は人間の判断の問題である．そういった複数の評価尺度が存在する中で，報酬信号をどう設定するかというのは，最適な行動の獲得において重要な問題であることが示されたものと考えている．

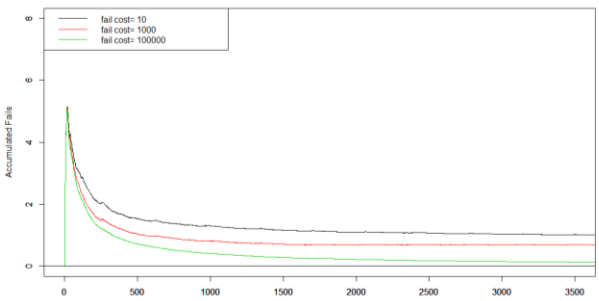

図７．fail cost と平均失敗回数の関係
（縦軸：平均失敗回数，横軸：ラウンド数，実線：fail cost=10 の場合，赤線：fail cost=1000 の場合，緑線：fail cost=100000 の場合）

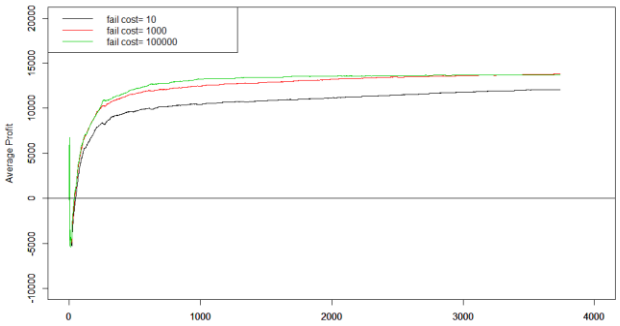

図８．fail cost と平均収益の関係
（縦軸：平均収益，横軸：ラウンド数，実線：fail cost=10 の場合，赤線：fail cost=1000 の場合，緑線：fail cost=100000 の場合）

