

イノベーションのジレンマにおける 「累次的破壊」についての社会シミュレーション分析

○小田樹 高橋真吾 (早稲田大学)

Social Simulation of Sequential Process of Multiple Disruptive Innovations in the Innovator's Dilemma

* I. Oda and S. Takahashi (University of Waseda)

概要— 市場でトップシェアを誇る既存製品が主要性能の面で劣る破壊的イノベーションにシェアを奪われてしまう「イノベーションのジレンマ」という現象について、従来理論では性能進化に関して線形モデルを用いて説明されることが通例であった。しかし、近年「軽量パソコン市場」のような線形モデルでは説明がつかない、2つ以上の破壊的イノベーションが同時に関わる破壊現象が発生している。本研究ではこのような現象を「累次的破壊」と定義し、Richard N. Foster の「技術のSカーブ理論」を用いてモデル化する。またエージェント・ベース・シミュレーションを用いて累次的破壊が発生する状況とそれに対する対策のタイミングの判断基準について分析する。

キーワード: イノベーションのジレンマ, 技術のSカーブ, ネットブック, エージェント・ベース・シミュレーション

1 研究背景と研究課題

イノベーションの多くは既存製品よりも高い性能を生み出す効果のある「持続的技術」を元に生まれる「持続的イノベーション」である。このようなイノベーションは登場時に既存製品の主要性能（既存企業が積極的に技術開発を行い、消費者にも性能の高さをアピールしているものこと）よりも性能が高いことが多く、持続的技術の採用という経営判断も比較的容易である。

しかし、時に持続的イノベーションが現れていないにも関わらず、既存企業が占有する市場構造が破壊されてしまうことがある。これはChristensen¹⁾が提唱する、「破壊的イノベーション」と呼ばれるものであり、市場における主要性能とは異なる性能（本論文では副次性能と呼称する）の面で強みを持っている。しかし多くの場合で登場当初、主要性能の面では既存製品に劣っていることから既存企業が軽視しがちであり、時間とともに性能進化をすることによってシェアを奪われてしまうことがあるのである。このような現象を「イノベーションのジレンマ」と呼ぶ。

イノベーションのジレンマの生じる原因として、「既存製品が顧客の求める性能を大幅に超え、破壊的イノベーションが顧客の求める性能に近づくこと」と「バリューネットワークによって戦略変更が遅れる」ことを挙げている。破壊的イノベーションは当初は主要性能で劣っていても副次性能においては優れているため、既存企業の優良顧客とは異なる顧客層に少ないながら売れることになる。そして破壊的イノベーションは時間をかけて性能進化することにより、市場の消費者の求める主要性能の水準に追いつくことがある。原因としては企業の開発は消費者が求める性能の向上速度よりも速いことが挙げられる。同様に既存企業も性能進化を続けるが、元々消費者が求める性能レベルの商品であるため、消費者にとって過剰な性能となってしまう。破壊的イノベーションは主要性能以外でも強みがあるため、消費者が求める性能水準に追いつくと、性能過剰を起こした既存製品から優良顧客ですらも流出してしまうのである。しかし既存企業は自身の

優良顧客に対して確立した利益モデルがあったり、株主であったりと深いつながりがあるため、主要性能の改善をより求める優良顧客の意見にどうしても傾倒してしまい、破壊的イノベーションへの移行をなかなか決断することができない。このような優良顧客との繋がりはChristensen¹⁾の述べる「バリューネットワーク」の一種であり、優良顧客から脱却した製品開発・ビジネスモデル構築がなかなかできないことが性能過剰および破壊現象の原因と呼ばれる。

Christensenは性能進化およびそれに伴う性能過剰の原理を説明する際、Fig.1のような2つの性能の推移を直線的に捉える線形モデルを提唱しており、エージェント・ベース・シミュレーションを用いたイノベーションのジレンマに対する施策の分析も山本ら²⁾や矢野ら³⁾の研究に代表されるように多くの研究にて性能進化は線形モデルを前提としている。

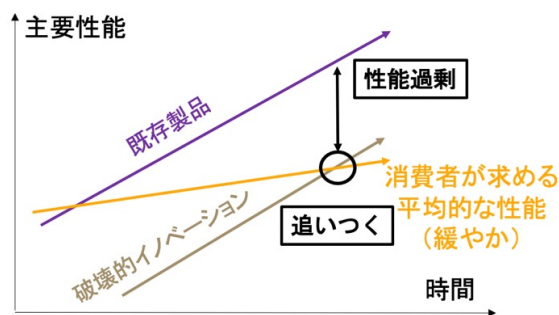


Fig. 1: 破壊理論における線形モデル

しかしこの線形モデルが前提では説明がつかない事例も発生してきている。その一例が「軽量パソコン市場」である。2000年代に軽量パソコン市場を形成したのはノートパソコンであったが、2010年に一気に売り上げを伸ばした破壊的イノベーションであるタブレット端末によって2013年までにシェアを抜かれてしまった。この二つの製品間においては従来の破壊理論が適用することができるが、実は軽量パソコン市場においてはタブレットが出現する前に「ネットブック」と呼

ばれるノートパソコンにとっての破壊的イノベーションが2005年に出現しており、Fig. 2のように2007~9年にかけて十数%のシェアを獲得していた⁴⁾⁵⁾⁶⁾。

しかしネットブックにとってもタブレットは破壊的イノベーションであり、ノートパソコンと共にシェアを奪われてしまった⁷⁾。線形理論を前提とすれば、性能進化を遂げ、一定のシェアを獲得できた破壊的イノベーションは順にトップシェアを獲得するはずである。

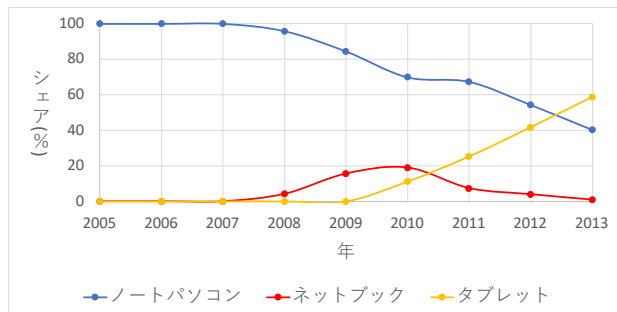


Fig. 2: 軽量パソコンの事例のシェアの実データ

このように線形理論では説明がつかない事象は「映像レンタル市場」⁸⁾や「音楽メディア市場」⁹⁾においても見られ、イノベーションのジレンマにおける性能進化に関して理論の改変が必要であると言える。

