

平成 25 年度
修士論文
汎用的なウェブページ最適化手法の提案

平成 26 年 3 月
指導教員 松尾豊准教授

東京大学大学院工学系研究科技術経営戦略学専攻
37-126847 飯塚修平

概要

スマートフォンや E コマースの普及に後押しされてウェブ関連の市場規模は成長を続けており、シリコンバレーを中心に世界中のスタートアップによって日々新たなウェブサービスが生み出されている。スタートアップは新たなビジネスモデルを世に問う起業活動であり、ユーザのニーズが不確実な状況でサービスを構築して成長させことが多い。不確実性が大きい中で着実にサービスを成長させるための手法として注目されているのが、データに基いてユーザのニーズを把握するウェブページ最適化である。

ウェブページ最適化では一部の機能やデザインを変更したパターンを用意し、それらにランダムにユーザを振り分ける。そしてユーザの反応を計測して比較を行い、最もパフォーマンスの高いパターンを探索する。低いコストで実践できて高い効果を得ることができる手法である。しかし対象となるウェブサイトの訪問者数が少ない場合は、パターンを評価するのに十分なデータが集まるまでに時間がかかってしまうという問題がある。

そこで本研究では、ウェブサイトの訪問者数およびテストする要素の数に応じて最適なパターンを探索することができる汎用的な最適化手法を提案した。まずウェブページ最適化を組み合わせ最適化問題のひとつとして捉え、ウェブページ最適化問題を定式化した。その後ウェブサイトの訪問者数およびテストする要素の数に応じて最適化を行うアルゴリズムを提案した。さらにシミュレーションおよび実際のウェブサイトへの導入実験を行い、提案アルゴリズムの有効性を評価した。

実験の結果、提案手法によって訪問者数が少ない場合でも多い場合でも既存手法と同程度またはそれ以上の正確度で最適なパターンを探索することができることを示した。また、実際のウェブサイトに提案アルゴリズムを実装したウェブページ最適化プログラムを導入することによって、提案アルゴリズムが機能することを示した。

本研究で提案した最適化手法によって、ウェブサイトの規模に関わらずデータにもとづいた改善を行うことができるようになる。また、ウェブ以外の分野の問題であってもウェブページ最適化問題の形に落としこむことができれば、提案手法を用いて最適化を行うことができる。

目次

概要	i
第1章 序論	1
1.1 研究の背景	1
1.1.1 スタートアップと仮説検証型ウェブサービス開発	1
1.1.2 仮説検証型ウェブサービス開発におけるウェブページ最適化	2
1.1.3 ウェブページ最適化の問題点	4
1.2 研究の目的	4
1.2.1 ウェブページ最適化問題の定式化	5
1.2.2 汎用的なウェブページ最適化手法の提案と評価	5
1.3 本論文の構成	5
第2章 関連研究	6
2.1 評価指標の設定にまつわる研究	6
2.2 仮説パターンの生成にまつわる研究	7
2.2.1 ウェブページの自動解析	8
2.2.2 ウェブページ要素の重要度推定	10
2.2.3 ウェブページの自動生成	10
2.3 探索手法にまつわる研究	10
2.3.1 既存のウェブページ最適化手法	11
2.3.2 組合せ最適化問題に対する探索手法	14
2.3.3 効率的な探索の工夫	16
2.4 最適化プログラムの実装方法	16
2.4.1 プロキシサーバ方式	16

2.4.2 クライアントサイド方式	17
2.4.3 サーバサイド方式	17
2.5 ウェブ特有の実験上の注意点	18
2.5.1 ロボット	18
2.5.2 目新しさ効果	18
2.5.3 ユーザ体験の一貫性	18
2.5.4 時間要因	19
第3章 ウェブページ最適化問題	20
3.1 問題の定式化	20
3.2 局所探索法を用いた解法	21
3.2.1 局所探索法の概要	22
3.2.2 近傍解の生成	23
3.2.3 暫定解の移動	23
3.3 推測統計学の活用	23
3.3.1 仮説検定	23
3.3.2 区間推定	24
3.3.3 解の評価に用いるサンプルサイズ	25
3.4 線形モデルによる最適解の予測	26
3.5 既存のウェブページ最適化手法の整理	26
3.5.1 評価指標および検定方法の設定	26
3.5.2 既存のウェブページ最適化手法の位置づけ	27
3.6 本章のまとめ	28
第4章 提案手法	29
4.1 提案手法の概要	29
4.1.1 戦略判断フェーズ	30
4.1.2 初期解決定フェーズ	31
4.1.3 解探索フェーズ	32
4.2 観測データの活用	32
4.3 アルゴリズム	33
4.3.1 事前準備	33

4.3.2	ベースラインアルゴリズム A:単純総当たり	34
4.3.3	ベースラインアルゴリズム B:単純山登り	35
4.3.4	提案アルゴリズム A: 観測データを活用した初期解決定	36
4.3.5	提案アルゴリズム B: 観測の短縮の導入	36
4.3.6	提案アルゴリズム C: 敗者解のカットオフの導入	38
4.4	本章のまとめ	38
第 5 章	実験	41
5.1	仮想の評価関数によるシミュレーション実験	41
5.1.1	問題設定	41
5.1.2	実験結果	43
5.1.3	まとめ	48
5.2	実際のウェブサイトで得られたデータによるシミュレーション実験	49
5.2.1	対象サイト	49
5.2.2	問題設定	50
5.2.3	実験結果	52
5.3	実際のウェブサイトにおける実験	53
5.3.1	実験内容	53
5.3.2	実験装置	56
5.3.3	実験結果	57
第 6 章	考察	61
6.1	提案アルゴリズムの適用可能性	61
6.1.1	限界観測数 N が非常に小さい場合	61
6.1.2	解空間 X が非常に大きい場合	62
6.1.3	時間によって評価関数が変化する場合	62
6.2	最適化問題における変数の抽出の重要性	63
6.3	実験に用いたウェブサイトの妥当性	64
6.4	ウェブページ最適化問題と実験計画法の関係	64
6.5	ウェブ以外の分野への応用可能性の検討	65
6.5.1	メディア製作への応用可能性	66
6.5.2	ものづくりへの応用可能性	66

第7章　まとめ	68
付録A　実サイトB仮説パターン例	70
謝辞	72
参考文献	73

図目次

1.1	リーンスタートアップで提案されている構築-計測-学習のフィードバックループ ([1] から引用)	2
1.2	バラック・オバマ氏公式ウェブサイトのトップページの A/B テストで用いられた写真とボタンのバリエーション ([2] から引用)	3
2.1	Optimizely での仮説パターン生成例 (対象は筆者のポートフォリオサイト)	7
2.2	Bricolage によるテンプレートとコンテンツの入れ替え	11
4.1	提案手法のコンセプト	30
5.1	問題 0 における提案アルゴリズム A の正確度	43
5.2	問題 1 における提案アルゴリズム A の正確度	44
5.3	問題 2 における提案アルゴリズム A の正確度	45
5.4	問題 3 における提案アルゴリズム A の正確度	46
5.5	問題 4 における提案アルゴリズム A と提案アルゴリズム B の正確度	47
5.6	問題 4 における提案アルゴリズム B と提案アルゴリズム C の正確度	48
5.7	実サイト A の相関図ページ	50
5.8	実サイト A から得た評価関数における各アルゴリズムの正確度	52
5.9	実サイト B の記事ページ例	54
5.10	ウェブページ最適化プログラムの全体像	56
5.11	観測回数と暫定解の期待値の推移	58
5.12	x_1 : 期待値の比較	60
5.13	x_2 : 期待値の比較	60
5.14	x_3 : 期待値の比較	60
5.15	x_4 : 期待値の比較	60

5.16	$x_1 \times x_2$: 期待値の比較	60
6.1	評価関数に非線形性が生じる変数の抽出例	63
A.1	仮説パターン $x = (0, 0, 0, 0)$	70
A.2	仮説パターン $x = (0, 0, 0, 1)$	70
A.3	仮説パターン $x = (1, 0, 0, 0)$	71
A.4	仮説パターン $x = (1, 2, 2, 1)$	71
A.5	仮説パターン $x = (0, 1, 0, 0)$	71
A.6	仮説パターン $x = (0, 0, 1, 0)$	71

表目次

3.1	評価指標の分類と対応する確率分布	26
3.2	既存のウェブページ最適化手法とウェブページ最適化問題における探索手法 と工夫の対応付け	27
4.1	提案アルゴリズムの整理	40
5.1	シミュレーションの問題設定	42
5.2	実サイト A における各施策の効果	51
5.3	実サイト B を構成する変数と値	55
5.4	実サイト B における実験結果	57
5.5	暫定解と期待値の推移	59

第1章

序論

1.1 研究の背景

1.1.1 スタートアップと仮説検証型ウェブサービス開発

近年のウェブの著しい発展によって、我々の生活にウェブが深く入り込むようになった。スマートフォンや E コマースの普及に後押しされてウェブ関連市場は成長を続けており、米国のシリコンバレーを中心に世界中で日々新たなウェブサービスが生み出されている。特にウェブサービスによって短期間で急激な成長を狙う起業活動のことをスタートアップと呼ぶ。

スタートアップは自分達のアイデアが受け入れられるかわからない不確実性の中でサービスを開発することが多い。綿密なマーケット調査やユーザヒアリングを行なったとしても、発言と実際の行動との間に乖離があることもあるため、実際にユーザにウェブサービスを利用してもらう前からユーザが求めているものを的確に把握することは難しい。

ユーザが求めていないサービスに無駄な時間と資金を投入することによるスタートアップの失敗を防ぐための方法論として生まれたのが、リーンスタートアップをはじめとする仮説検証型ウェブサービス開発である [3][4]。リーンスタートアップではスタートアップそのものを大きな実験として捉える。仮説となるビジネスのアイデアをもとに、ウェブサービスという製品を構築する。そしてその製品に対する実際のユーザの反応を計測して得たデータをもとに検証することで、ユーザが求めている機能や体験について学び、サービス構築に活用する（図 1.1 参照）。この方法論は 2008 年に Eric Ries がリーンスタートアップと称して自身のブログ^{*1}で提案し、その後同名の著書 [1] が出版されたことによって広く知られるようになった。

^{*1} Lessons Learned: The lean startup

<http://www.startuplessonslearned.com/2008/09/lean-startup.html>

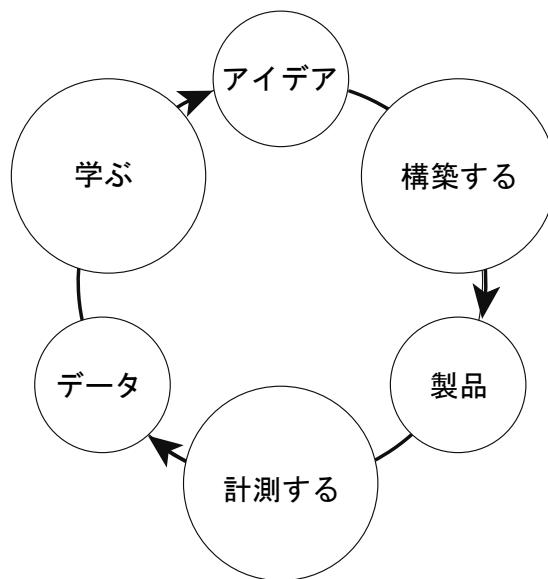


図 1.1 リーンスタートアップで提案されている構築-計測-学習のフィードバックループ ([1] から引用)

1.1.2 仮説検証型ウェブサービス開発におけるウェブページ最適化

仮説検証型ウェブサービス開発では、ユーザの反応を計測するためにウェブページ最適化とよばれる手法を用いる。ウェブページ最適化とはデザインや機能の一部が異なるウェブページ（仮説パターンと呼ぶ）を用意し、それらのパターンにユーザを振り分けて実際に見せた時の反応の違いを計測することで、広告のクリック率やページ滞在時間などの評価指標が高いパターンを発見する手法のことである。ウェブページ最適化を用いることで、ウェブサイト開発者はデータに基いてユーザが求めている機能や行動を促すデザインについて学ぶことができる。実験を通して得た知見に基いてウェブサイトに変更を施していくことで、着実にサービスを成長させることができる。

この手法は「サービスのお申し込みボタンをクリックする」、「メールアドレスを登録する」などの、サービスの収益につながるユーザの行動を喚起するウェブページのデザインを探求する目的で用いられることが多い。そのため、実験に基づいてウェブページを改善する手法はランディングページ^{*2}最適化 (LPO: Landing Page Optimization) という名でも親しまれている。複数のパターンの比較対照実験を行う手法を総じて A/B テストと呼ぶこともある。

ユーザにクリックを促すボタンの文言やページに掲載する写真素材など、デザイン上の細か

^{*2} ユーザが検索エンジンの検索結果や広告のリンクをクリックした際に最初に表示されるページ

な違いがユーザの行動を左右することが知られている。たとえば 2008 年アメリカ合衆国大統領選挙では、バラック・オバマ陣営がウェブページ最適化を活用して多額の献金を集めることに成功した [2]。当時バラック・オバマ氏の公式ウェブサイトには支援者からの献金を受け付ける機能が備わっていたが、献金を行うにはメールアドレスを登録する必要があり、訪問者のメールアドレス登録率が低いことがボトルネックになっていた。そこで、バラック・オバマ陣営はトップページで A/B テストを行い、訪問者のメールアドレス登録率を向上させるデザインを探索することにした。

実験では 6 種類の写真と 4 種類のボタンのバリエーションから 24 種類のパターンが生成され、約 30 万人もの訪問者に対してテストが行われた（図 1.2 参照）。その結果、最もパフォーマンスの高いパターンはオリジナルのパターンに対して 40.6% も登録率が高いことがわかった。このパターンを採用したところ、バラック・オバマ陣営は 2,880,000 のメールアドレスと 6000 万ドルの献金を追加で獲得することに成功した。