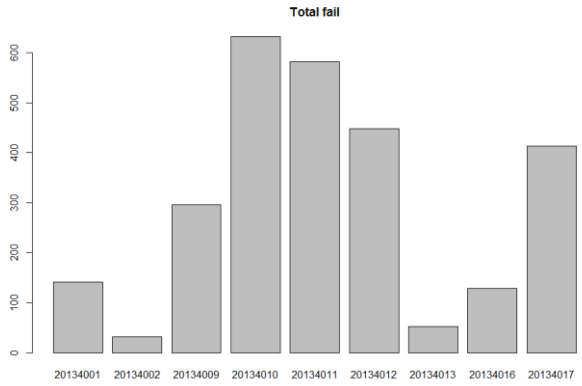

図９．失敗回数（fail cost=1000）
（縦軸：失敗回数，横軸：ATM ID）

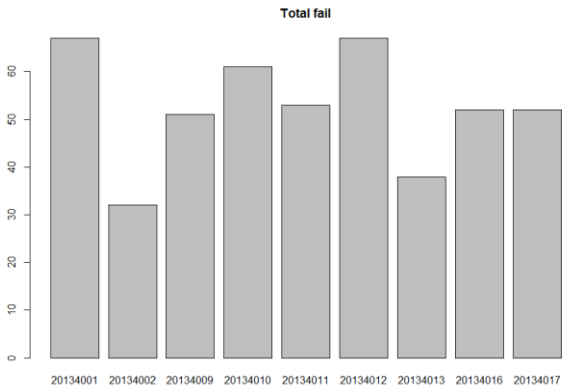

図１０．失敗回数（fail cost=100000）
（縦軸：失敗回数，横軸：ATM ID）

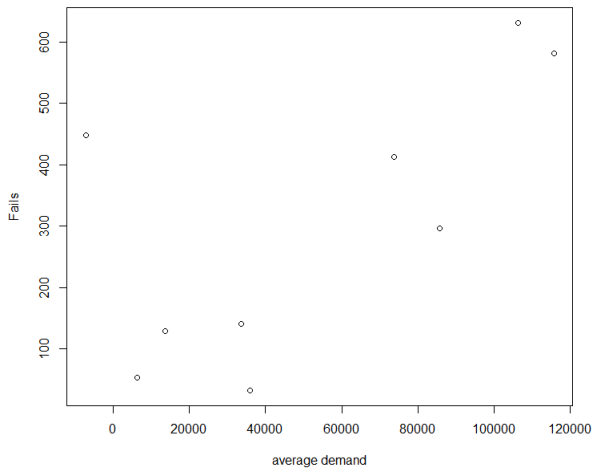

図１１．失敗回数と平均需要の関係（fail cost=1000）
（縦軸：失敗回数，横軸：平均需要）

最後に，需要変動傾向の異なる場合の学習結果について，各ＡＴＭでの評価結果と共に述べることにする．図９は fail cost=1000，図１０は fail cost=100000 の場合の各ＡＴＭでの総失敗回数を比較したグラフである．fail cost=1000 では，ＡＴＭ毎に失敗回数にバラツキが存在するが，fail cost=100000 では，バラツキと失敗回数ともに小さくなっていることが分かる．

それぞれの場合に，各ＡＴＭの平均需要量と失敗回数をプロットしたものを図１１，１２に示す．図１１より，平均需要量が大きいＡＴＭでは，失敗回数も多くなっていることが分かる．これは，平均需要量が多いＡＴＭでは，残量の変動が大きくなるため，本論文で設定した残量の context の最小値 200000 以下と 200000～500000 の context の間を頻繁に行き来することが起こり，最適な行動として，供給と無供給の間での報酬の期待値が一意に定まりきらなかったものと考えられる．これに対して，図１２では，失敗回数と平均需要量の間の相関は無くなっており，fail cost を十分に大きく設定することで，設定した context に対する最適な行動を獲得しているものと考えられる．fail cost を大きくすると，context が 200000～500000 での最適な行動として，報酬の期待値に有意な差が生じ，供給行動が学習されていた結果と思われる．

これを示すため，図１３と１４に，ATM ID=20134010 の fail cost=1000，fail cost=100000 の場合の後期のラウンドでの運用結果を示す．図１３では，低残量での低供給で推移し，失敗が頻繁に発生しているが，fail cost を高く設定した図１４では，残量を比較的高いレベルで維持するように供給が実施され，失敗を回避していることが分かる．これより，context の設定値に対して fail cost を適切に設定することが，各 context での供給行動の最適化に影響することが示唆される．前述と同様に，fail cost の適切な設定は，失敗の回避以外にも，様々な条件への汎用性にも影響することが分かった．

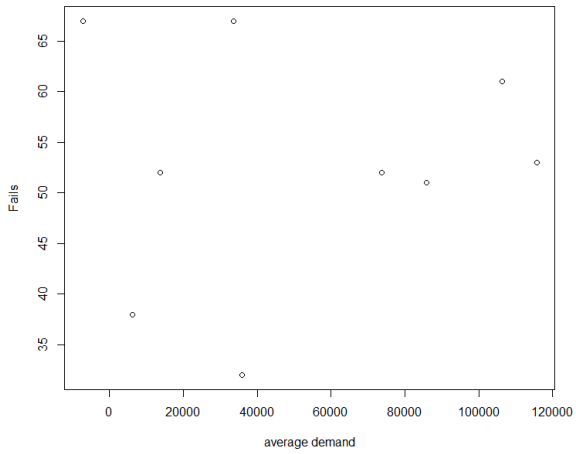

図１２．失敗回数と平均需要の関係（fail cost=100000）
（縦軸：失敗回数，横軸：平均需要）

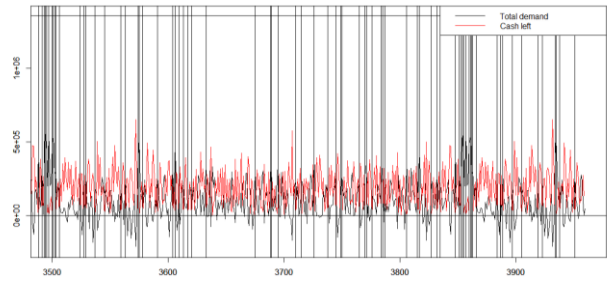

図１３．ID20134010 の 3800～4000 ラウンドでの装填量（fail cost=1000）
（黒線：需要変動，赤線：残量，縦軸：装填量，横軸：ラウンド数，縦棒は失敗時）