性能進化そのものの理論としてはFoster¹⁰⁾の「性能は投入資源量に対してSカーブのような推移を辿る」と述べた「技術のSカーブ理論」が挙げられる。一方でChristensenは破壊理論の提唱当初に、Fosterの理論に関して破壊理論への適応は不適切と述べていた¹⁾ものの、2018年の論文¹¹⁾にて、近年のイノベーションは急速に進化するものも多く線形を仮定することが誤った判断につながる可能性を問題点としてあげている。Downesら¹²⁾も近年のイノベーションの特徴として急速な進化性について言及しており、加えてイノベーションが次々と出現し、十分にシェアを獲得できずに次にやってくるイノベーションにシェアを奪われてしまうことも特徴と述べている。これらの示唆は前述の線形モデルでは説明がつかない事例においても共通し、線形モデルの傾きを変化させるだけではなく、性能進化の急峻性と次々と破壊的イノベーションが現れる状況を再現したモデルの構築の必要があると言える。

また追われる立場の企業の施策の実行タイミングの判断基準についても様々な見解がある。施策の1つとして新たなイノベーションへの完全移行が挙げられるが、Christensenは消費者の要求する性能水準と破壊的イノベーションの性能を、線形推移を前提に予測すべきだと従来述べていた¹⁾。しかし近年になり線形に捉えることの問題点についても言及している¹¹⁾。一方でFosterはイノベーション同士の性能進化を、Sカーブを前提に予測すべきだと述べている¹⁰⁾。

そのため、イノベーターは次々と急速な進化を伴って出現するイノベーションへの対策が必要であり、どのタイミングで施策を実行するか判断がより難しくなっていることを問題点として挙げている。従来の破壊現象ではChristensenの基準がより効果があることがわかっているものの、これが前述の「軽量パソコン市場」のような破壊現象では両者にどれほどの効果に違いがあるのかについても検証する必要がある。

2 研究目的

本研究では「軽量パソコン市場」のような市場にて多くのシェアを獲得している「既存製品」を脅かす「第一の破壊的イノベーション」が市場に存在する状況下において、さらに後から出現した「第二の破壊的イノベーション」が前出の製品のシェアを奪う状況下を想定する。この状況を「累次的破壊」と定義し、各製品の性能進化のどのような組み合わせの時に発生するかをシミュレーションによって定量的に分析する。

そして実際に累次的破壊が発生する状況下において「第一の破壊的イノベーター」が「第二の破壊的イノベーション」にどのタイミングで移行するかをどのような基準で判断するべきかを定量的に分析する。

3 累次的破壊の定義

本研究の対象現象である「累次的破壊」は以下の3つの条件を全て満たすものとする。

- ① ある市場における「既存製品」に対して「第一のイノベーション」が出現し、「第二のイノベーション」が誕生している場合
 - ② 「第一のイノベーション」と「第二のイノベーション」がどちらもChristensen¹⁾が定義する「破壊的イノベーション」に該当する場合
 - ③ 「第一のイノベーション」が一定のシェアを獲得するも、「既存製品」を追いやる前に「第二のイノベーション」が連続的に「第一のイノベーション」や「既存製品」からシェアを奪った場合
- ①と②を満たしていても、第一の破壊的イノベーターが既存企業のシェアを完全に奪い、その後第二の破壊的イノベーターにシェアを奪われるという状況下では従来の線形モデルで説明がついてしまう。

しかしFig. 3のように一時的に第一の破壊的イノベーターがトップシェアを奪ったとしても、既存企業のシェアのほとんどを奪う前に第二の破壊的イノベーターシェアを奪われてしまう場合は「既存企業」と「第一の破壊的イノベーター」の破壊現象と「第一の破壊的イノベーター」と「第二の破壊的イノベーター」の破壊現象は同時進行で発生していたと言えるため、連続性があると言える。

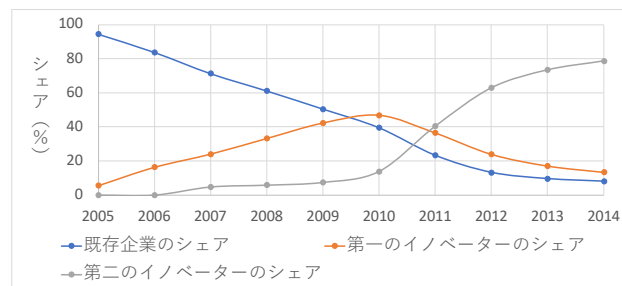


Fig. 3: 連続的な破壊のシェア推移の一例

4 モデルと妥当性検証

4.1 モデルとアルゴリズム

本研究では、消費者と企業が商品を通じて相互作用

を起こす状況を再現することができる CAMCaT (Co-evolutionary Agent-based Model for Consumers and Technologies) フレームワーク¹³⁾を用いて累次的破壊を再現することができるモデルを構築した。本モデルの特徴としては以下の3つが挙げられる。

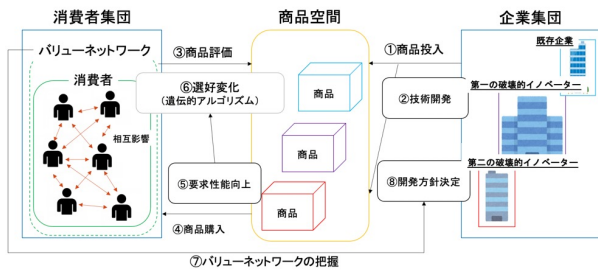


Fig. 4: 本モデルの概要

- (ア) 商品性能と消費者の求める性能を元にした購買
- (イ) 商品性能および消費者の要求性能の非線形進化
- (ウ) バリューネットワークの影響

本モデルでは1世代ごとに Fig. 4 のアルゴリズムを実施する。

まず消費者は企業が投入した商品を購入する際に、商品の性能を表す商品属性値(Pro)と自身の求める性能を表す要求属性値($DemandValue$)を元として、技術属性ごとに需給割合($SupplyDemandRatio$)を算出し評価値とする。なお商品属性値はその企業の提供できる性能を表す「技術属性値」($technicalValue$)に準ずる。この際、1を超えた場合は1に置き換える。これにより破壊理論における性能過剰・性能不足の概念を表現する(特徴(ア))。

$SupplyDemandRatio_t$

$$= \begin{cases} \frac{Pro_t}{DemandValue_t} & (Pro_t < DemandValue_t \text{ のとき}) \\ 1 & (Pro_t \geq DemandValue_t \text{ のとき}) \end{cases} \quad (1) \quad technicalValue_t$$

t : モデルにおける世代数

$$Utility = utilityWeight_1 * Repeat_t + utilityWeight_2 * SupplyDemandRatio_t \quad (2)$$

そして式(2)のように需給割合を全技術属性分の平均を取った上で再購入数(何世代連続で該当する企業の商品を買っていたか) ($Repeat$)を効用ウェイト ($utilityWeight$)を掛けた上で足し、効用値($Utility$)を算出する²³⁾。この効用値がカットオフ値($Cutoff$)を下回っていた場合は0に置き換える²³⁾。そしてロジットモデルに従って²³⁾効用値を用いて購買を決定する。