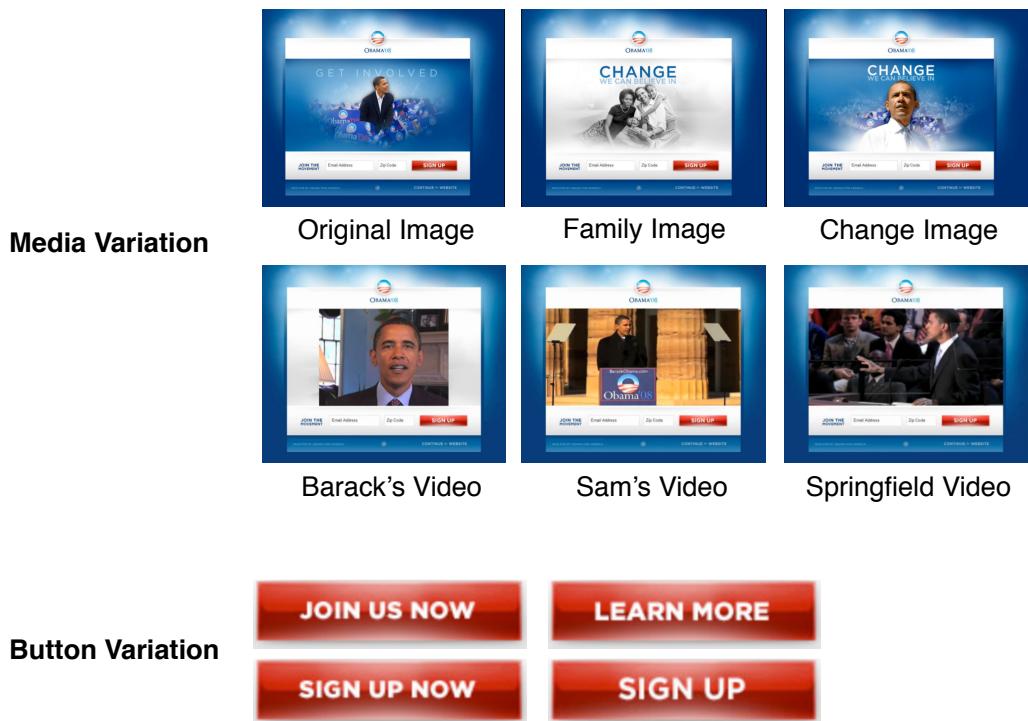


図 1.2 バラック・オバマ氏公式ウェブサイトのトップページの A/B テストで用いられた写真とボタンのバリエーション ([2] から引用)

この技術チームは、その後 2010 年に Optimizely^{*3}というウェブサービスを立ち上げており、ウェブエンジニアリングの知識がないウェブサイトオーナーでも簡単に A/B テストを行うことができるようにするためのプラットフォームを提供している。この他にも Google が提供する Google Website Optimizer^{*4}、Visual Website Optimizer^{*5}、日本国内で展開する planBCD^{*6}など国内外で様々なウェブページ最適化ツールが誕生している。

1.1.3 ウェブページ最適化の問題点

訪問者数が多いウェブサイトでは短期間にたくさんのデータを得ることができるため、早い段階で実験の結果を得ることができる。しかし訪問者数が少ないウェブサイトで同じ数のパターンを評価しようとすると、十分なデータが集まるまでに時間がかかってしまう。実験に時間がかかることはウェブサービスの成長を鈍化させるだけではなく、季節性による外的要因が混入する余地が大きくなるため実験上好ましくない。そのため、訪問者数が少ないウェブサイトは効率的にデータを収集・活用して最適化を行う必要がある。

1.2 研究の目的

そこで本研究では、ウェブページの訪問者数とテストするパターンの数に応じて汎用的に最適化を行うことができるウェブページ最適化手法を提案する。まずウェブページ最適化における実験の設計を「評価指標の設定」「仮説パターンの生成」「探索手法の決定」という 3 つのプロセスにわけて考え、それぞれにまつわる関連研究および手法を網羅的に説明する。その後、ウェブページ最適化と組み合わせ最適化問題の類似性に着目し、ウェブページ最適化問題を定式化する。局所探索法を用いた解法および効率的な探索のための工夫について整理した後、提案手法のアルゴリズムを説明する。最後にシミュレーションおよび実際のウェブサイトへの導入実験を通して、提案手法の有効性を評価する。

^{*3} Optimizely <https://www.optimizely.com/>

^{*4} Google Website Optimizer <http://www.google.com/analytics/>

現在は同社が提供するアクセス解析ツール Google Analytics のひとつの機能として統合されている。

^{*5} Visual Website Optimizer <http://visualwebsiteoptimizer.com/>

^{*6} planBCD <https://planb.cd>

1.2.1 ウェブページ最適化問題の定式化

ウェブページ最適化でテストする仮説パターンはウェブページを構成するボタンやラベルなどの要素の組み合わせによって構成されている。したがって、ウェブページ最適化は評価指標を最大化する要素の組み合わせを探しだす組合せ最適化問題として定式化することができる。本研究ではウェブページ最適化問題を組合せ最適化問題のひとつとして捉えて定式化を行い、局所探索法による解法を提示する。また、効率的な探索のために導入することができる工夫について整理する。

1.2.2 汎用的なウェブページ最適化手法の提案と評価

ウェブサイトによって訪問者数およびテストするパターンの数といった制約が異なるため、制約に応じて汎用的に最適化できることが望ましい。そこで本研究ではウェブサイトの訪問者数とテストするパターンの数に応じて最適化を行うウェブページ最適化手法を提案する。さらに様々な工夫を導入することで、効率的な探索を行うことできることを示す。

最後に、シミュレーションおよび実際のウェブサイトへの導入実験によって提案手法を評価する。シミュレーションでは筆者が与えた仮想的な評価関数と、実際のウェブサイトにて観測されたデータをもとに設定した評価関数の2つのケースで評価を行う。実際のウェブサイトへの導入実験では、提案手法のアルゴリズムにもとづいて実装したウェブページ最適化プログラムを実験対象となるウェブサイトに組み込み、実際に最適化を行うことで有効性を評価する。

1.3 本論文の構成

本論文の構成は以下のとおりである。第2章では「評価指標の設定」「仮説パターンの生成」「探索手法の決定」という3つのプロセスにわけてウェブページ最適化の関連研究を説明する。第3章ではウェブページ最適化問題の定式化を行う。第4章ではウェブサイトの訪問者数とテストするパターンの数に応じてウェブページ最適化を行うことができるアルゴリズムを提案する。第5章ではシミュレーションおよび実際のウェブサイトへの導入実験について説明する。第6章では第3章から第5章までの考察を行い、最後の第7章を本論文のまとめとする。

第2章

関連研究

本章ではウェブページ最適化の関連研究を説明する。2.1節から2.3節では、ウェブページ最適化を順番に「評価指標の設定」「仮説パターンの生成」「探索手法の決定」の3つのプロセスに分けて考え、それぞれのプロセスに関連する研究を説明する。仮説パターンとはデザインや機能がオリジナルと一部異なるウェブページのことである。ウェブページ最適化では、この仮説パターンの中から広告クリック率やページ滞在時間といった評価指標を最大化するパターンを探しだす。2.4節ではウェブページ最適化プログラムの構築方法について説明する。2.5節では実験を行う上で注意したいウェブ特有の外的要因について説明する。

2.1 評価指標の設定にまつわる研究

ウェブページ最適化では一般的な比較対照実験と同様に、統計学的仮説検定および区間推定を用いて仮説パターン間の優劣を評価する事が多い。設定する評価指標によって検定手法および必要なサンプルサイズが異なるため、予め適切な評価指標を設定することはウェブページ最適化を用いた意思決定をする上で重要である[5]。

設定した検定手法が正しくパターン間の差異を捉えられることを確かめるために、ウェブページ最適化を行う前にA/Aテストを行うことが望ましい。A/Aテストでは全く同じパターンを2つ用意し、それぞれにランダムにユーザを割り振ってデータを収集する。有意水準5%で検定を行う場合は、A/Aテストで得られた2群のデータの間に差が認められる確率は約5%になるはずである。5%を大きく超える確率で差が認められる場合は第一種過誤が大きすぎるため、設定した検定方法が適切でないと考えられる[6]。

評価指標の性質を顧みて当てはまりの良い確率分布を求める研究も行われている。たとえばサイト内における各ユーザの総クリック数を評価指標とする場合、すべてのユーザがある一定

の確率でクリックを行うと仮定するポアソン分布よりも、ユーザによってクリックする確率が違うことをモデルに含めることができる負の二項分布のほうが当てはまりが良いことが示されている [7]。

2.2 仮説パターンの生成にまつわる研究

最適化の対象となるウェブページと評価指標を設定した後は、評価指標を向上させるために考えられる仮説を列挙し、それぞれをユーザに表示する仮説パターンとして実装する。現在提供されているウェブページ最適化ツールで仮説パターンを生成する方法は主に 2 つある。

ひとつは、ウェブサイト開発者自身が対象となるウェブページの一部のデザインや機能を変えた仮説パターンを用意する方法である。それぞれの仮説パターンの URL を登録することで、ウェブページ最適化ツールが訪問者の振り分けおよびパフォーマンスの計測を自動で行う。Optimizely などの一部のウェブページ最適化ツールではウェブブラウザ上で仮説パターンの生成を可能にする仕組みを導入しており、文言の変更やボタンの色の変更などデザイン上の限られた部分であれば、ウェブサイト構築の知識が無いユーザでも編集ができるようになっている（図 2.1 参照）。

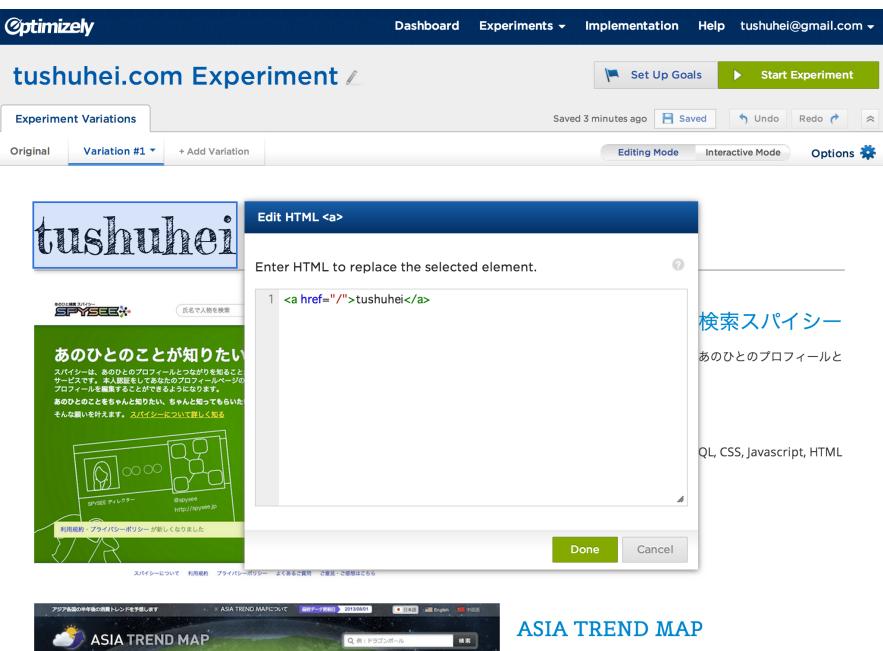


図 2.1 Optimizely での仮説パターン生成例（対象は筆者のポートフォリオサイト）

もうひとつは、仮説パターンの生成をクラウドソーシング^{*1}プラットフォームを介してウェブデザイナーに依頼する方法である。国内のウェブページ最適化ツール planBCD がこの方式を採用したサービスを展開している。クラウドソーシングを用いた仮説パターン生成方法では、まずユーザが対象となるウェブページと向上したい評価指標をクラウドソーシングプラットフォームに登録する。その後、その案件を見たウェブデザイナーがデザイン案を作成して同プラットフォームにアップロードする。アップロードされたデザイン案は対象のウェブページでテストされ、実際に向上したパフォーマンスに応じてウェブデザイナーに報酬が支払われる仕組みとなっている。

ウェブサイト開発者は仮説を生み出すために、まずログデータをみて現行のウェブページの問題点を認識し、実際のウェブページを触ったりユーザの声を聞いたりしながらユーザのアクションを阻害している要因を想像する。そして阻害要因を取り除く解決方法を考え、思いついたアイデアを仮説として構築するというプロセスを経て仮説パターンを生成する。つまり、ウェブサイト開発者は以下の一連のタスクを行なっていることになる。

- ウェブページの解析 ウェブページを見て、要素に分解して解釈する。
- 要素の重要度の推定 評価指標に大きく関連する重要な要素を抽出する。
- 仮説パターンの生成 抽出された要素の代替案を構築して仮説パターンとする。

仮説パターンの生成は人間が行う事が多いが、これらひとつひとつのタスクについては自動化を実現するための研究が進められている。本節では、これらのタスクを自動化する研究について説明する。

2.2.1 ウェブページの自動解析

ウェブページを自動で解析する試みは Webpage Segmentation と呼ばれ、情報抽出、検索エンジンの精度向上、ウェブページの再生成を主な動機として研究が進められてきた [8][9]。ウェブページを解析し、ウェブページを情報のかたまりとして意味のあるブロックに分けることを目的としており、主にウェブページの DOM 構造とテキスト情報に着目したアプローチとウェブページの視覚情報に着目したアプローチの 2 つがある。

^{*1} ウェブサービスを介して不特定多数の人間に業務を委託すること。

ウェブページの DOM 構造とテキスト情報に着目したアプローチ

ウェブページを HTML^{*2}のまま解析し、ソースコードに記されている DOM^{*3}の構造およびテキスト情報に着目してウェブページを構成する要素を抽出するアプローチである。HTML のタグに着目した手法では、<table> や などの特定のタグがウェブページのレイアウトおよびコンテンツの分割にとって大きな役割を果たしていることに着目し、ウェブページを分割する指針として活用している [10]。テキスト情報に着目した手法には、要素内に含まれる文字列の密度に着目したもの [9]、要素内に含まれる単語に意味解析を行なってその類似度に着目したものなどがある [11]。