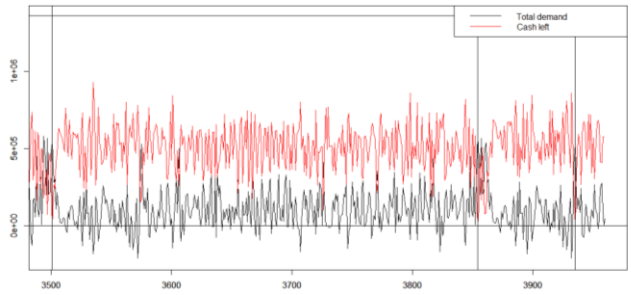

図１４．ID20134010 の 3800～4000 ラウンドでの装填量（fail cost=100000）
（黒線：需要変動，赤線：残量，縦軸：装填量，横軸：ラウンド数，縦棒は失敗時）

5. 結論と今後の課題

本論文では，人工知能による運転管理の自動化を目的として，Contextual Multi-armed Bandit Algorithm による自動運転管理方式を検証した．変動する需要を安定的に満たす問題を Multi-armed Bandit 問題として定式化し，報酬信号を一連の行動選択のトータルのコスト収益として設定した．この際に，収益に影響する３つの要素を導入した．行動選択手法に未知の行動を確率的に探索する UCB(Upper Confidence Bound)アルゴリズムを導入，さらに，状況を表す context を導入することで，変化に適応的な行動選択を実現するようにした．評価では，ＡＴＭ内の現金量を管理する問題を取り上げ，実際のＡＴＭの入出金需要を利用して評価した．その結果，9 台全てのＡＴＭにおいて収益向上と失敗の低減を達成することを確認し，有効性を確認した．

今後の課題としては，以下が残されている．

（１）性能向上

本研究では，提案手法が収益を確保することを確認したが，熟練者の収益と比較すると同等かそれ以下である．今後は，さらなる収益拡大に向けて，現場の環境情報や，過去の熟練者の知見などを取り込めるような柔軟性のある手法の検討を進める予定である．また，過去の需要変動のパターンなども考慮するために，context をより複雑な信号により判定する手法も必要である．

（２） context などの条件設定の汎用化

本研究では，context や arm，報酬などの条件設定は，事前に実施されている．評価結果より，これらの設定値は最終的に学習される行動にも影響するため，ユーザが調整しなければならないと考える．今後は，強化学習についての知識を持たないユーザでも簡易に設定できるようなシステム構成の検討を進める予定である．

（３） 実環境での検証

本研究は，中国の９つのＡＴＭという限られた条件設定で実施されたため，中国内の他の場所に設置されたＡＴＭでの動作や，他国での動作，さらには他業種での動作についての評価が必要である．例えば，上水道の配水コントロールや，生産管理などにも適用可能と考えている．それぞれのドメインに依存した要素を取り込む必要があると考えられる．

参考文献

[1]Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. Vol. 1. No. 1. Cambridge: MIT press, 1998.

[2] Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, et al. Human- level control through deep reinforcement learning. Nature, 518(7540):529- 533, 2015.

[3]Gerald Tesauro. Temporal difference learning and td-gammon. Communications of the ACM, 38(3):58-68, 1995.

[4] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. SIAM Journal on Computing, 32(1):48- 77, 2002.

[5] Jordan B. Pollack and Alan D. Blair. Why did td-gammon work. In Advances in Neural Information Processing Systems 9, pages 10- 16, 1996.

[6] Gerald Tesauro. Temporal difference learning and td-gammon. Communications of the ACM, 38(3):58-68, 1995.
[7] John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. Automatic Control, IEEE Transactions on, 42(5):674- 690, 1997.
[8] David Silver, et al. Mastering the game of Go with deep neural networks and tree search, Nature, 529, 484-489, 2016
[9] D. A. Berry and B. Fristedt. Bandit Problems: Sequential Allocation of Experiments. Monographs on Statistics and Applied Probability. Chapman and Hall, 1985.
[10] Li, Lihong, et al. A contextual-bandit approach to personalized news article recommendation. Proceedings of the 19th international conference on World Wide Web. ACM, 2010.
[11] Langford and T. Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In Advances in Neural Information Processing Systems 20, 2008.
[12] D. Agarwal, B.-C. Chen and P. Elango. Explore/exploit schemes for web content optimization. In Proc. of the 9th International Conf. on Data Mining, 2009.