購入後、消費者は購入した商品と自身の要求属性値の差分を基に要求属性ウェイト($DemandWeight$)を掛けた上で要求属性値を変化させる。ただし消費者は企

業の性能進化に比べ緩やかに求める性能水準、つまり要求属性値を進化させていく¹⁾ため、要求属性向上割合($ImproveRatio$)を除算することにより表現する。この割合は主要性能と副次性能において異なる¹⁾。要求属性ウェイトは消費者がどの属性を重視するかを表している。差分は一定ではないため、要求属性値も非線形に進化する。

$$DemandValue_{t+1} = \frac{DemandWeight_t}{ImproveRatio} * (Pro_t - DemandValue_t) + DemandValue_t \quad (3)$$

差分がマイナスの場合は0で置き換える。その後遺伝的アルゴリズムにより消費者の相互影響を再現する。

一方で企業は商品投入を完了させると、前世代で決定した技術開発方針、つまり「技術属性ウェイト」に基づいて性能進化を行う。本モデルではSカーブによる進化を導入するが、Fosterは性能進化を推定する場合はロジスティック関数を用いている¹⁰⁾ため、本モデルにおいても「技術関数」として導入している。

$$f(t) = \frac{K}{1 + \exp\{-a(t - t_0)\}} + C \quad (4)$$

a, K, t_0, C : 定数(モデルにおける初期値)

企業が性能進化に費やすことができる資源量には限りがあり、どの性能を重視するかによってその量には差が生まれる。本モデルではFosterの述べる「投入資源量」¹⁰⁾に対応する概念として「技術属性ウェイト」($technicalWeight$)を導入し、式(5)のように技術関数の増分にかけることにより非線形進化を表現する。

$$= technicalWeight_t * \{f(t) - f(t-1)\} + technicalValue_{t-1} \quad (5)$$

式(3)~(5)の企業と消費者のアルゴリズムは本モデルの特徴(イ)に該当する。

企業は自身の製品を購入してくれた消費者の要求属性値の平均値($DemandAverage$)を受け取り、次の世代の属性ごとの要求属性ウェイトを決定する。ここで更新ウェイト($updateWeight$)は消費者の声と自身の前世代の方針をどれほど重要視するかを表す。これは本モデルの特徴(ウ)に該当する。

$technicalWeight_{t+1}$

$$= updateWeight_1 * \frac{DemandAverage_{tn}}{\sum_{tn=1}^4 DemandAverage_{tn}} + updateWeight_2 * technicalWeight_t \quad (6)$$

上記までが施策シナリオ分析の実施の可否に関わらない「通常のアプローチ」である。

本研究では本モデルの妥当性検証(→4.2)を行ったのち、技術関数の組み合わせを変化させることによって数十種類の「状況シナリオ」を構築し、累次的破壊が発生する組み合わせを探索する(→5.2)。その後累次的破壊が発生する「状況シナリオ」において第一の破壊的イノベーターがどのタイミングで施策を取るべきかの判断基準の有効性を検証するための「施策シナリオ」を構築し、分析する(→5.3)。以下はその「施策シナリオ」に関するアルゴリズムである。

施策シナリオ分析においては、Christensen および Foster に共通する「新たなイノベーションへの完全移行」を前提に、移行を決定するタイミングをどのように予測するかについて分析を行う。旧来の2者の基準に加え、それを混合させた判断基準の3つを分析する。

- I. 第二の破壊的イノベーションの主要性能と消費者の求める主要性能が線形推移すると仮定、比較 (Christensen の基準)
- II. 第二の破壊的イノベーションと第一の破壊的イノベーションの主要性能がロジスティック関数に従って推移すると仮定、比較 (Foster の基準)
- III. 第二の破壊的イノベーションの主要性能がロジスティック関数に従い、消費者の求める主要性能が線形に推移すると仮定、比較 (混合基準)

上記の判断基準において、線形で予測する場合は、第二の破壊的イノベーションが参入した世代から現世代までの技術属性値(要求属性値)より1世代あたりの傾きを算出し、2年後(本モデルでは1世代は3ヶ月である²⁾³⁾ため、8世代)の値を予測する。ただし消費者の平均の要求属性値を取得するのは現実世界においても誤差が発生してしまうことが考えられるため、消費者における予測誤差範囲(*consumerPredictErrorRange*)で一様乱数に従って予測値に誤差が発生する。

また、ロジスティック関数に従って予測する場合は以下の式(7)に基づいて行う。

$$\begin{aligned} \text{predictValue}_{t+8} \\ = \text{predictWeight} * \{f(t) - f(t-8)\} \\ + \text{technicalValue}_t \end{aligned} \quad (7)$$

上記の予測ウェイトは、第一の破壊的イノベーションの予測においては自身の技術ウェイトを代入する。なお、技術ウェイトは世代ごとに変化するため、実際の2年後の値とは誤差が小さいながら発生する。第二の破壊的イノベーションにおいては誤差ウェイト(*errorWeight*)を予測ウェイトに導入することによ

て実際の予測のズレを表現する。第二の破壊的イノベーションの予測を開始した世代において、誤差ウェイトは以下の範囲において一様分布によって決定される。

$$\begin{aligned} \text{predictBasement}^{-\text{errorRange}} \\ < \text{errorWeight} < \text{predictBasement}^{\text{errorRange}} \end{aligned} \quad (8)$$

ロジスティック関数を用いた予測は商品が市場に出てから時間が経てばより容易になる¹⁰⁾ため、世代が進むごとに誤差範囲は誤差減衰値(*errorAttenuation*)分だけ小さくなる。ただし企業は予測において第二の破壊的イノベーションを実際の伸びよりも過大評価してしまうか、それとも過小評価してしまうかに関しては世代ごとに大きくは変化しないため¹⁰⁾、誤差ウェイトは一度決定した後はある程度の範囲に沿ってのみ変化する。これを誤差変化範囲(*serachRange*)と定義する。世代ごとに誤差ウェイトは以下のように変化する。

$$\text{errorWeight}_{t+1} = \text{errorWeight}_t + \text{searchWeight} \quad (9)$$

*serachWeight*は0を平均として誤差変化範囲における一様乱数で決定される。ただし、誤差ウェイトが $\text{predictBasement}^{-\text{errorRange}}$ よりも小さい場合は $\text{predictBasement}^{-\text{errorRange}}$ に置き換える。また、 $\text{predictBasement}^{\text{errorRange}}$ より大きい場合は $\text{predictBasement}^{\text{errorRange}}$ に置き換える。これにより、ロジスティック関数を用いた予測は商品が市場に出てから時間が経てばより容易になる様¹⁰⁾を表現する。