これらのアプローチはテキスト情報をそのまま扱うので、低コストで解析を行うことができるというメリットがある。しかし HTML タグの利用法はウェブサイト開発者によって異なる上に、記法の流行り廃りもあるため、同一のアプローチがあらゆるウェブページに適用できるとは限らない。また、近年はブラウザが HTML5^{*4}に対応したことによって CSS^{*5}の表現力が増しており、DOM 構造から視覚構造を特定することが難しくなっている [12]。W3C^{*6}によってセマンティクスを表現する HTML と視覚構造を表現する CSS を分離することが推奨されており^{*7}、今後ますますこの流れは加速していくだろう。

ウェブページの視覚情報に着目したアプローチ

ウェブページをレンダリングすることでウェブページ全体を画像として捉え、コンテンツを分割するエッジを検出することでブロックへの分割を行うアプローチである。代表的なものに、VIPS アルゴリズムがある [13]。VIPS アルゴリズムでは、DOM 構造の葉ノードをサブブロックとして列挙し、その間に存在するエッジを画像情報から見つけ出すことによって、ウェブページのブロック分割を行う。分割されたブロックのツリー構造を再構築する Zone Tree Model という手法も提案されている [14]。

視覚情報のみに基づいて解析を行うため、HTML の記法や CSS の記法の影響を受けること

^{*2} HyperText Markup Language: ウェブページを記述するためのマークアップ言語である。

^{*3} Document Object Model: HTML および XML 文章をツリー構造をもったオブジェクトとして扱うための枠組み

^{*4} HTML の 5 回目にあたる改訂版。新たな HTML タグや CSS 属性が追加され、文章以外の要素をより柔軟に表現できるようになった。

^{*5} Cascading Style Sheets: ウェブページの視覚構造を記述するための枠組み

^{*6} World Wide Web Consortium: ウェブで利用される技術の標準化をすすめる非営利団体

^{*7} Conformance: requirements and recommendations <http://www.w3.org/TR/html4/conform.html#deprecated> (2014 年 1 月現在)

なくブロックへの分割が可能である。しかし、画像へのレンダリング処理を行うため、テキスト処理に比べて膨大な計算量が必要になるというデメリットがある。

2.2.2 ウェブページ要素の重要度推定

ウェブページを構成する要素が、ウェブサイトの目的に占める重要度を推定する研究が行われている。対象となる E コマースサイトをレイアウト・デザイン・表示する情報の種類などの属性によって表現しておき、それぞれのページに対するユーザの反応やヒアリングの結果を計測して、それぞれの属性の影響を算出する研究が行われている [15]。複数のオークションサイトを数十の属性で表現し、それぞれのウェブサイトにおけるユーザの入札意思から各属性がユーザの行動に与える影響度を算出する研究も行われている [16]。

2.2.3 ウェブページの自動生成

ウェブページを自動生成する試みは、デスクトップ向けのウェブページを自動でスマートフォン向けのページにレイアウトを再生成したり、ユーザの嗜好に応じてウェブページのデザインを自動で変更することを主な動機として研究が行われてきた。レイアウトの再生成には、Web Page Segmentation で抽出したブロックの重要度の推定が密接に関わっており、重要度が高いものから優先的に並べていくアプローチがある [17][18]。

ブロックの重要度を推定する手法のひとつは、同じドメインに含まれているウェブページの HTML の比較を行うことで、広告やナビゲーションなどコンテンツと関係のない部分を重要度の低いノイズとして取り除くものである [19]。

複数のウェブサイトの要素の対応付けをユーザに入力してもらうことで学習し、テンプレートとコンテンツを入れ替えて新しいウェブページを生成する手法も提案されている [12](図 2.2 参照)。

2.3 探索手法にまつわる研究

本節では、仮説パターンの中から評価指標の高いパターンを探索するために提案されている手法について紹介する。仮説パターンはウェブページを構成する要素がオリジナルのウェブページと異なる値を取ることによって生成される。ウェブページを構成する要素を変数、それぞれのパターンを変数の組み合わせによって生成される解とすると、評価指標が高いパターンを探索することは各変数の最適な組み合わせを探索することに相当する。したがって、仮説パ

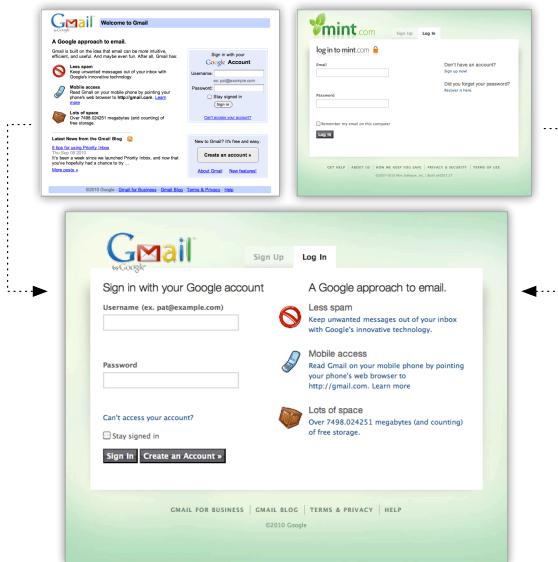


図 2.2 Bricolage によるテンプレートとコンテンツの入れ替え

ターンの中から評価指標の高いパターンを探索することは、一種の組み合わせ最適化問題として捉えることができる。

まず 2.3.1 項では実際のウェブページ最適化ツールで用いられている最適化手法について説明する。2.3.2 項で組み合わせ最適化問題で一般的に用いられている探索手法について説明し、2.3.3 項で効率的な探索のために導入できる工夫について説明する。

2.3.1 既存のウェブページ最適化手法

A/B テスト

A/B テストはウェブページ最適化の最もシンプルな形であり、たくさんの先行事例が存在している [5]。A/B テストではオリジナルの解を対照群 A、仮説による操作を施した解を実験群 B として、ユーザを 50% ずつランダムに振り分ける。それぞれのユーザの反応を比較することで、最適解を探し出す。対照群 A と実験群 B で複数の変数に変更を加えてしまうと、2 群間の差異がどの変数の効果に依るものかがわからなくなってしまうため、値を変更する変数は一つに絞って行うことが多い。

多変量テスト

A/B テストはひとつの変数について最適な値を探す上では理解しやすい手法だが、最適化したい変数が複数ある場合は、A/B テストではひとつの実験が終わるまでは他の変数の最適

化を行うことができないため、時間がかかってしまう。そこで、複数の変数の最適化を同時に一回のテストで行うのが多变量テスト (Multivariate Testing) である。多变量テストではすべての実行可能解を列挙してランダムにユーザを割り振り、それぞれのユーザの反応を比較することで最適解を見つけ出す。さらに分散分析や回帰分析などの解析を行うことによって、各変数の影響度を算出することができる [20]。

このように戦略的に予め評価する解を設定し、収集されたデータに分析を加えて結果を導く手法を実験計画法と呼ぶ。実験計画法には主に総当り実験 (Full Factorial Test Design) と直交配列実験 (Fractional Factorial Test Design) の二つの方法がある。

- 総当り実験 総当り実験はすべての組み合わせを列挙し、対照比較実験を行う方法である。総当り実験ではすべての解を評価するため、厳密に最適解を見つけることができるが、変数の数が多くなると組み合わせ爆発によって評価すべき解の数が膨大になってしまふ。
- 直交配列実験 直交配列実験は Sparsity-Of-Effects Principle と呼ばれる「3つ以上の変数による相乗効果は十分に小さいため無視できる」「一部の2変数間の相乗効果は小さいため無視できる」といった仮定を置くことで、評価する解を絞って対照比較実験を行う手法である。総当り実験に比べて評価すべき解の数を大幅に削減できるという利点があるため、古くから農業や工業の分野で活用されてきた [21][22]。

直交配列実験で評価する解は、直交計画に基いて決定される。直交計画とは、それぞれの変数が取りうる値を同じ回数だけ評価することができるよう設計された実験の割りつけ方法である。直交計画はベクトルの直交性に関する複雑な計算を要するため、予め作成された直交表をもとに割りつけを行うことが多い [21]。

直交配列実験は総当り実験と違って一部の解しか評価しないため、各変数における最適な値の組み合わせを最適解とする。

バンディットアルゴリズム

これまでに説明した探索手法では、できるだけ少ない回数で最適解を見つけ出すことに主眼を置いていたが、バンディットアルゴリズムでは実験中に得られる利益にも着目する。バンディットアルゴリズムは実験期間中でも評価値の期待値が高い解を優先的に表示することで利益をあげつつ、今までに評価していない解も表示させながら最適解を探索する [23]。ここではバンディットアルゴリズムの中でも代表的な epsilon-greedy、Softmax、そして UCB1 と呼

ばれるアルゴリズムを説明する。

- **epsilon-greedy** epsilon-greedy アルゴリズムではあらかじめ非常に小さな値としてパラメータ $0 < \varepsilon < 1$ が設定されており、 ε の確率で探求を行い、 $1 - \varepsilon$ の確率で活用を行う。探求が選択された場合は最も評価値の期待値が大きい解を選んで表示する。一方、活用が選択された場合には実行可能解の中からランダムに一つの解を選んで表示する。このように探求と活用の間を行き来することで未評価の解を評価しつつも、評価値の期待値が大きい解を優先的に表示することで実験による損失を免れている。簡単に実装ができる反面、解の期待値に関わらず探求または活用を選択するため、期待値に大きな差がある場合でも期待値が低い解を選んでしまう可能性がある。
- **Softmax** Softmax アルゴリズムでは、解の評価値の期待値に応じて表示する確率を変化させる。解空間を X 、観測データから算出される解 $x \in X$ の評価値の期待値を \bar{y}_x とすると、解 x をユーザに表示する確率 $p(x)$ は式 2.1 によって表される。

$$p(x) = \frac{\exp(\bar{y}_x/\tau)}{\sum_{x \in X} \exp(\bar{y}_x/\tau)} \quad (2.1)$$

τ は温度と呼ばれるパラメータであり、探究心の強さを表している。温度が非常に高いとき、すなわち $\tau \rightarrow \infty$ のときは解 x を選ぶ確率 $p(x)$ は $1/|X|$ に収束するため、すべての解が均等の確率で選ばれることになる。逆に温度が低いときは \bar{y}_x が効き始めるため、最も評価値の期待値が高い解が 1 に近い確率で選ばれるようになる。

- **UCB1** UCB1 ではこれまでに紹介したアルゴリズムとは異なり、ランダム性を用いない。UCB1 では基本的に評価値の期待値 \bar{y}_x が最も高い解を選ぶ戦略だが、解を選んだ回数に応じてボーナスが追加される。解 $x \in X$ を表示した回数を t_x とすると、解 x の UCB 値 u_x は

$$u_x = \bar{y}_x + \sqrt{\frac{2 \log(\sum_{x \in X} t_x)}{t_x}}$$

と算出され、この UCB 値を最大にする解 x が選択される。

バンディットアルゴリズムは実験を行いながら、その実験の過程で評価値の低い解をフィルタリングし、最適解を常に表示する状態に徐々に移行するアルゴリズムの枠組みだということができる。つまり、仮説パターンとアルゴリズムさえ設定しておけば、人手を挿むことなく自動で最適化を行うことができる [23]。Amazon.com^{*8} のトップページでは同様の手法を用い

^{*8} Amazon <http://www.amazon.com/>

て最適化の自動化を実現している [5][24]。

2.3.2 組合せ最適化問題に対する探索手法

次に、一般的な組合せ最適化問題に適用される探索手法について説明する。

全探索

全探索は、解空間のすべての解について評価を行い、最も評価値が高いものを選択する手法である。確実に最適解を発見することができる厳密解法だが、解空間のサイズによっては膨大な評価回数が必要になる。

ランダム探索

ランダム探索は解空間から無作為に抽出して評価を行い、その中で最も評価値が高いものを最適解として選択する手法である。全探索に比べると評価回数を減らすことができるが、解の抽出は無作為に行われるため最適解を選択することができる保証は無い。

山登り法

山登り法 (HC: Hill Climbing) は暫定解に小さな操作を施すことによって近傍解を生成し、近傍解に暫定解よりも評価値が高い解があればそれを暫定解と入れ替えるという動作を繰り返すことで最適解を探索する。評価値が高い解に似た解も同様に評価値が高いというヒューリスティクスを導入した探索手法である。このように近傍解との比較を繰り返す探索手法全般を局所探索と呼ぶ [25]。

簡単に実装できる反面、極値に値が収束してしまい局所解に陥りやすいという欠点がある。局所解への収束を克服するための工夫として、解が収束した後に再びランダムに初期値を選択して探索を開始するランダム再スタート山登り法 (Multi-start Hill Climbing) がある。

焼きなまし法

焼きなまし法 (SA: Simulated Annealing) は局所探索法の一種であり、山登り法の局所解に陥ってしまう欠点を克服するための工夫を取り入れた手法である。焼きなまし法では近傍解の評価値が暫定解より悪い場合でも、ある確率で暫定解の更新を行うこと（受理と呼ぶ）を許可する。受理が行われる確率は温度と呼ばれるパラメータによって決定され、温度が高いほど受理が行われる確率が高くなる。探索の初期では高い温度を設定することで局所解に収束させず大域的な探索を行い、探索が進むにつれて温度を低くしていくことで最適解に収束させる。

良い解を得るために、温度を低下させる方法（冷却スケジュールと呼ぶ）を適切に設定する必要があり、パラメータを試行錯誤して調整する必要がある。

このようにヒューリスティクスを応用して、汎用的な問題に対応することができるよう改良したアルゴリズムの枠組みのことをメタヒューリスティクスと呼ぶ。メタヒューリスティクスは特定の問題の解法に依存せず、最適解の探索精度を多少犠牲にしながらも困難な問題に対して高速に近似解を得ることができるという特徴を持つ [25]。