上記のように比較対象の値を予測したのち、第二の破壊的イノベーションの技術属性値を分子とした上で、施策IおよびIIIでは消費者の要求属性値の平均を、施策IIでは第一の破壊的イノベーションの技術属性値を分母とする割合を算出する。この値が一定の値よりも大きくなったかで移行を判断する。この一定の値とは、市場のどの顧客を重要視するかによって異なり、「a.ハイエンド重視(90%)」「b.ミドルレンジ重視(85%)」「c.ローエンド重視(80%)」の3つの顧客選択施策として設定する。例えば、施策III-aでは2年後に第二のイノベーションの技術属性値が消費者の要求技術属性値の平均の90%にまで迫ることが見込まれた時、第二のイノベーションに移行するという対策を実行する。

第二のイノベーションへの移行を決定した後は6~12世代の範囲内の一様乱数に従った世代分、技術開発を取りやめる(購買活動は引き続き行われる)。その後、第二のイノベーションの技術属性値を踏襲して「通常のアプローチ」を再開する。この範囲はChristensen および Foster の述べる具体例に準拠している¹¹⁾¹⁰⁾。

4.2 妥当性検証の方法と結果

本モデルが研究目的を満たすために妥当なモデルかを以下のステップにて検証を実施する。

① 性能進化の実データ, シェアデータの調査

入力データとしてシェアデータと性能の実データを調査する. 性能に関してはこの市場において重要視されていた「CPU」「メモリ」「価格」「重量」のデータを抽出する. なお, CPU は Pass-Mark 社による評価指標である「CPU Mark」¹⁴⁾を調査し, その値の大小を性能とした. モデルにおいてはこれら4つの要素を「技術属性」と表現する. なお, 「主要性能」が「CPU」「メモリ」に該当し, 「副次性能」が「価格」「重量」に該当する. モデルにおいても4つの技術属性が導入され, それぞれ技術属性値が算出される.

② ロジスティック関数に近似

非線形最小二乗法を用い, 性能進化の実データから入力用の技術関数であるロジスティック関数の係数(式(4)における a, K, t_0, C)を決定する. Fig. 5は「メモリ」における実データと近似を行なった技術関数である. 同様に「CPU」, 「価格」および「重量」において技術関数の係数を決定した.

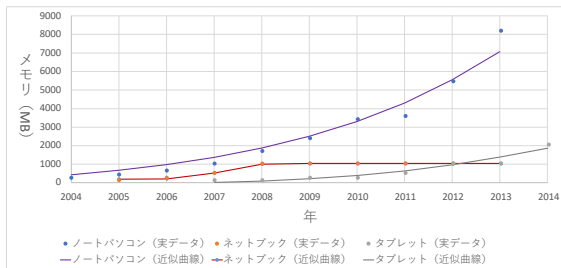


Fig. 5: メモリの実データと技術関数

軽量パソコン市場の事例においては4つの技術属性に3つの企業エージェントが存在するため, 計12種の技術関数が決定されることになる.

③ シミュレーション実験の実施

Table 1: シミュレーション実験の初期値

変数内容	初期値
世代数	36 ⁷⁾
総技術属性数	4 ⁷⁾
第二のイノベーターの参入世代	11 ⁷⁾
技術属性ウェイト	Table 2 参照
消費者総数	100 ²⁾³⁾
試行回数	1000 ²⁾³⁾
GA ウェイト	(0.5, 0.5) ²⁾³⁾
GA 交叉確率	0.6 ²⁾³⁾
GA 突然変異確率	0.01 ²⁾³⁾
更新ウェイト	(0.5, 0.5) ²⁾³⁾
効用ウェイト	0.13
要求属性向上割合 (主要性能)	9.8
要求属性向上割合 (副次性能)	2.7
カットオフ値	2.39
要求属性値初期分散	4.0

Table 2: 技術属性ウェイトの初期値⁷⁾

	CPU	メモリ	価格	重量
既存企業	1.2	1.2	0.8	0.8
第一の破壊的イノベーター	0.8	0.8	1.2	1.2
第二の破壊的イノベーター	1.0	1.0	0.8	1.2

シミュレーションを実施し, 各世代のシェアデータと技術属性値のデータを出力する. ②で決定した技術関数以外のパラメータは以下のように初期値を設定する. Table 1における「効用ウェイト」から「要求属性値初期分散」は実証研究や先行研究からは決定できないため, calibration によって決定する.

効用ウェイトから要求属性値初期分散に関してパラメータを変化させ, 感度分析を行った結果 Fig. 6のように推移した. 技術関数の係数を変化させるとシェア推移は大きく異なるため(→5.2) 一方, これらのパラメータはシェア推移の振る舞いに関して重大な変化をもたらしているとは言えない. よって calibration にて再現すべきデータに最も即したパラメータの組み合わせを探索した結果, Table 1の組み合わせとなった.

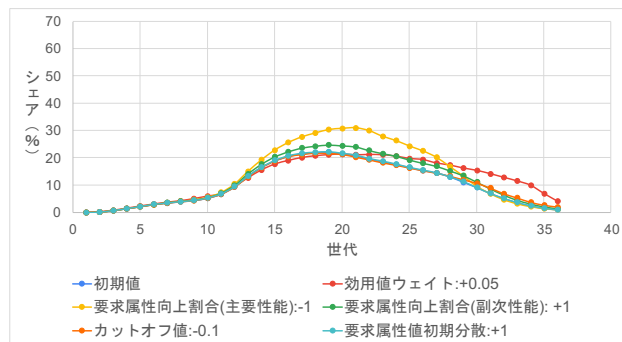


Fig. 6: シミュレーション実験における感度分析 (第一のイノベーターのシェアの平均推移)

④ 「軽量パソコン市場」の事例における「累次的破壊」に関わる Stylized Fact を検証

累次的破壊の基本原理に関わる定型化された事実である「Stylized Fact」を満たすかどうかを検証する. 累次的破壊における定型化された事実とは, 累次的破壊の定義および累次的破壊の直接的な原因である性能進化の組み合わせを指す. 満たされていた場合はモデルが妥当であると結論づけることができる.

④における Stylized Fact の観点は「累次的破壊の定義を満たすシェア推移・技術属性値の推移を再現できるか」と「バリューネットワークの影響を受けた技術属性値も Foster の S カーブ理論に則った推移をしているかどうか」である. これらの観点に則り, 以下の6つの観点を本シミュレーションに適応させた Stylized Fact

として設定する. なお, 4 以外は全て世代(位置)が実データとの誤差が一定以内に収まるかどうかで判定する.

1. 第二の破壊的イノベーションが既存企業のシェア平均値を追い抜く世代
2. 第一の破壊的イノベーションのシェア平均値が最大値をとる世代
3. 第二の破壊的イノベーションが第一の破壊的イノベーションのシェア平均を追い抜く世代
4. 各技術属性値の平均の推移が単調増加である
5. 各技術属性値の平均の推移の変曲点位置
6. 各技術属性値の平均の推移の企業同士の交点位置

4.3 妥当性検証の結果

本研究では先行研究や実証研究により変数の初期値を決定し, それ以外は calibration によって決定する. Fig. 2 の実際に得られたシェアデータに対し, Fig. 7 のようなシミュレーション結果が得られた. これは Stylized Fact 1,2,3 を満たすことが確認された.

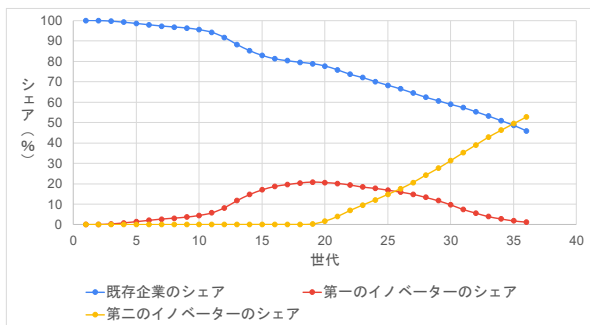


Fig. 7: 軽量パソコンの事例を用いたシミュレーション結果 (シェアデータの平均値の推移)

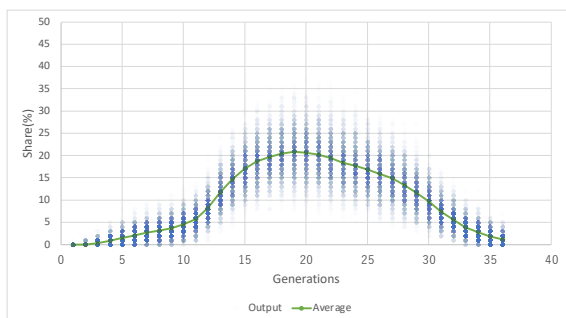


Fig. 8: 軽量パソコンの事例における第一の破壊的イノベーターのシェア推移 (平均値と全試行のばらつき)

また, Fig. 9 はメモリに当たる技術属性値 2 の各企業エージェントと消費者の要求属性値の平均の推移である. これらの出力結果と Fig. 9 に描写された技術関数の係数をもとに, Stylized Fact 4,5,6 を検証し, 満たすことを確認した.

同様に他の技術属性値の推移も Stylized Fact 4,5,6 を満たした.

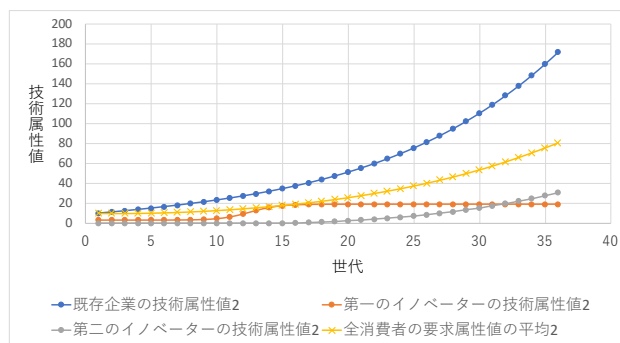


Fig. 9: 軽量パソコンの事例を用いたシミュレーション結果 ('メモリ'の技術属性値の平均の推移)

5 シナリオ分析

5.1 シナリオ分析の方針

シナリオ分析ではどのような性能進化の組み合わせの状況において累次的破壊が発生するのかを「状況シナリオ」を作成し分析する. そして, Christensen および Foster に共通する「新たなイノベーションへの完全移行」を実施するタイミングをどのように測るべきかについて「施策シナリオ」を用いて分析する.

5.2 状況シナリオ分析

状況シナリオ分析では, 「基準状況シナリオ」として Christensen の線形モデルに準じた状況を再現するシナリオを作成する. その上で, 性能進化の観点において累次的破壊の発生につながる以下の 6 つの要因を組み合わせることによって, どの組み合わせの「状況シナリオ」の際に累次的破壊が発生するかを検証する. なお状況シナリオの名前は採用する要因を組み合わせで呼称する. 例えば 1 と 3 を適用した状況シナリオは「状況シナリオ 13」と命名する.

1. 既存製品の成長がより加速していく
2. 第一のイノベーションの性能が頭打ちになる
3. 第一のイノベーションの性能が比較的低い
4. 第二のイノベーションの性能が比較的高い
5. 第二のイノベーションの性能が急速に進化する
6. 第二のイノベーションの進出がより早い

これらの要因ごとに基準状況シナリオの技術関数の係数の初期値を一定値変化させる. この変化量の妥当性は, 研究背景において述べた 3 つの累次的破壊の事例のシェアデータの概形を満たすかで検証を行った.

上記のように要因を組み合わせることにより技術関数の概形を変化させる. 妥当性検証では 1 通りの技術関数の組み合わせのみでシミュレーション実験を行なったが, 状況シナリオ分析では様々な組み合わせの状況を実験し, 累次的破壊が発生する状況はどのようなものがあるか, また発生するものではどのような判断

をするべきかを検証する。

上記の組み合わせによって累次的破壊の定義を全て満たす状況シナリオは 63 種のうち 30 種存在した。また、30 種のうち 28 種は Downes らの述べていた特徴である 5 または 6 のいずれかの要因を含んでいた。これらの状況シナリオと基準状況シナリオを用いて施策シナリオ分析を実施する。

5.3 施策シナリオ分析

施策シナリオの効果は第二のイノベーターが進出した世代から最終世代までの第一のイノベーターの平均シェアを算出し、施策なしの場合からどれほどの割合で平均シェアが改善したかで評価する。この評価方法によって移行後に回復したシェアだけでなく、移行しなければ得られたはずのシェアの機会損失分も考慮することができる。

施策シナリオにおける変数は実証研究だけでは決定できないため、以下のように初期値と範囲を決定し、感度分析を行って初期値が異なっても影響がないかを検証し、決定する。

Table 3: 施策シナリオ分析における初期値の感度分析

変数名	変数記号	初期値	変化量
誤差減衰値	<i>errorAttenuation</i>	0.03	±0.01
誤差範囲の基底	<i>predictBasement</i>	4.0	±1.0
誤差変化範囲	<i>serachRange</i>	0.2	±0.1
消費者の誤差範囲	<i>consumerPredictErrorRange</i>	0.2	±0.1