タブーサーチ

タブーサーチ (TS: Tabu Search) はタブーリストを導入することで、同じ解を探索してしまうサイクルが発生することを防ぎ、効率的に探索を行うことを可能にするメタヒューリスティクスである。山登り法や焼きなまし法と同様に局所探索法の一種であるが、暫定解から近傍解への更新が行われるたびにその記録がタブーリストに記されていく点が異なる。ある暫定解から近傍解への更新が以前にも行われておりタブーリストに記されていた場合はその評価を行わず、評価値が悪くてもタブーリストには記されていない近傍解への更新を優先することで、局所解へ陥ることを防いでいる。タブーリストの長さがより良い最適解を得るために重要なパラメータになっており、試行錯誤して適切に設定する必要がある。

遺伝的アルゴリズム

遺伝的アルゴリズムは生物の進化過程を模して探索を行うメタヒューリスティクスである。これまでに述べた山登り法、焼きなまし法およびタブーサーチはひとつの解について更新を行う単点探索手法だが、遺伝的アルゴリズムでは複数の個体群による多点情報を用いて探索を行う多点探索手法に分類される [25]。様々な分野で広く活用されているアルゴリズムであり、実際に E コマースサイトにおいて遺伝的アルゴリズムを活用して商品を見せる順番を最適化する研究も行われている [26]。

遺伝的アルゴリズムでは「世代」と呼ばれるサイクルを繰り返すことで、最適解を求める。世代と世代の間には「淘汰」「交叉」「突然変異」の3つの操作が施し、次の世代で探索を行う解集合を決定する。

- 淘汰 淘汰は評価値の低い解を探索対象の解集合から取り除く操作である。自然界において環境への適合度が低い個体が、自然から淘汰されることに対応している。
- 交叉 交叉は淘汰を耐えて生き残った解同士で、解の一部を交換したものを次の世代で探索する解集合に追加する操作である。優秀な解の良い部分同士を兼ね備えた解が生まれる。

れる可能性があるため、さらに評価値の高い解を得られる可能性がある。

- 突然変異 突然変異は解のある変数を他の値に書き換えることである。適合度の低い解を再び生成してしまうリスクがある一方、解集合の収束を防いで多様性を確保するために重要な役割を果たす操作である。

2.3.3 効率的な探索の工夫

ヒューリスティクスを用いた探索は予め設定するパラメータによって大きく性能が変わるために、そのチューニングが重要な課題のひとつである。直交配列実験に基いて推定した最適解を初期解として局所探索を行うことで、無作為に初期解を選択して局所探索を行う場合よりも良い解に到達することができる事が示されている [27][28]。実験計画法を用いて解空間全体の特性を把握した上で最適化を行うことで、効率的な探索を実現している。

一方、強化学習の分野を中心に、解に対する評価値がばらつきを持った値として与えられる場合の最適化問題に対する解法が研究されている。各解に対する評価値がばらつきを持った値として与えられる状況で探索を行う場合、区間推定によって各解の評価値がとりうる値域を推定することで、解の比較を行うことができる [29]。また、明らかに評価値が低い解を他の解と同じ回数観測することは観測回数に無駄が生じるため、有意に他の解に劣る解を評価対象から外すことによって探索を高速化する方法がレーシングアルゴリズムとして提案されている [30]。レーシングアルゴリズムはメタヒューリスティクスにも応用され、その工夫がもたらす効率性が示されている [31]。

2.4 最適化プログラムの実装方法

本節ではウェブページ最適化を行うプログラムを実装する方法について説明する。解の割り当てとウェブページの書き換えを担う部分を設置する場所によって主に3種類の方法に分けることができる。

2.4.1 プロキシサーバ方式

ブラウザとウェブサーバの間にプロキシサーバ^{*9}を挟み、解の割り当てとウェブページの書き換えを担わせる手法である。ウェブページ最適化に必要な機能はプロキシサーバ内で完結す

^{*9} クライアントからウェブサーバへの接続の際に中継するサーバ

るため、既存のクライアントサイドおよびサーバサイドのコードには変更を加えることなくウェブページ最適化を行うことができる [32]。

その一方で、すべてのアクセスがプロキシサーバを介することになるので、プロキシサーバにバグが含まれるとウェブサイト全体のパフォーマンスに影響を与えててしまう。また、ウェブサーバからプロキシサーバにウェブページが送られてきた時点でウェブページの書き換えが行われるため、バックエンドのアルゴリズムに関するテストは行うことが出来ないという欠点がある [5]。

2.4.2 クライアントサイド方式

ユーザのトラッキングとコンテンツの修正を行う javascript^{*10}で記されたプログラムを作成し、ウェブページとともにユーザにダウンロードさせる手法である。ウェブページ開発者は javascript コードを HTML コードに埋め込むだけで簡単にウェブページ最適化をはじめることができるため、Google Website Optimizer, Optimizely など多くのウェブページ最適化ツールがこの手法を採用している。プロキシサーバを用いた実装とは異なり新しい設備投資の必要もなく、既存のコードの一部に変更を加えるだけでウェブページ最適化をはじめることができます。

しかし、変更できる要素は javascript によってアクセスできる部分に限られてしまう。また、javascript の実行速度はクライアントのコンピュータのスペックやブラウザの性能によるため、読み込み時間を遅延させる可能性がある [5]。

2.4.3 サーバサイド方式

ウェブサイトのサーバサイドでユーザのトラッキングおよびコンテンツの修正を行う手法である。これまでの手法とは異なり、バックエンドのアルゴリズムに関するテストも含めて自由にテストを構築することができる。ウェブページ最適化を導入する前と同様のサーバサイドの処理によって最適化を行うことができるため、読み込み時間の遅延によってユーザ体験を損なうことなく実装することができる。

しかし、サーバサイドのコードの修正を行うため、ウェブページ最適化をはじめるための変更コストが大きいという欠点がある。また、ウェブページ最適化のコードがウェブサイトを表

^{*10} プログラミング言語のひとつ。ウェブブラウザに実装され、動的なウェブページやインターフェイスの開発に用いられる。

示するためのコードに深く入り込むことになるため、ウェブページ最適化のための実験を行うことがウェブサイトにバグを埋め込むリスクをはらんでいる [5]。

2.5 ウェブ特有の実験上の注意点

実験によってある変数の値を変更した効果を検証するために、操作を及ぼした要素以外は可能な限り条件を揃えたいが、完全に制御するのが難しいウェブ特有の外的要因がいくつか存在する。ウェブページ開発者はこれらの外的要因が入り込む可能性があることを意識しながら、実験を設計することが望ましい。

2.5.1 ロボット

ウェブには検索エンジンのクローラやメールアドレス収集ボットなどの様々なロボットが存在しており、常にあらゆるウェブサイトを巡回している。ロボットは一般的な人間の操作とは異なる動きを示すので実験のサンプルからは除外することが望ましいが、近年はクッキーを受け取るロボットやブラウザを介して動くロボットが登場しており、確実にロボットを特定することは難しくなっている [6][7]。

2.5.2 目新しさ効果

ナビゲーションバーが取り払われたりレイアウトが変更されたりデザイン上の大変更が実験に投入された場合、その変更がユーザにとって目新しいために、投入直後のクリック数が異常に増えるといった効果が現れる可能性がある。目新しさ効果が落ち着く十分な期間を取つて計測することが望ましい [5][24][33]。

2.5.3 ユーザ体験の一貫性

ウェブページ最適化では同一のユーザに同一のパターンを表示するために、クッキーを用いてブラウザに情報を埋め込み、同じブラウザからのアクセスからは同じパターンを返すように実装する。しかし、ユーザによっては同一のウェブサイトに複数のデバイスからアクセスすることもあるため、そこで異なるパターンが割り振られると、一人のユーザが複数のバージョンを見てしまうことになり、ユーザの混乱を招いてしまう可能性がある [5][6]。

2.5.4 時間要因

ウェブページを取り巻く様々な外的要因は常に変化しているため、同一のパターンであっても時間によって評価値が変化することがある。時間要因は訪問者の利用シーンや利用形態の変化であったり、季節性のあるイベントやキャンペーンによる嗜好の変化であったり様々である。たとえば、以下のような例が考えられる [23]。

- 時間帯によるユーザの利用形態の変化 朝と夕方の通勤時間帯ではスマートフォンからのアクセスが増加する。一方、昼間は職場や学校のパソコンからのアクセスが増加する。
- 曜日によるユーザの利用形態の変化 平日は仕事での調査目的でアクセスするユーザが増加する。一方、休日は暇つぶしの目的でアクセスするユーザが増加する。
- 時期によるユーザの嗜好の変化 ハロウィーンの時期はオレンジと黒を基調としたロゴやデザインに魅力を感じる。一方、クリスマスの時期は赤と緑と白を基調としたものに魅力を感じる。

データを収集する際には平日と休日で分ける、できるだけ直近のデータを使うなどの工夫を施すことが望ましい。

第3章

ウェブページ最適化問題

ウェブページ最適化とは、デザインや機能の一部が異なるウェブページのバリエーションの中からクリック率やページ滞在時間などの評価指標を最大化するパターンを探索することである。それぞれのパターンはウェブページを構成するボタンやラベルなどの要素の組み合わせによって構成されている。したがって、ウェブページ最適化は評価指標の値を最大化する要素の組み合わせを探しだす組合せ最適化問題のひとつとして捉えることができる。

3.1 節では、ウェブページ最適化問題の定式化を行う。3.2 節では、局所探索法を用いたウェブページ最適化問題の解法を説明する。3.3 節と 3.4 節では、効率的な探索を可能にするために導入できる工夫について説明する。3.5 節ではウェブページ最適化問題の視点から、既存のウェブページ最適化手法を整理する。

3.1 問題の定式化

ウェブページのパターン $x = (x_1, \dots, x_m)$ はウェブページを構成する m 個の要素の組み合わせによって形成されている。それぞれの要素 x_i は離散値であり、その取りうる値の集合 $V_i = \{v_{i1}, \dots, v_{il_i}\}$ の中からひとつ選ばれる。ただし要素 x_i が取りうる値の数を l_i とする。

たとえば、ある食料品を紹介するウェブページ x が商品の写真 x_1 キャッチコピー x_2 購入ボタンの色 x_3 という 3 つの要素の組み合わせによって形成されているとき、購入ボタンの色 x_3 は $V_3 = \{ \text{赤, 青, 緑, 黄} \}$ のうちひとつの値を取ることになる。

ウェブサイト開発者はウェブページのパフォーマンスを向上させるために、ウェブページの評価指標 $f(x)$ を最大化するパターン x^* を求めたい。つまり、

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in X} f(\mathbf{x}) \quad (3.1)$$

なるパターン x^* を見つけることがウェブページ最適化の目的となる。

しかしウェブサイト開発者は評価指標 $f(x)$ を直接観測することは出来ない。パターン x をユーザに表示した時のユーザの反応から計測される観測値 y がある確率分布 $p(y|x)$ に従って与えられるのみである。ここで評価指標 $f(x)$ が観測値 y の条件付き期待値 $\mathbb{E}[y|x]$ によって推定できるとすると、式 3.1 は式 3.2 のように書き換えることができる。

$$x^* = \arg \max_{x \in X} \mathbb{E}[y|x] \quad (3.2)$$

実際にウェブページ最適化を行う場合は実験に割くことができる時間が限られていることが多く、限られたユーザ数から得られる観測値をもとに最適化を行う。ウェブページ最適化のために収集できる観測値の数を限界観測数 N 、 n 番目のユーザに表示するパターン x_n とそれにに対する観測値 y_n の対の集合 $Y = \{(x_1, y_1), \dots, (x_n, y_n)\}$ を観測データと呼ぶことになると、この制約は式 3.3 のように表すことができる。

$$\sum_{x \in X} |Y| \leq N \quad (3.3)$$

式 3.2 と式 3.3 より、ウェブページ最適化問題は式 3.4 のように定式化することができる。

ウェブページ最適化問題

解 $x \in X$ について得られる観測値を y 、限界観測数を N 、観測データを Y とするとき、下式を満たす解 x^* を求めよ。

$$x^* = \arg \max_{x \in X} \mathbb{E}[y|x] \text{ s.t. } \sum_{x \in X} |Y| \leq N \quad (3.4)$$

3.2 局所探索法を用いた解法

ウェブページ最適化問題は一種の組合せ最適化問題であるため、組合せ最適化問題に用いることができる基本的な探索手法を導入することができる。確実に最適解を発見することができる厳密解法である総当たり探索を適用することもできるが、ウェブページ最適化問題には下記の特徴があり、観測回数に対して強い制約が与えられている。そのため、すべての解を探索する厳密解法は必ずしもウェブページ最適化問題においては最適な探索手法とは言えない。

ウェブページ最適化問題の特徴

- 組み合わせによる解の表現 解 x は独立した変数の組み合わせによって表現されるため、変数の数に対して解空間のサイズ $|X|$ が指数関数のオーダーで増加する。
- ばらつきを持った評価値 解 x に対する評価関数 $f(x)$ の値は、ある確率分布 $p(y|x)$ に従うばらつきを持った観測値 y として与えられる。
- 限界観測数 限界観測回数 N が予め設定されている。

厳密解法を適用することが難しい場合、最適性の保証は無いがある程度良い解を与える近似解法としてヒューリスティクスを用いた探索手法がある。ヒューリスティクスを用いた探索は高速に近似解を得ることができる解法として広く用いられており、観測回数の制約が大きいウェブページ最適化問題と相性が良い解法と考えられる。特に組合せ最適化問題では Proximate Optimality Principle と呼ばれる「良い解同士は似通った構造を持っている」という仮定が成り立つことが多く、局所探索法が広く用いられている [34]。ここではウェブページ最適化問題の解法として局所探索法を取り上げて説明する。

3.2.1 局所探索法の概要

まず局所探索法のアルゴリズムについて説明する (Algorithm 1 参照)。

Algorithm 1 局所探索法のアルゴリズム

Choose $x \in X$.

repeat

$X' \leftarrow \text{Neighbors}(x)$ ▷ 近傍解の生成

$x \leftarrow \text{Move}(x, X')$ ▷ 暫定解の移動

until 終了条件

return x

局所探索法では、まず適当に初期解 $x \in X$ を設定する。この初期解を暫定解とし、暫定解から近傍解 X' を生成する。近傍解の中に暫定解より良い解があればその解に移動する。局所探索法ではこの近傍解の生成と暫定解の移動を終了条件を満たすまで繰り返すことで、最適解を求める [35]。