Table 3 に従い感度分析を行ったところ、Fig.10 のような結果になった。このように、ほとんど変化がないため、設定した初期値において問題ないと言える。

上記および Table1, Table2 のパラメータセットを用い、施策シナリオ分析を状況シナリオ分析において累次的破壊が発生したシナリオに対して実施する。

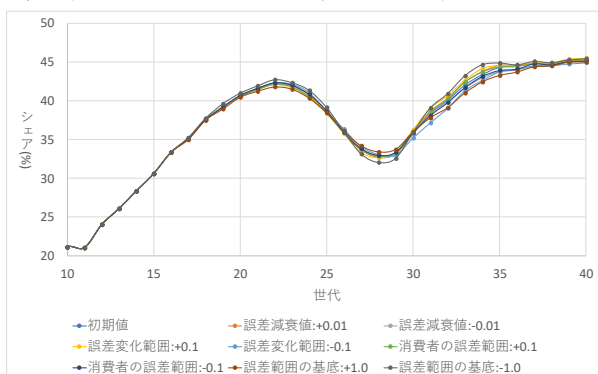


Fig. 10: 施策シナリオ分析における感度分析の結果 (第一のイノベーターのシェアの平均推移)

施策シナリオ分析の結果は Table 4 のようになった。なお、「相対有効度が 70%以上であった回数」とは、各状況シナリオにおいて最も優れている判断基準(最優秀判断基準)の改善率の値に対し、その判断基準の改善率の値の割合(これを相対有効度と定義する)が 70%以上で合った場合にカウントする。つまり最も優秀だったわけではないが、基準としての効果に大きな違いはなく十分効果があったと言えるものである。

Table 4: ミドルレンジ重視における各判断方法の優劣

	Christensen	Foster	混合
最優秀判断基準であった回数 (+基準状況)	2	14	13
相対有効度が 70%以上であった回数 (+基準状況)	2	25	26

Table 4 のように累次的破壊が発生する全 30 種の状況下においては Christensen の基準では効果がほとんどないことがわかった。

これらの施策シナリオを詳細に分析するにあたって、30 種の状況シナリオは以下の 4 つのタイプに分けることができる。

- A) Foster の基準と混合基準に一定の効果がある状況
- B) Christensen の基準と混合基準に一定の効果がある状況
- C) Foster の基準に大きな効果がある状況
- D) 施策に効果がない状況

タイプ A は B~D に該当しない 23 種の状況シナリオである。タイプ A の状況シナリオは Foster または混合基準が最優秀判断基準となり、もう片方の基準も 70%以上の相対有効度を有する。つまりタイプ A に該当する状況シナリオでは Foster および混合基準に一定の効果があるといえる。

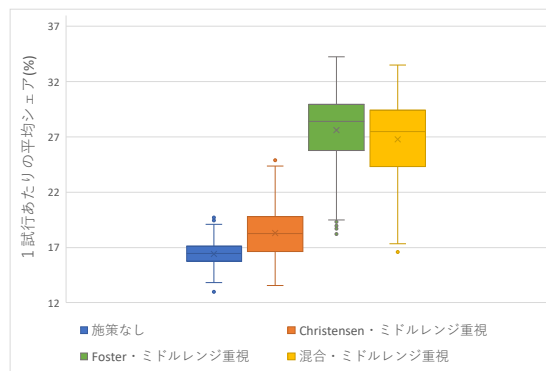


Fig. 11: 状況シナリオ 3456 ミドルレンジ重視におけるシェア平均の分布

Fig. 11 は状況シナリオ 3456 のミドルレンジ重視の場合の性能進化における判断基準ごとの平均シェアの分布である。移行そのものを行わない「施策なし」に

対し、Christensen の判断基準はほとんど平均シェアが変わらず、効果が少ないことが分かる。一方で Foster の判断基準や混合基準は平均や分布の上限も高く、効果があることが分かる。

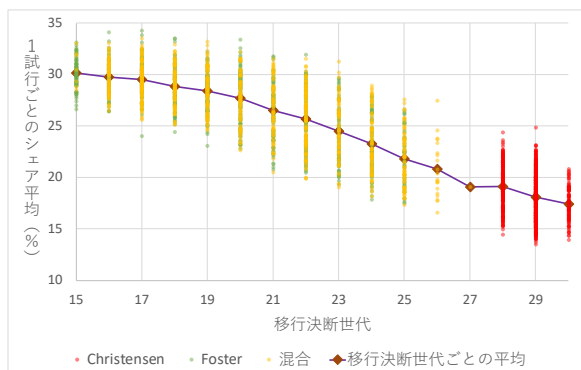


Fig. 12: 状況シナリオ 3456 ミドルレンジ重視における移行決断世代とシェア平均の分布

Fig. 12 は移行決断世代、つまり各判断基準を用いた結果移行段階に入らなくてはならないと決断した世代ごとの平均シェアの分布と、その平均の折れ線グラフである。この折れ線は各移行決断世代における判断基準を問わない平均値である。移行決断世代から第一のイノベーターは移行準備期間に入る。Fig. 12 のように、Christensen は移行決断世代が遅くなっていることがシェアを獲得できない理由であることが分かる。一方で混合基準は移行決断世代がかなり分散しており、これが Fig. 11 における分散の高さの原因と言えらる。Foster や混合基準の分散が大きいのは S カーブを予測することが線形推移を予測するよりも難しいからであり (Foster 自身も予測の困難性について言及している¹⁰⁾)、かつ混合基準は Foster の基準と違い、第一のイノベーターの性能という予測者自身の推移ではなく、消費者の平均と第二のイノベーションの性能という他者(社) 同士の予測する必要がある、その分誤差が生じやすいからであるといえる。

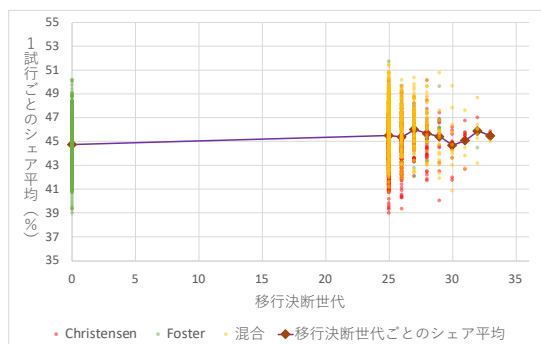


Fig. 13: 状況シナリオ 24 ミドルレンジ重視における移行決断世代とシェア平均の分布

タイプ B は状況シナリオ 24, 状況シナリオ 234, 状

況シナリオ 6, 状況シナリオ 236 が該当する。A と異なり Christensen の判断基準に効果があったものである。タイプ B は混合基準も効果が見られた。

Fig.13 は状況シナリオ 24・ミドルレンジ重視の移行決断世代ごとの平均シェアの分布である。ここで移行決断世代が 0 になるものは、移行決断ができず、移行期間にうつることができなかった試行である。Fig.13 のように Foster の判断基準では移行決断ができなかったことが「施策なし」に比べて効果がなかった結果であるといえる。

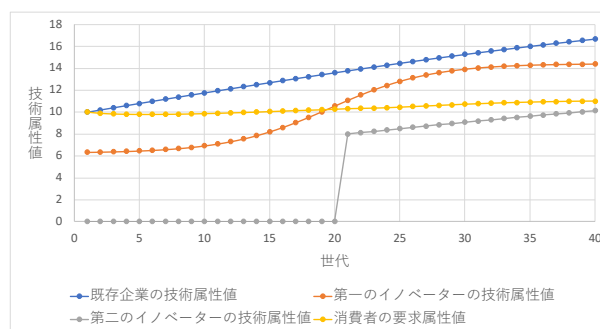


Fig. 14: 状況シナリオ 24 施策なしにおける技術属性値と要求属性値の平均の推移

Fig.14 は状況シナリオ 24・施策なしの場合の技術属性値と消費者エージェントの平均要求属性値の推移である。第二のイノベーションの性能進化が線形であり、かつ第一の破壊的イノベーターは消費者の要求属性値に対し性能過剰を起こしてしまっていることが分かる。このように状況シナリオ 24 では累次的破壊は発生しているものの、Christensen らが従来唱えていた破壊現象と同等の状況であるため、イノベーション同士で比べてしまう Foster の判断基準では効果がなく、消費者と第二のイノベーションで比べる Christensen に効果があるといえる。実際、タイプ B に該当する状況シナリオいずれにも要因 5 の第二のイノベーションの急成長性は含まれていない。なお、混合基準は第二の破壊的イノベーターをロジスティック関数、つまり急成長すると予測したものの、消費者と比べているため移行決断ができ、ある程度の効果があったといえる。しかし、B に該当する状況シナリオは 30 種のうち、4 種と少なく、要因 5 を欠いた状態で累次的破壊が起こるには第一の破壊的イノベーションが性能過剰を起こすものの、トップシェアを維持し続けることができないという条件が重ならなくてはならないことを考えると、比較的発生しにくい状況であるといえる。

タイプ C は状況シナリオ 16,1356 が該当するもので、Foster の判断基準のみ効果があるものである。Fig. 15 のように混合基準のシェアが下に間延びしてしまい、1000 試行における平均が下がってしまう状況である。

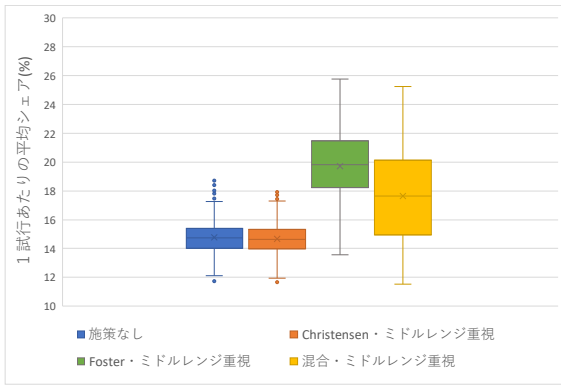


Fig. 15: 状況シナリオ 1356 ミドルレンジ重視におけるシェア平均の分布

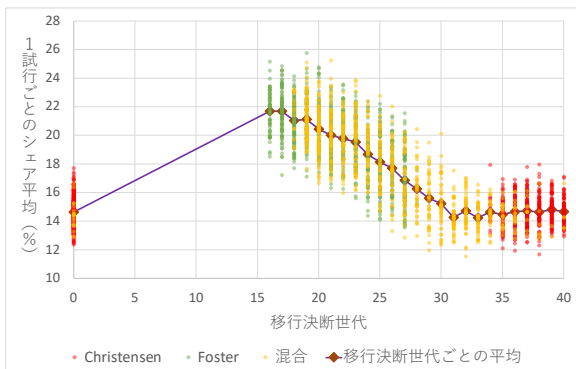


Fig. 16: 状況シナリオ 1356 ミドルレンジ重視における移行決断世代とシェア平均の分布

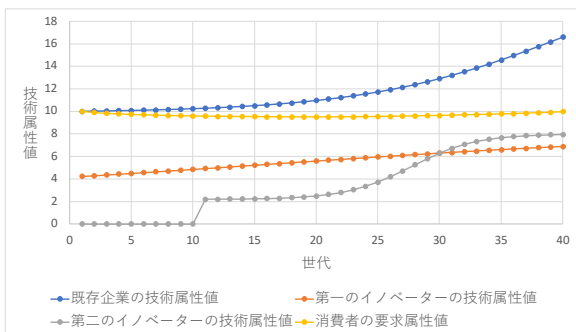


Fig. 17: 状況シナリオ 1356 ミドルレンジ重視における移行決断世代とシェア平均の分布

Fig. 16 のように Christensen の基準では移行決断ができない、または大きく遅れているだけでなく、混合基準も Foster の基準に比べて決断が遅れ、かつ移行決断世代がばらついていることがわかる。状況シナリオ 1356 では移行が遅れば遅れるほど獲得シェアが下がってしまっていることを考えると、混合基準は移行決断が全体的に遅れ、かつブレも大きいことも効果がない原因であると言える。

Fig. 17 のように状況シナリオ 1356 では消費者の要求属性値に追いつくことなくシェアの逆転が発生している。そのため、消費者の要求属性値と比較する Christensen や混合基準では効果が少ないのである。ただし、第二の破壊的イノベーションをより脅威だと認識し、過大評価してしまった試行では、消費者の求め

る性能にかなり近づくと予測するため、結果的に適切なタイミングで移行することができる。一方で Foster の基準においても第二の破壊的イノベーションを過小評価してしまうと、Fig. 15 の緑色のバーの最小値が施策なしの平均と同じくらいになってしまうように、失敗してしまう試行も存在する。

タイプ C のような状況では Foster の基準に効果が大きい、混合基準も失敗する確率も高いものの成功する試行も Fig. 15 のように少なくはないこと、また、そもそもこのように消費者の求める性能に追いつかない状況にも関わらずシェアが逆転することは 30 種のうち 2 種と珍しいことを考えると、判断基準は必ず Foster の基準であるべきとは言えない。

タイプ D はどの基準も効果がなかった状況である。Fig. 18 はタイプ D に唯一該当する状況シナリオ 1256・ミドルレンジ重視の移行決断世代別の平均シェアの分布である。今までの事例と異なり、移行決断が早すぎるとかえって平均シェアが低いことがわかる。特に Foster や混合基準は早い段階で移行決断してしまっており、「施策なし」に該当する移行決断世代 0 の試行に比べて平均シェアが小さい。