3.2.2 近傍解の生成

近傍解とは、暫定解に小さな操作を加える事で得ることのできる、暫定解と似た解のことです。この際に加える操作のことを近傍操作と呼ぶ[35]。ウェブページ最適化問題における解 x は独立な変数の組み合わせによって表現されるため、ある変数 x_i の値を取りうる値の集合 V_i に含まれる他の値に入れ替えることによって、近傍解 X' を生成することができる。ウェブページの例で言えば、ウェブページに含まれるボタンの色 x_i がもともと赤だったものを、青や緑などの色に変えることに相当する。値を入れ替える変数 x_i の数および V_i から取り出す値の数によって近傍解の大きさ $|X'|$ が変化する。

3.2.3 暫定解の移動

暫定解の移動とは暫定解と近傍解の間で解の評価を行い、暫定解より良い解が近傍解にあれば暫定解を更新することである。ここで解の評価とは、ある解 x の評価値を求めることである。ウェブページ最適化問題では解 x の真の評価値 $f(x)$ を直接観測することができないため、解 x の観測値の条件つき期待値 $E[y|x]$ によって解の評価を行う。

一般に近傍解は複数存在するため、解を評価する順番および暫定解の移動を行うルールによって探索の挙動が変化する。したがって、局所探索法において暫定解の移動と解の評価は密接な関係にある。

3.3 推測統計学の活用

ウェブページ最適化問題では解 x に対する評価値がばらつきをもった観測値 y として与えられる。ばらつきを定量的に扱い効率的な探索を実現するために、推測統計学的な手法を活用することができる。ここでは推測統計学的の代表的な手法である仮説検定と区間推定をウェブページ最適化問題に活用する方法について説明する。

3.3.1 仮説検定

推測統計学的手法を用いて複数の解を評価する一つの方法として、仮説検定がある。仮説検定では証明したい仮説を対立仮説、対立仮説に相反する仮説を帰無仮説と呼ぶ。仮説検定では帰無仮説のもとで検定統計量を求め、検定統計量が帰無仮説と矛盾することを示す背理法によって、帰無仮説を棄却して対立仮説を採択する。

検定統計量と帰無仮説の矛盾は、対象の検定統計量が従う分布において与えられた検定統計量が発生する確率 (p 値と呼ぶ) を求めることによって評価する。 p 値が与えられた有意水準 α よりも小さい場合には、帰無仮説の上では妥当ではない事象が起きたと判断し、帰無仮説を棄却して対立仮説を採用する。逆に、有意水準 α よりも大きかった場合には、帰無仮説を棄却できず受容する。有意水準 α は一般的に 0.05 や 0.01 などの小さな値を取ることが多い [36]。

検定に用いる検定統計量は評価値に仮定する確率分布によって異なる。評価値が正規分布に従って与えられると仮定する場合は t 値を算出する t 検定によってその平均値の差を検定する。評価値がベルヌーイ試行によって二値で与えられる場合は、二項検定やカイ二乗検定によってその二値が与えられる比率の違いを検定する。評価値がある確率分布に従う仮定する検定手法をパラメトリック手法、評価値がある確率分布に従うことを仮定しない検定手法をノンパラメトリック手法と呼ぶ。仮説検定を用いる際には検定する値と評価値に仮定する確率分布に応じて適切な検定方法を選択し、検定を行う。

暫定解と比較する近傍解が複数個ある場合には、多重比較法を用いることで複数解の検定を行うことができる。しかし、多重比較を用いた場合は多重性が生じる可能性があるため、適切な補正を行なった上で検定を行う。

3.3.2 区間推定

区間推定は、観測値から真の評価値が取りうる範囲を推定することで比較を行う方法である。 $Y_x = \{(x_t, y_t) \in Y | x_t = x\}$ なる Y の部分集合をサンプル Y_x とすると、信頼係数を 95% としたとき、 $|Y_x|$ が十分に大きく観測値が正規分布に従って与えられると仮定するならば、評価指標 $f(x)$ は

$$\mu_{Lx} = \mathbb{E}[y|x] - 1.96 \frac{s_x}{\sqrt{|Y_x|}} \leq f(x) \leq \mathbb{E}[y|x] + 1.96 \frac{s_x}{\sqrt{|Y_x|}} = \mu_{Ux} \quad (3.5)$$

と推定することができる。ただし、

$$s_x = \sqrt{\frac{\sum_{y_i \in Y_x} (y_i - \mathbb{E}[y|x])^2}{|Y_x|}}$$

である。 μ_{Lx} は評価関数 $f(x)$ の下側信頼限界、 μ_{Ux} は評価関数 $f(x)$ の上側信頼限界と呼ばれる。

区間推定を活かして探索を効率化する方法として、レーシングアルゴリズムがある。レーシングアルゴリズムでは観測データから比較対象の解 $x \in X$ の評価値の信頼区間 $[\mu_{Lx}, \mu_{Ux}]$ を隨時推定する。

ある解 x_l が式 3.6 を満たすとき、その解を敗者解として近傍解から取り除くことで、有望な近傍解の評価に多くのサンプルサイズを割くことができる。本研究ではこの工夫を敗者解のカットオフと呼ぶ。

$$\mu_{Ux_l} < \max_{x \in X \setminus \{x_l\}} \mu_{Lx} \quad (3.6)$$

一方、ある解 x_w が式 3.7 を満たすとき、その解を勝者解と呼ぶ。勝者解が発見された時点で観測を中断して勝者解に暫定解を移動することで、観測数を節約できるようになる。本研究ではこの工夫を観測の短縮と呼ぶ。

$$\mu_{Lx_w} > \max_{x \in X \setminus \{x_w\}} \mu_{Ux} \quad (3.7)$$

3.3.3 解の評価に用いるサンプルサイズ

統計学的手法を用いる場合は、各解の評価に用いるサンプルサイズも重要な要素である。解の評価に用いるサンプルサイズを小さくすると暫定解の移動を行う頻度が上がるため探索は速くなるが、解の評価の精度が落ちてしまい、真の評価値が低い解に移動してしまう確率が大きくなる。一方解の評価に用いるサンプルサイズを大きくすると、暫定解の移動を行う頻度が下がるため探索は遅くなるが、解の評価の精度が向上するため、真の評価値が低い解に移動する確率が小さくなる。ウェブページ最適化問題では、探索のスピードと解の評価の精度の間のトレードオフの中で探索を行う。

区間推定において評価値が推定される幅のある値以下にするために必要なサンプルサイズは式 3.5 を n について逆算することによって得られる。

一方、2 つの解の平均値を仮説検定によって比較するために必要なサンプルサイズ n は、要求する有意水準 α 、検出力 $1 - \beta$ 、そして期待する効果量 Δ から算出することができる。効果量 Δ とは、比較したい平均値 μ と μ' の差を標準偏差 σ で割ることで標準化したものであり、

$$\Delta = \frac{|\mu' - \mu|}{\sigma}$$

と表される。有意水準 $\alpha = 0.05$ 、検出力 $1 - \beta = 0.80$ としたとき、解あたりのサンプルサイズは

$$n = \frac{16}{\Delta^2} = \frac{16\sigma^2}{(\mu' - \mu)^2}$$

と算出される [37]。

3.4 線形モデルによる最適解の予測

組合せ最適化問題に局所探索法を用いる場合、探索の初期値の設定によってその後の探索のパフォーマンスが変わることが多い。線形モデルによる最適解の予測では、

$$y = a_0 + \sum_{i=1}^{l_1-1} a_{1i}x_1^i + \cdots + \sum_{i=1}^{l_m-1} a_{mi}x_m^i$$

と表される回帰モデルを仮定し、各変数 x_i において観測値の期待値を最大化する値を組み合わせたものを最適解 x^* とする。評価値が高いと予測される解から探索を行うことで、ランダムに選択された解から探索を行う場合に比べて高速に最適解に到達することができる。

3.5 既存のウェブページ最適化手法の整理

本節ではウェブページ最適化問題の視点からあらためて既存のウェブページ最適化手法を見直し、整理を行う。

3.5.1 評価指標および検定方法の設定

ウェブページ最適化問題に局所探索法を用いる場合は、あらかじめ設定した評価指標に応じた確率分布を仮定し、適切な方法を用いて解の評価を行う。表 3.1 に示すように、ウェブページ最適化に用いる評価指標はその性質に応じて 3 種類に分けることができる [7]。「検定方法例」には仮説検定を用いて解の評価を行う際に適用する検定方法の例、「評価指標例」列には E コマースサイトにおいて訪問を単位とした時の評価指標の例を示した。

表 3.1 評価指標の分類と対応する確率分布

評価指標の性質	仮定する確率分布例	検定方法例	評価指標例
二値	ポアソン分布	二項検定	購入の有無、クリックの有無
カウント値	負の二項分布	カイ二乗検定	サイト内検索回数、購入金額
連続値	正規分布	<i>t</i> 検定	滞在時間

ひとつの目標に対して考えられる評価指標は一種類であるとは限らない。たとえば動画共有サイトのようにユーザにできるだけ長時間滞在してもらい、広告をクリックしてもらうことで広告収入を得ているサイトであれば、広告のクリック率以外にウェブサイトの滞在時間も評価

指標として設定することができる。その他にも訪問あたりの動画再生数や総視聴時間のような評価指標も利用できる。実験を行う際はウェブサイトの目的に合わせて有効な評価指標をリストアップし、表 3.1 で整理した分類に合わせて評価指標の背後にある確率分布を推定したうえで、できるだけばらつきの小さい評価指標を用いることが望ましい。

3.5.2 既存のウェブページ最適化手法の位置づけ

反復型 A/B テスト

反復型 A/B テストでは、無作為に選んだ解を初期解 x_0 として局所探索を開始する。暫定解のある変数 x_i の値を変化させた解を近傍解 $x' \in X'$ とし、近傍解 x' の中に暫定解より評価値の高い解があればその解に移動することで、ひとつの A/B テストが完了する。その後また新たな変数 x_j を取り出し、探索を繰り返す。

多变量テスト

多变量テストは総当たり探索手法を採用して解の比較を行うことに相当する。直交配列実験は各变数の最適な値の組み合わせを最適解として推定しているという点で、3.4 節で説明した「線形モデルによる最適解の予測」工夫を用いていると言える。

バンディットアルゴリズム

バンディットアルゴリズムでは解空間に含まれるすべての解 $x \in X$ を一様の「スロットマシンの腕」とみなして探索を行う点で総当たり探索に相当する。バンディットアルゴリズムでは各解 x の条件付き期待値 $\mathbb{E}[y|x]$ に応じて、各解に割り当てるサンプルサイズを変化させる。これは 3.3.2 項で説明した「敗者解のカットオフ」工夫に相当している。

最後に、既存のウェブページ最適化手法とウェブページ最適化問題を解くための探索手法および導入される工夫との関係を表 3.2 にまとめる。

表 3.2 既存のウェブページ最適化手法とウェブページ最適化問題における探索手法と工夫の対応付け

既存のウェブページ最適化手法	探索手法	工夫
反復型 A/B テスト	山登り法	なし
全探索実験	総当たり	なし
直交計画実験	総当たり	最適解の予測
バンディット	総当たり	敗者解のカットオフ

3.6 本章のまとめ

ウェブページ最適化問題は、解に対する評価値がばらつきを持った確率的な値として与えられる上に、観測回数についての制約がある中で探索を行う組合せ最適化問題である。あらゆる探索手法を適用して解くことができるが、ウェブページ最適化問題は観測回数について強い制約が与えられているため、高速に近似解を得ることができるヒューリスティクスを用いた探索が有効な解法だと考えられる。

ヒューリスティック探索のひとつである局所探索法では、近傍解の生成と暫定解の移動を繰り返して探索を行う。近傍解は暫定解のある変数の値を他の値に入れ替えることによって生成することができる。暫定解の移動は観測値から算出される条件付き期待値を比較することによって実現される。

観測値のばらつきを定量的に扱い、探索を効率化するために統計学的手法を活用することができる。区間推定を応用したレーシングアルゴリズムでは、暫定解を有意に下回る敗者解の観測を中断する「敗者解のカットオフ」、暫定解を有意に上回る勝者解が見つかった時点で暫定解の移動を行う「観測の短縮」を導入することで、観測数を無駄なく活用することができる。

「線形モデルによる最適解の予測」では、解空間全体から得られた観測値から線形モデルによって予測された最適解を探索の初期解として用いることで、初期解をランダムに選んだ場合に比べて効率的に探索を行うことを可能にする。

第4章

提案手法

4.1 節ではウェブページ最適化問題において限界観測数および解空間のサイズに応じて最適解を探索することができる手法を提案する。4.2 節では、ウェブページ最適化問題に導入することができる「観測データの活用」工夫について説明する。4.3 節では、提案手法の具体的な探索の手順をアルゴリズムとして示す。

4.1 提案手法の概要

3.2 節にてウェブページ最適化問題は観測回数について強い制約があるため、厳密解法よりも高速に近似解を発見する近似解法が有効であることを説明し、局所探索法探索を用いた解法を示した。しかし、ヒューリスティクスを用いた探索は初期解の設定や近傍解のサイズなどのパラメータによってその探索の精度が大きく変化する。特にウェブページ最適化問題では、解の評価に用いるサンプルサイズの設定が探索の効率性を左右する。与えられた限界観測数と解空間のサイズに応じた最適化を行うためには、状況に応じて適切なサンプルサイズを設定できることが望ましい。

たとえば、限界観測数に余裕があり解空間のサイズも十分小さい場合は、解の比較に十分なサンプルサイズを割いて着実に暫定解を更新することが望ましい。一方、限界観測数に余裕がなく解空間のサイズも膨大である場合には、精度をある程度犠牲にしても解空間全体から評価値が高いと予測される解を探し出したい。あまりにも限界観測数が少なくテストが困難な場合は探索 자체をあきらめ、テストする変数の削減やテスト期間の延長をサイト開発者に促す決定もありうる。

そこで提案手法では局所探索法による解探索を行う前に戦略判断を行う。与えられた限界観測数と解空間のサイズから、解空間全体からの最適解の推定および解の評価のために適切なサ

ンプルサイズを算出する。限界観測数に余裕がない場合はすべての観測数を最適解の推定に割き、仮定する予測モデルによっては精度が落ちる可能性もあるが、ある程度現状を改善することができる解を予測する。限界観測数に余裕がある場合は、予測された最適解を初期解として局所探索を行い、予測モデルで見逃されてしまった最適解を探索する。戦略判断は最適解の推定と局所探索の最適な分岐点を決定するためのものである。以上が提案手法のコンセプトである（図 4.1 参照）。

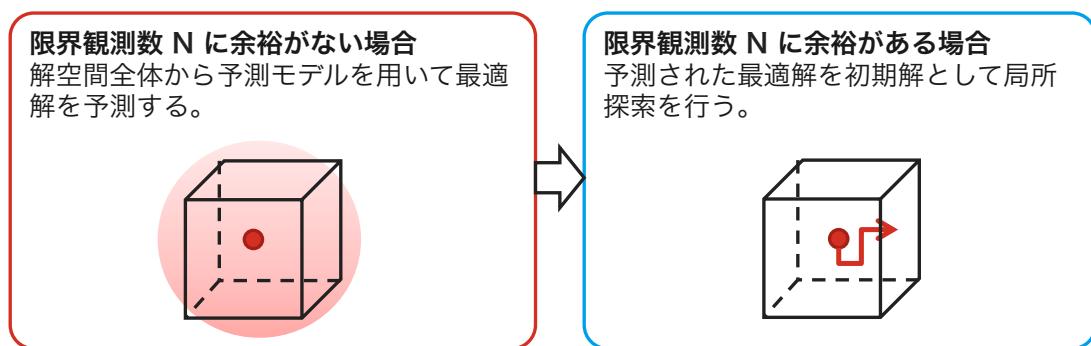


図 4.1 提案手法のコンセプト

提案手法は「戦略判断フェーズ」「初期解決定フェーズ」「解探索フェーズ」の 3 つのフェーズで構成されている。

戦略判断フェーズでは与えられた限界観測数、解空間のサイズからその後の初期解決定フェーズおよび解探索フェーズに用いるサンプルサイズを算出する。

初期解決定フェーズでは解空間全体から収集された観測値から予測モデルに基いて最適解を予測する。

解探索フェーズでは初期解決定フェーズで推定した最適解を初期解として局所探索を行い、残りの観測回数でさらに評価値が高い解の探索を行う。

4.1.1 戦略判断フェーズ

戦略判断フェーズでは解空間のサイズから、以下のサンプルサイズを算出する。

- 初期解決定フェーズに必要最低限なサンプルサイズ n_0
- 初期解決定フェーズで最適解を予測するのに十分なサンプルサイズ n_1
- 解探索フェーズにおける解の評価に十分なサンプルサイズ n_2

初期解決定フェーズに必要最低限なサンプルサイズ n_0 とは、解空間全体から最適解を予測するのに最低限必要な観測回数のことである。最適解を予測するためには、最低でも各変数が取りうるすべての値について一度は観測する必要がある。したがって、変数 x_i が取りうる値の数を l_i とすると、初期解決定フェーズに必要最低限なサンプルサイズは式 4.1 のように求めることができる。

$$n_0 = \sum_{i=1}^m l_i \quad (4.1)$$

初期解決定フェーズで最適解を予測するのに十分なサンプルサイズ n_1 は、各変数が取る値によって評価値に差が生じることを評価するのに十分な大きさが求められる。ある変数が取る値によって評価値の平均値が変化することを検定する方法のひとつに、分散分析がある。ここでは分散分析に必要なサンプルサイズを、初期解決定フェーズに用いるサンプルサイズ n_1 として採用することにする。

分散分析で用いるサンプルサイズは統計的検定に関するパラメータである有意水準 α 、検出力 β 、効果量 Δ に加えて群の数 k を与えることで求めることができる。予測のためには各変数のすべての値について観測できれば良いので、 $k = \sum_{i=1}^m l_i$ として算出する。

解探索フェーズにおける解の評価に十分なサンプルサイズ n_2 は 3.3.3 項で説明した推測統計学において 2 群の平均値の差の検定を行われるために用いられる t 検定で要するサンプルサイズを利用するすることにする。

このようにして算出される各サンプルサイズ n_0, n_1, n_2 と限界観測数 N から、その後の探索戦略を判断する。限界観測数 N が初期解決定フェーズに必要最低限なサンプルサイズ n_0 に満たない場合は、最適化を断念する。限界観測数 N が n_0 を上回るが初期解決定フェーズで最適解を予測するのに十分なサンプルサイズ n_1 に満たない場合は、限界観測数 N すべてを初期解決定フェーズに割く。限界観測数 N が n_1 を上回る場合は初期解決定フェーズの後に解探索フェーズに入り、局所探索法を用いてより良い解の探索を行う。

4.1.2 初期解決定フェーズ

初期解決定フェーズでは、まず解空間全体から無作為に抽出した解 $x \in X$ から観測データ Y を得る。観測回数が限界観測数 N または初期解決定フェーズに用いるサンプルサイズ n_1 に到達した時点で、観測データ Y を活用して最適解 x^* の予測を行う。

4.1.3 解探索フェーズ

解探索フェーズでは、初期解決定フェーズで予測された最適解 x^* を初期解として局所探索を行う。初期解決定フェーズで適用した予測モデルが当てはまらなかつたために見逃されてしまった、より評価値の高い解を探索することを目指す。

4.2 観測データの活用

ウェブページ最適化問題では観測データを活用することで、さらに効率的な探索が可能になる。観測データをもとに解空間の特性について予測を立て、より有望な解がありそうなところに効率的に観測数を割くことができれば、少ない観測数でもより良い解を探索することができるようになる。

観測データからその解空間の特性について予測する手法として、機械学習が広く用いられている。ウェブページ最適化問題でも機械学習を用いることで、観測データから解空間の有用な規則やルールを抽出することができる。ウェブページ最適化問題で機械学習を用いる場合の入力データは、解 x が素性ベクトル、観測値 y がラベルとなる。つまり、式 4.2 で表される訓練データを学習器に与えることで解析を行う。

$$\{(x_1, y_1), (x_2, y_2), \dots\} \quad (4.2)$$

機械学習の主な活用例としてひとつは回帰分析がある。ここでは回帰分析の例として重回帰分析を取り上げる。重回帰分析は被説明変数に説明変数が与える影響を定量的に分析するための手法である。重回帰分析では、まず被説明変数と説明変数の関係のモデルを回帰式として表現する。その後、回帰式中の各説明変数にかかる偏回帰係数と定数項を求めることで、各説明変数の影響度を分析する。偏回帰係数と定数項は最小二乗法によって求めことが多い。

探索中に観測データに重回帰分析を用いることで、未観測の解についても回帰式に基いて評価値を予測することができるようになる。あらゆる解の評価値が予測できれば、探索が局所最適解に陥った際に評価値が高いと予測される解にジャンプしたり、偏回帰係数に基づいて近傍解を取る方向および順番に重み付けをするといった工夫が考えられる。

3.4 節で説明した、評価値を最大化する変数値の組み合わせを最適解とする「線形モデルによる最適解の予測」工夫も重回帰分析を用いた観測データの活用例のひとつとして捉えることができる。回帰分析の方法には他にもロジスティック回帰やサポートベクター回帰などの手法があり、被説明変数および説明変数のタイプやデータの性質によって使い分けることが多い。

回帰分析以外の観測データの活用方法としてベクトルの類似度を用いる方法も考えられる。ウェブページ最適化問題における解 x は m 次元のベクトルなので、コサイン類似度やシンプソン係数などによって解の間の類似度を定義することができる。「よい解同士は似通った構造を持っている」というヒューリスティックな仮定を置けば、暫定解と類似度が大きい解を優先して探索を行うといった工夫が可能になる。これは類似度を用いて近傍解を定義することに相当する。

他には回帰分析や分散分析によって算出された各変数の影響度によって探索の優先順位の重み付けを行う工夫も考えられる。このようにして探索空間を絞り込んでいくことで、与えられた限界観測数 N を効率的に評価値が高い解の探索に割くことができるようになる。

3.3.2 項で説明したレーシングアルゴリズムも、探索空間を絞り込むためのひとつの工夫である。レーシングアルゴリズムでは解 x の空間的な情報を用いず解の期待値のみで評価を行なっているが、上述したような工夫を用いることで、さらに効率的な探索が可能になるとされる。

つまりウェブページ最適化問題では、逐次的に観測データの解析を行うことによって探索を効率化することができる。提案手法では、初期解決定フェーズおよび解探索フェーズにこれらの工夫を用いることで、より良い探索を実現する。

4.3 アルゴリズム

4.3.1 事前準備

ウェブページ最適化問題で定義される解空間 X と限界観測数 N の他に、各フェーズで用いるサンプルサイズを算出するために統計的検定に関するパラメータである有意水準 α 、検出力 $1 - \beta$ 、効果量 Δ を渡す。有意水準 α は帰無仮説を棄却する正確さを表すものである。検出力 $1 - \beta$ は比較する解に差がある時に、差があると検出することができる確率を表すものである。効果量 Δ は平均値の差を標準化したものである。適用する分野によって異なるが、 $\alpha = 0.05$ 、 $\beta = 0.20$ と設定することが望ましいとされている [38]。検定を適用する領域や問題によって検出を求められる効果量 Δ は異なるため明確な基準は無いが、行動科学の分野では Cohen によって効果量の大きさの基準が示されている [39]。

まず提案アルゴリズムは戦略判断フェーズに入り、与えられたパラメータから 4.1.1 項で説明した各サンプルサイズ n_0, n_1, n_2 を算出する (Algorithm 2 行 2~4 参照)。 $N < n_0$ の場合は最適解の予測が不可能と判断し、最適化の実行を断念する (Algorithm 2 行 6 参照)。

Algorithm 2 提案アルゴリズムの全体

Require: α as the significant level.

Require: $1 - \beta$ as the power.

Require: Δ as the effect size.

Require: $N > 0$ as the number of visitors.

Require: X as the set of solutions.

- 1: Set Y as an empty set for observed data.
- 2: Set $n_0 \leftarrow \sum_{i=1}^m l_i$
- 3: Set $n_1 \leftarrow n_{ANOVA}(\alpha, \beta, \Delta, X)$
- 4: Set $n_2 \leftarrow n_{TTEST}(\alpha, \beta, \Delta)$
- 5: **if** $N < n_0$ **then**
- 6: Give up optimization.
- 7: $\mathbf{x}^*, N, Y \leftarrow Initialization(N, X, Y, n_1)$
- 8: **while** $N > 0$ **do**
- 9: $\mathbf{x}^*, N, Y \leftarrow Move(\mathbf{x}^*, N, Y, n_2)$

- 10: **return** \mathbf{x}^* as optimal pattern.
-

$N \geq n_0$ の場合は初期解決定フェーズに限界観測数 N 、解候補 X 、そして 4.1.1 節で算出したサンプルサイズ n_1 を渡し、観測値の収集および最適解 x^* の推定を行う (Algorithm 2 行 7 参照)。初期解決定フェーズ完了時点でまだ観測数が余っていた場合は次の解探索フェーズに移り、 $N > 0$ の間、探索を行う。解探索フェーズの解比較には 4.1.1 節で算出したサンプルサイズ n_2 が用いられる。解探索フェーズでは観測回数がある限り解の比較および移動を繰り返し、最適解 x^* を更新する (Algorithm 2 行 9 参照)。

アルゴリズムの終了条件は限界観測数 N を使い果たすことであり、 $N = 0$ となった時点で探索が終了して最適解 x^* が出力される (Algorithm 2 行 10 参照)。以上が提案手法のアルゴリズムの概観である。

4.3.2 ベースラインアルゴリズム A: 単純総当たり

次に、ベースラインとなるアルゴリズムについて説明する。ベースラインアルゴリズム A (単純総当たり) は、提案手法と異なり観測データを活用した初期値決定、ヒューリスティクスを用いた探索のいずれも用いない探索手法である。単純総当たりは解空間の中から無作為に解 $x \in X$ を選択して観測を行う。観測値が蓄積された観測データ Y から、最も期待値が高いも

のを最適解 \mathbf{x}^* として出力する (Algorithm 3 参照)。

Algorithm 3 ベースラインアルゴリズム A:単純総当たり

Choose $\mathbf{x} \in X$ randomly.

$y \leftarrow \text{Observe}(\mathbf{x})$

Add (\mathbf{x}, y) to Y .

$\mathbf{x}^* \leftarrow \arg \max_{\mathbf{x} \in X} \mathbb{E}[y|\mathbf{x}]$

return \mathbf{x}^*

4.3.3 ベースラインアルゴリズム B:単純山登り

ベースラインアルゴリズム B (単純山登り) では、提案手法と異なり観測データを活用した初期値決定を行わないが、無作為に抽出した初期解から局所探索法を用いて探索を行う。今回は、シンプルな局所探索法の実装として山登り法を用いることにする。暫定解と近傍解の比較にはサンプルサイズ n_2 の観測数を割く。その後暫定解と近傍解の平均値の比較を行い、移動するか否かを判断する (Algorithm 4 参照)。

Algorithm 4 ベースラインアルゴリズム B:単純山登り

◊ 初期解決定フェーズ: ランダム選択

function INITIALIZATION(N, X, Y, n_1)

Choose $\mathbf{x} \in X$ randomly.

return \mathbf{x}, N, Y

◊ 探索フェーズ: 山登り探索

function MOVE(\mathbf{x}^*, N, Y, n_2)

$X' \leftarrow \{\mathbf{x}^*\} \cup \text{Neighbors}(\mathbf{x}^*)$

for $n = 1$ to n_2 **do**

for $\mathbf{x} \in X'$ **do**

$y \leftarrow \text{Observe}(\mathbf{x})$

 Add (\mathbf{x}, y) to Y .