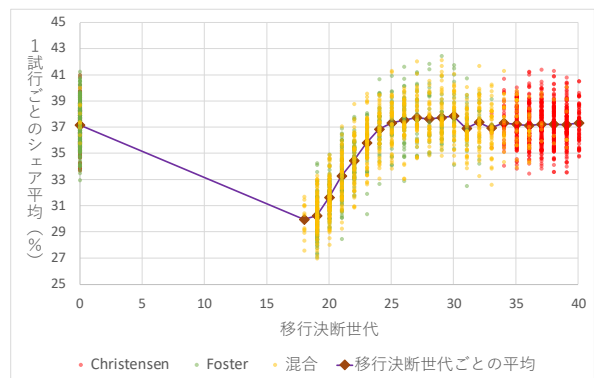


Fig. 18: 状況シナリオ 1256 ミドルレンジ重視における移行決断世代とシェア平均の分布

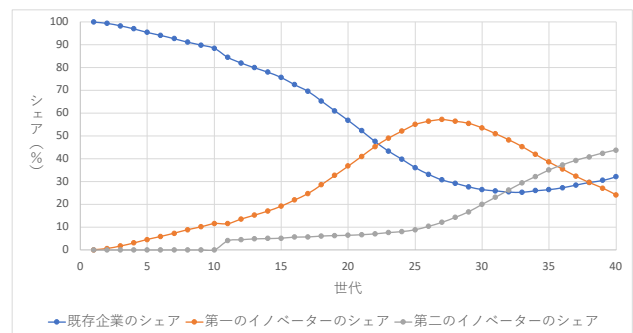


Fig. 19: 状況シナリオ 1256・施策なしにおける各企業エージェントの獲得シェアの平均推移

Fig. 19 は状況シナリオ 1256・施策なしにおける各企業エージェントの獲得シェア平均のグラフである。第一のイノベーターのシェアは第 30 世代頃からシェア

の下降が始まっている。そのため、第 25 世代未満で移行決断をしてしまうと、6~12 世代の移行期間を経ても第 24~30 世代で移行を完了してしまうと、この頃に第一のイノベーションを用いて獲得できたはずのシェアを獲得できないという機会損失が発生してしまう。

ただしこのような状況もたった 1 つの組み合わせと起こりにくく、唯一イノベーターが恐れるシナリオであるものの、やはり他の状況シナリオに該当する可能性の方が圧倒的に高いことを考えると、Foster や混合の判断基準に効果がないとは言えない。

これらの 4 つのタイプごとの分析のように、累次的破壊が発生する状況では Christensen の判断基準は効果が少ないことがわかった。Foster の判断基準は多くの状況で効果があるものの、第二の破壊的イノベーションが緩やかに進化するシナリオでは効果があまりなかった。混合基準ではばらつきが大きいという欠点があるものの、比較的多くのシナリオで一定の効果があり、最も頑強性がある基準と言える。

Table 5: ミドルレンジ重視における各判断方法の優劣

	ハイエンド重視	ミドルレンジ重視	ローエンド重視
Christensen	3	1	19
Foster	1	2	27
混合	2	1	27

また、「顧客選択における判断基準」も分析した結果、Table 5 のように 30 種の累次的破壊の状況下において Christensen の基準では 19 種が、Foster の基準と混合基準では 27 種が「ローエンド重視」施策が最もシェアを獲得することができた。Christensen は従来理論において破壊的イノベーションに対してはよりローエンドを重視すべきだと述べており、これが累次的破壊においても同様であることが今回わかった。

6 結論

本研究では従来の性能進化の線形仮定では説明のつかない「累次的破壊」の定義・事例調査を行い、Sカーブ理論に基づいたモデル化を行った。そのモデルによって特に Downes らが言及した後からやってくるイノベーションの急速な成長性と早期の出現が特に累次的破壊が発生する要因になってくることがわかった。また、第一の破壊的イノベーターの施策として Christensen の基準では効果が少なく、Christensen と Foster の判断基準を混合させた基準に関して、ばらつきが大きいという欠点はあるものの状況に左右されにくく頑強性が高いことがわかった。また、顧客選択施策としてはよりローエンドを重視するべきであることがわかった。

参考文献

- 1) Clayton M. Christensen, “The innovator’s dilemma: when new technologies cause great firms to fail”, Harvard Business School Press (1997) (伊豆原弓訳, 玉田俊平太監修, 「イノベーションのジレンマ—技術革新が巨大企業を滅ぼすとき— 増補改訂版」, 翔泳社 (2001))
- 2) 山本智也, 高橋真吾, 「イノベーションのジレンマの原因に関する仮説の検証と優良企業が取るべき対策の考察」 経営情報学会全国研究発表大会要旨集 **2014f(0)**, 89/92 (2014)
- 3) 矢野勇暉, 高橋真吾, 「ローエンド破壊と新市場破壊発生時における優良企業の対策方法の分析」, 経営情報学会誌, **24-4**, 255/262 (2016)
- 4) 総務省, 「平成 26 年版情報通信白書 第 1 部第 2 節 ICT 産業構造のパラダイムシフト」, <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h26/html/nc122130.html>, 最終閲覧日: 2023/01/20
- 5) Neil Parmar, “Tablets poisonous for netbooks market”, The-NationalNews, 2013, <https://www.thenationalnews.com/business/tablets-poisonous-for-netbooks-market-1.308428>, 最終閲覧日: 2023/01/20
- 6) John Paczkowski, “The Netbook Is on Its Last Legs”, All-ThingsD, 2013, <https://allthingsd.com/20130415/the-netbooks-on-its-last-legs/>, 最終閲覧日: 2023/01/20
- 7) 藤本雄一郎, 破壊的イノベーション: 市場の構造変化の見極めと対処方法, 中央経済社 (2013)
- 8) Rokon Zaman, “Netflix Disruptive Innovation – renting to streaming”, THE WAVES Technology, Society and Policy, <https://www.the-waves.org/2022/03/15/netflix-disruptive-innovation-renting-to-streaming/>, 最終閲覧日: 2023/01/14
- 9) Maxwell Wessel, “Why Spotify Will Kill iTunes”, Harvard Business Review, <https://hbr.org/2011/07/why-spotify-will-kill-itunes>, 最終閲覧日: 2023/01/14
- 10) Richard N. Foster, “Innovation: The Attacker’s Advantage”, Summit Books (1986) (大前研一訳, 「イノベーション—限界突破の経営戦略」, TBS ブリタニカ, (1987))
- 11) Clayton M. Christensen, Rory McDonald, Elizabeth J. Altman, Jonathan E. Palmer, “Disruptive Innovation: An Intellectual History and Directions for Future Research” Special Issue: Managing in the Age of Disruptions, **55-7**, 1043/1078 (2018)
- 12) Larry Downes, Paul Nunes, “Big Bang Disruption: Strategy in the Age of Devastating Innovation”, Portfolio (2014) (江口 泰子 訳, 「ビッグバン・イノベーション -一夜にして爆発的成長から衰退に転じる超破壊の変化から生き延びよ」, ダイヤモンド社 (2016))
- 13) Kotaro Otori, Shingo Takahashi, “Market Design for Standardization Problems with Agent-based Social Simulation”, Journal of Evolutionary Economics, **22 - 1**, 49/77 (2012)
- 14) PASSMARK Software, “CPU Benchmark Year on Year Performance”, <https://www.cpubenchmark.net/year-on-year.html>, 最終閲覧日: 2023/01/27