$N \leftarrow N - 1$

$\mathbf{x}^* \leftarrow \arg \max_{\mathbf{x} \in X'} \mathbb{E}[y|\mathbf{x}]$

return \mathbf{x}^*, N, Y

4.3.4 提案アルゴリズム A: 観測データを活用した初期解決定

提案アルゴリズム A では、観測データの活用を用いて初期解の決定を行い、その後局所探索法によって最適解を探索する。ベースラインアルゴリズムと異なり初期解を決定するために観測数を消費するが、線形モデルによって評価値が高いと予測される初期解から探索を開始することができる。初期解決定フェーズでは観測データを活用して、「線形モデルによる最適解の予測」を用いる。すなわち、各変数について評価値の期待値を最大化する組み合わせを最適解として採用する。解探索フェーズではベースラインアルゴリズム B と同様に山登り法を用いる。観測データを用いた予測によって初期解を決定するアルゴリズムを Algorithm 5 に示す。

Algorithm 5 提案アルゴリズム A: 観測データの活用を導入した初期解決定フェーズ

◇ 初期解決定フェーズ: 観測値の活用

```

function INITIALIZATION( $N, X, Y, n_1$ )
    Set  $\mathbf{x}^*$  as empty vector.
    for  $n = 1$  to  $n_1$  do
        Choose  $\mathbf{x} \in X$  randomly.
         $y \leftarrow Observe(\mathbf{x})$ 
        Add  $(\mathbf{x}, y)$  to  $Y$ .
         $N \leftarrow N - 1$ 
    for  $i = 1$  to  $m$  do
         $v^* \leftarrow \arg \max_{v \in V_i} \mathbb{E}[y|x_i = v]$ 
         $x_i^* \leftarrow v^*$ 
    return  $\mathbf{x}^*, N, Y$ 

```

提案アルゴリズム A では、まず n_1 回無作為に解 $\mathbf{x} \in X$ を選び、観測データ Y を収集する。観測データから解を構成するそれぞれの変数 x_i について条件付き期待値 $\mathbb{E}[y|x_i = v]$ を最大にする変数値 $v^* \in V_i$ を求め、その値の組み合わせ \mathbf{x}^* を局所探索の初期解とする。

4.3.5 提案アルゴリズム B: 観測の短縮の導入

次に、提案アルゴリズム A をベースにして「観測の短縮」工夫を導入する。ここでは信頼区間を用いた統計的推定によって解の評価を行う。解の評価を行う際、勝者解はその下側信頼限界が他の全ての解の上側信頼限界よりも高い解として定義される（式 3.7 参照）。

初期解決定フェーズでは各変数の最適な値を算出する際に観測の短縮を用いる。各変数につ

いて他の変数値のすべての上側信頼限界を上回る下側信頼限界を持つ変数値があれば、その値を最適解を構成する変数値として固定する。すべての変数についてそのような変数値が発見され、最適解を構成する変数値がすべて埋められた時点で観測を終了し最適解を出力する。観測回数 n_1 を到達する前に最適解を推定できなかった場合はアルゴリズム A と同様に、各変数で最も評価値の期待値が高い値を組み合わせたものを最適解として出力する (Algorithm 6 参照)。

Algorithm 6 提案アルゴリズム B1: 観測の短縮を導入した初期解決定フェーズ

◇ 初期解決定フェーズ: 観測の短縮

function INITIALIZATION(N, X, Y, n_1)

Set \mathbf{x}^* as empty vector.

for $n = 1$ to n_1 **do**

Choose $\mathbf{x} \in X$ randomly.

$y \leftarrow \text{Observe}(\mathbf{x})$

Add (\mathbf{x}, y) to Y .

$N \leftarrow N - 1$

for $i = 1$ to m **do**

$v^* \leftarrow \arg \max_{v \in V_i} \mathbb{E}[y|x_i = v]$

if $\mathbb{E}[y|x_i = v^*]$ is winner **then**

$x_i^* \leftarrow v^*$

if every element of \mathbf{x}^* is filled **then**

return \mathbf{x}^*, N, Y

for $i = 1$ to m **do**

$v^* \leftarrow \arg \max_{v \in V_i} \mathbb{E}[y|x_i = v]$

$x_i^* \leftarrow v^*$

return \mathbf{x}^*, N, Y

解探索フェーズにおいては暫定解と近傍解の評価に観測の短縮を導入する。暫定解および近傍解の中に勝者解が見つかればその時点で観測を終了し、最適解を出力する (Algorithm 7 参照)。

Algorithm 7 提案アルゴリズム B2: 観測の短縮を導入した解探索フェーズ

◇ 解探索フェーズ: 観測の短縮

```

function MOVE( $\mathbf{x}^*, N, Y, n_2$ )
     $X' \leftarrow \{\mathbf{x}^*\} \cup \text{Neighbors}(\mathbf{x}^*)$ 
    for  $n = 1$  to  $n_2$  do
        for  $\mathbf{x} \in X'$  do
             $y \leftarrow \text{Observe}(\mathbf{x})$ 
            Add  $(\mathbf{x}, y)$  to  $Y$ .
             $N \leftarrow N - 1$ 
            if  $\mathbb{E}[y|\mathbf{x}]$  is winner then
                 $\mathbf{x}^* \leftarrow \mathbf{x}$ 
            return  $\mathbf{x}^*, N, Y$ 
     $\mathbf{x}^* \leftarrow \arg \max_{\mathbf{x} \in X'} \mathbb{E}[y|\mathbf{x}]$ 
    return  $\mathbf{x}^*, N, Y$ 

```

4.3.6 提案アルゴリズム C: 敗者解のカットオフの導入

次に、提案アルゴリズム B をベースにして「敗者解のカットオフ」工夫を導入する。解の評価中に敗者解を発見した時点で敗者解を観測対象の近傍解から取り除くことで、残った有望な近傍解に観測数を集中させて高速化を図る。敗者解はその上側信頼限界が、ある解の下側信頼限界を下回る解である（式 3.6 参照）。

初期解決決定フェーズでは各変数の変数値の比較に敗者解のカットオフを用いる。各変数についてある変数値の下側信頼限界を下回る上側信頼限界を持つ変数値があれば、その変数値を持つ解すべてを解空間 X から取り除く。このようにして有望な解にできるだけ多くの観測数を割くようにし、最も評価値の高い変数値の発見を高速化する（Algorithm 8 参照）。

解探索フェーズでは敗者解が見つかった際に敗者解を近傍解 X' から取り除くことで、有望な解に観測数を割くようにする（Algorithm 9 参照）。

4.4 本章のまとめ

本章ではウェブサイトの限界観測数と解空間のサイズに応じて探索を行う手法を提案した。提案手法は戦略判断フェーズ、初期解決決定フェーズ、解探索フェーズの 3 つのフェーズからなる手法である。

Algorithm 8 提案アルゴリズム C1: 観測の短縮と敗者解のカットオフを導入した初期決定フェーズ

◇ 初期決定フェーズ: 観測の短縮 + 敗者解のカットオフ

function INITIALIZATION(N, X, Y, n_1)

 Set \mathbf{x}^* as vector.

for $n = 1$ to n_1 **do**

 Choose $\mathbf{x} \in X$ randomly.

$y \leftarrow \text{Observe}(\mathbf{x})$

 Add (\mathbf{x}, y) to Y .

$N \leftarrow N - 1$

for $i = 1$ to m **do**

if $\mathbb{E}[y|x_i = v]$ is loser **then**

$X \leftarrow \{\mathbf{x}|x_i \neq v\}$

if $\mathbb{E}[y|x_i = v]$ is winner **then**

$x_i^* \leftarrow v$

if every element of \mathbf{x}^* is filled **then**

return \mathbf{x}^*, N, Y

for $i = 1$ to m **do**

$v^* \leftarrow \arg \max_{v \in V_i} \mathbb{E}[y|x_i = v]$

$x_i^* \leftarrow v^*$

return \mathbf{x}^*, N, Y

戦略判断フェーズでは与えられた限界観測数、解空間のサイズおよび統計的検定の精度を表すパラメータから、初期決定フェーズに必要最低限なサンプルサイズ n_0 、初期決定フェーズで最適解を予測するのに十分なサンプルサイズ n_1 、解探索フェーズにおける解の評価に十分なサンプルサイズ n_2 の 3 つを算出する。これによって「最適化を断念する」「初期決定フェーズにすべての限界観測数を割いて最良解を予測する」「初期決定フェーズの後に解探索フェーズで更に良い解を探索する」のいずれかの探索戦略が決定される。

初期決定フェーズでは与えられたサンプルサイズ n_1 を割いて解空間 X 全体の観測を行い最適解の予測を行う。ここでは線形モデルを仮定して予測を行う場合を説明した。

解探索フェーズでは初期決定フェーズ終了後余った観測回数を割いて、局所探索法によつて更に良い解の探索を行う。ここでは山登り探索を用いて探索を行う場合を説明した。

最後に本章で説明した各アルゴリズムを表 4.1 に整理する。

Algorithm 9 提案アルゴリズム C2: 観測の短縮と敗者解のカットオフを導入した解探索フェーズ

◇ 解探索フェーズ: 観測の短縮 + 敗者解のカットオフ

```

function MOVE( $\mathbf{x}^*, N, Y, n_2$ )
     $X' \leftarrow \{\mathbf{x}^*\} \cup Neighbors(\mathbf{x}^*)$ 
    for  $n = 1$  to  $n_2$  do
        for  $\mathbf{x} \in X'$  do
             $y \leftarrow Observe(\mathbf{x})$ 
            Add  $(\mathbf{x}, y)$  to  $Y$ .
             $N \leftarrow N - 1$ 
            if  $\mathbb{E}[y|\mathbf{x}]$  is loser then
                Pop out  $\mathbf{x}$  from  $X'$ .
            if  $\mathbb{E}[y|\mathbf{x}]$  is winner then
                 $\mathbf{x}^* \leftarrow \mathbf{x}$ 
    return  $\mathbf{x}^*, N, Y$ 
 $\mathbf{x}^* \leftarrow \arg \max_{\mathbf{x} \in X'} \mathbb{E}[y|\mathbf{x}]$ 
return  $\mathbf{x}^*, N, Y$ 

```

表 4.1 提案アルゴリズムの整理

名前	初期値決定	解探索
ベースライン A	ランダム	総当たり
ベースライン B	ランダム	山登り
提案 A	線形モデル予測	山登り
提案 B	線形モデル予測 + 観測の短縮	山登り + 観測の短縮
提案 C	線形モデル予測 + 観測の短縮 + 敗者解のカットオフ	山登り + 観測の短縮 + 敗者解のカットオフ

第 5 章

実験

本章では、第 4 章で提案したアルゴリズムを評価するための実験およびその結果について述べる。5.1 節では著者が独自に設定した評価関数によってシミュレーションを行なう。最適化のために与えられる観測数や評価関数の特性を変化させ、提案アルゴリズムがそれらの状況に応じて適切に最適化を行うことを評価する。5.2 節では、一日数万ユーザが訪れる実際の大規模ウェブサイトでウェブページ最適化を行なった際に観測されたデータに基いて評価関数を設計し、シミュレーションを行なう。大規模ウェブサイトで提案手法が機能することを評価する。5.3 節では、提案手法に基いてウェブページ最適化を行う最適化プログラムを実装し、一日数百ユーザが訪れる実際の中規模ウェブサイトに導入する。提案手法が実際に最適化プログラムという形で機能すること、中規模サイトでも有効であることを評価する。

5.1 仮想の評価関数によるシミュレーション実験

5.1.1 問題設定

仮想状況におけるシミュレーションでは、表 5.1 に記す問題を独自に設定し、それぞれに対してベースラインアルゴリズムと提案アルゴリズムを適用する。評価関数 $f(x)$ の特性は評価関数に非線形性が含まれるか否かを表す。ここでいう非線形性とは、 x_1x_2 のように、異なる変数を掛けあわせた項が評価関数に含まれることを指す。誤差 ϵ は評価関数 $f(x)$ の値に附加される誤差の有無、すなわち観測値 y のばらつきの有無を表す。限界観測数 N は最適化のために用いることができる観測回数を表す。サンプルサイズ n_0 は最適化を行うために最低限必要なサンプルサイズであり、式 4.1 によって求められる。サンプルサイズ n_1 は初期値決定フェーズに割くサンプルサイズであり、4.1.1 項で説明した分散分析のサンプルサイズ算出方

表 5.1 シミュレーションの問題設定

	評価関数 $f(x)$ の特性	誤差 ϵ	限界観測数 N
問題 0	線形	$\epsilon = 0$	$n_0 < N < n_1$
問題 1	線形	$\epsilon > 0$	$n_0 < N < n_1$
問題 2	線形	$\epsilon > 0$	$n_1 \leq N$
問題 3	非線形	$\epsilon > 0$	$n_0 < N < n_1$
問題 4	非線形	$\epsilon > 0$	$n_1 \leq N$

法によって求められる。表 5.1 に記した問題に対して 4.3 節で説明した探索アルゴリズムを適用する。

仮想状況におけるシミュレーションでは変数の数 $m = 5$ 、それぞれの変数 x_i が取りうる値の数 $l_i = 3$ の状況を想定する。したがって、 $M = l^m = 3^5 (= 243)$ 通りの解候補が生成される。

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\} = \{(0, 0, 0, 0, 0), (0, 0, 0, 0, 1), \dots, (2, 2, 2, 2, 2)\}$$

この中から限界観測数 N の中で最適解 \mathbf{x}^* を探索することが解くべき問題となる。統計的検定にまつわるパラメータは

$$\alpha = 0.05, 1 - \beta = 0.8, \Delta = 0.2$$

と設定した。また、ここから算出されるサンプルサイズは以下のようになる。

$$n_0 = 5 \times 3 = 15, n_1 = 500, n_2 = 400$$

いずれも統計解析ソフト R の pwr パッケージ^{*1}を用いて算出した。

今回の実験では各アルゴリズムで最適化を開始した後、各時点において暫定解が最適解である確率（正確度と呼ぶ）を用いて各アルゴリズムのパフォーマンスを評価する。シミュレーションでは各解の真の評価値がわかっているため、暫定解 x が最適解 \mathbf{x}^* である確率を表す正確度を指標とした。それぞれ実験結果を示す折れ線グラフは 100 回シミュレーションを繰り返した時の平均値を示している。折れ線グラフに付随するエラーバーは標準誤差を表わしている。

^{*1} CRAN - Package pwr <http://cran.r-project.org/web/packages/pwr>

5.1.2 実験結果

問題 0 (線形、 $\epsilon = 0$ 、 $n_0 < N < n_1$)

最初に、問題 0 に対してベースライン A、ベースライン B そして提案アルゴリズム A を適用した結果を図 5.1 に示す。横軸は消費した観測回数 t の推移、縦軸は各時点における正確度を表している。

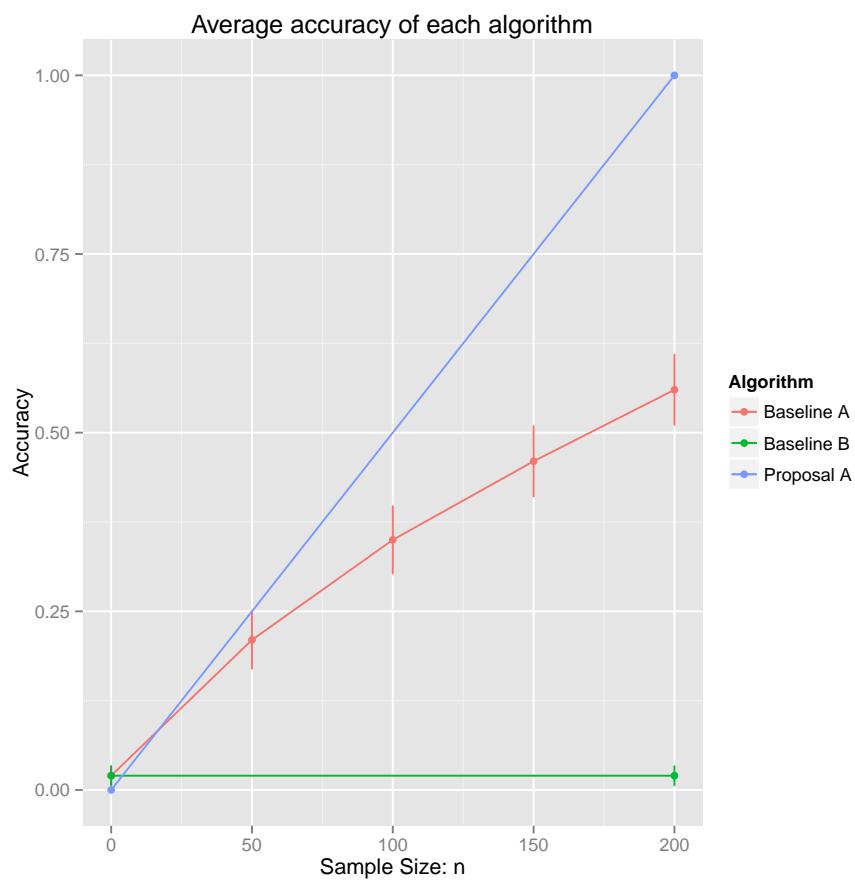


図 5.1 問題 0 における提案アルゴリズム A の正確度

ベースラインアルゴリズム A(単純総当たり)では、解空間 X 全体からランダムに解 x を抽出して観測値を収集し、観測データの中で最も期待値が高いものを最適解とする。観測回数を経るごとに最適解を抽出する確率が高くなるため、正確度が増加する。

ベースラインアルゴリズム B(単純山登り)では、暫定解と近傍解の比較にサンプルサイズ n_2 を割くことで着実に解を更新することができるが、限界観測数 N が十分でない場合は解の更新を十分に繰り返すことができず、低い正確度にとどまってしまう。

グラフ右上の青点で示される提案アルゴリズム A は評価関数が線形かつ誤差が含まれないため、確実に最適解を予測することができる。

問題 1 (線形、 $\epsilon > 0$ 、 $n_0 < N < n_1$)

次に、問題 1 に対してベースライン A、ベースライン B そして提案アルゴリズム A を適用した結果を図 5.2 に示す。問題 1 では問題 0 と異なり評価値に誤差 ϵ が入り込むため、最も観測データの平均値が高い解が最適解とは限らなくなる。そのため、先ほど着実に正確度を高めことが出来たベースラインアルゴリズム A の観測終了時点での正確度が評価値に誤差 ϵ が含まれない問題 0 の時と比べると低くなっている。

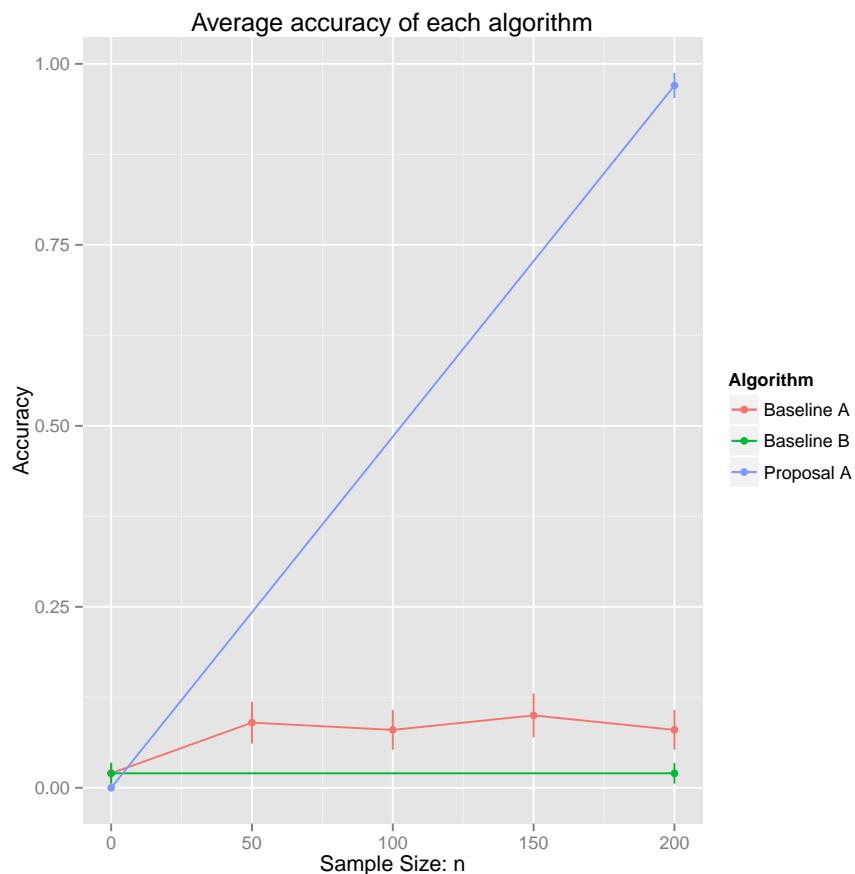


図 5.2 問題 1 における提案アルゴリズム A の正確度

一方でベースライン B アルゴリズムの正確度の增加分は問題 0 のときと大きな差がない。二つの解の比較に十分な観測数を割いた上で最適解の判断を行なっているため問題 0 のときと同様、着実に正確度を向上させることができている。しかし、使用できる観測回数が少ない場合はほとんど解の更新を行うことが出来ないため、結果的に低い正確度にとどまっている。

右上の青点で示された提案アルゴリズム A では線形モデルを仮定して最適解の予測を行なうため、非線形性の無い評価関数については高い精度で最適解を予測することができる。同じ観測回数であっても、ベースラインアルゴリズムと比較して高い正確度を達成することができる。

問題 2 (線形、 $\epsilon > 0$ 、 $n_1 \leq N$)

問題 2 に対してベースライン A、ベースライン B そして提案アルゴリズム A を適用した結果を図 5.3 に示す。問題 2 は問題 1 に比べて大きな限界観測数 N が与えられた場合を想定している。 $t > 500$ の探索フェーズに入り暫定解との比較を行なっても、提案アルゴリズム A は高い正確度を保っていることがわかる。評価関数に非線形性が無い限りは、線形モデルによる予測によって高い正確度で最適解を予測することができる。

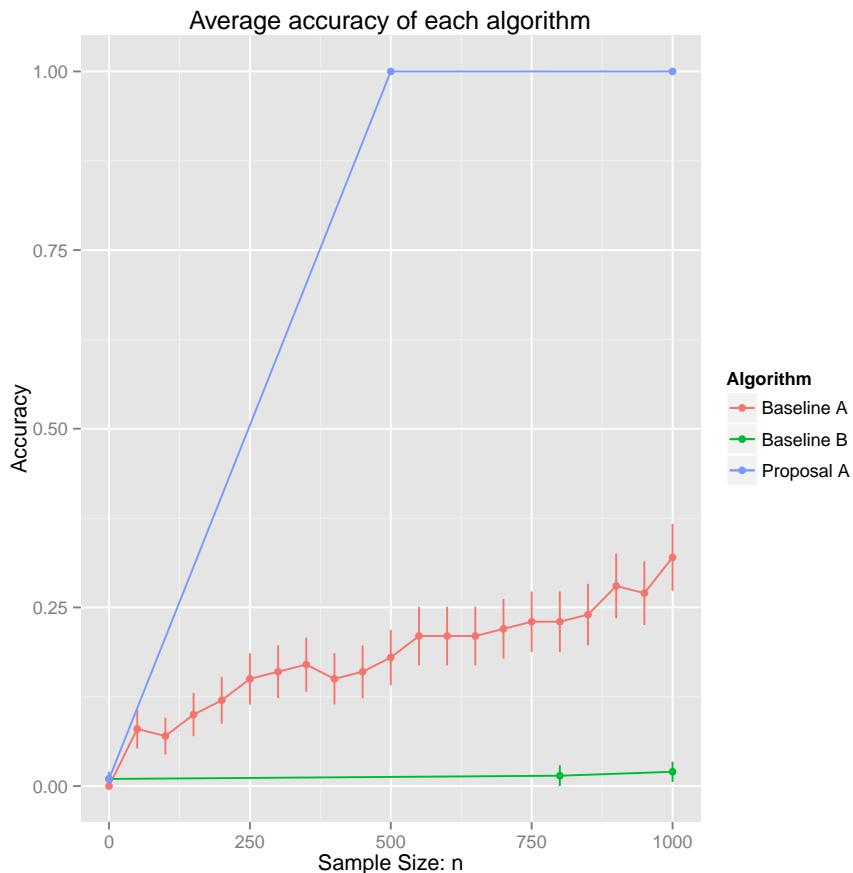


図 5.3 問題 2 における提案アルゴリズム A の正確度

問題 3 (非線形、 $\epsilon > 0$ 、 $n_0 < N < n_1$)

問題 3 に対してベースライン A、ベースライン B そして提案アルゴリズム A を適用した結果を図 5.4 に示す。問題 3 は限界観測数 N が少なく、評価関数に非線形性が含まれる場合を想定している。非線形性が強くなると線形モデルを仮定した最適解の予測の精度が落ちてしまう。そのため、提案アルゴリズム A の限界観測数 N 終了時の正確度はベースライン A と同程度まで落ちてしまう。提案アルゴリズム A は非線形性が強い目的変数に対しては力を発揮できない。

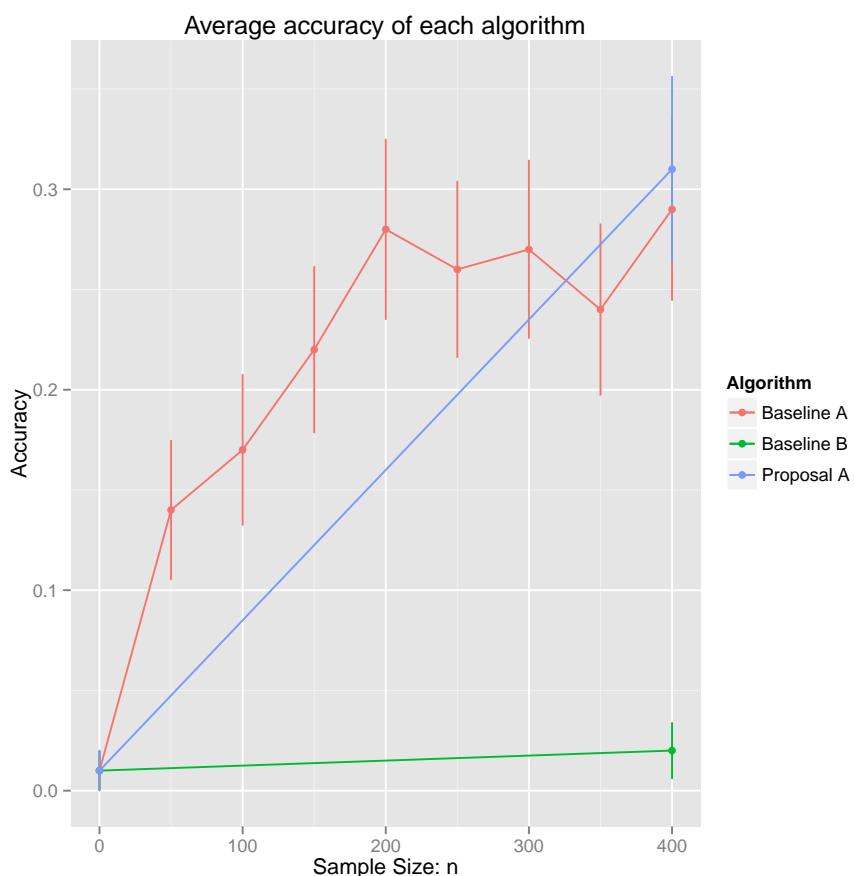


図 5.4 問題 3 における提案アルゴリズム A の正確度

問題 4 (非線形、 $\epsilon > 0$ 、 $n_1 \leq N$)

次に、問題 4 に対してベースライン A、提案アルゴリズム A、そして観測の短縮を導入した提案アルゴリズム B を適用した結果を図 5.5 に示す。問題 4 は問題 3 に比べて大きい限界観測数 N が与えられた場合を想定している。提案アルゴリズム A、提案アルゴリズム B ともに

$N = n_1$ 時点では正確度が 0.3 程度にとどまっているが、その後山登り探索を行うことで正確度を向上している。

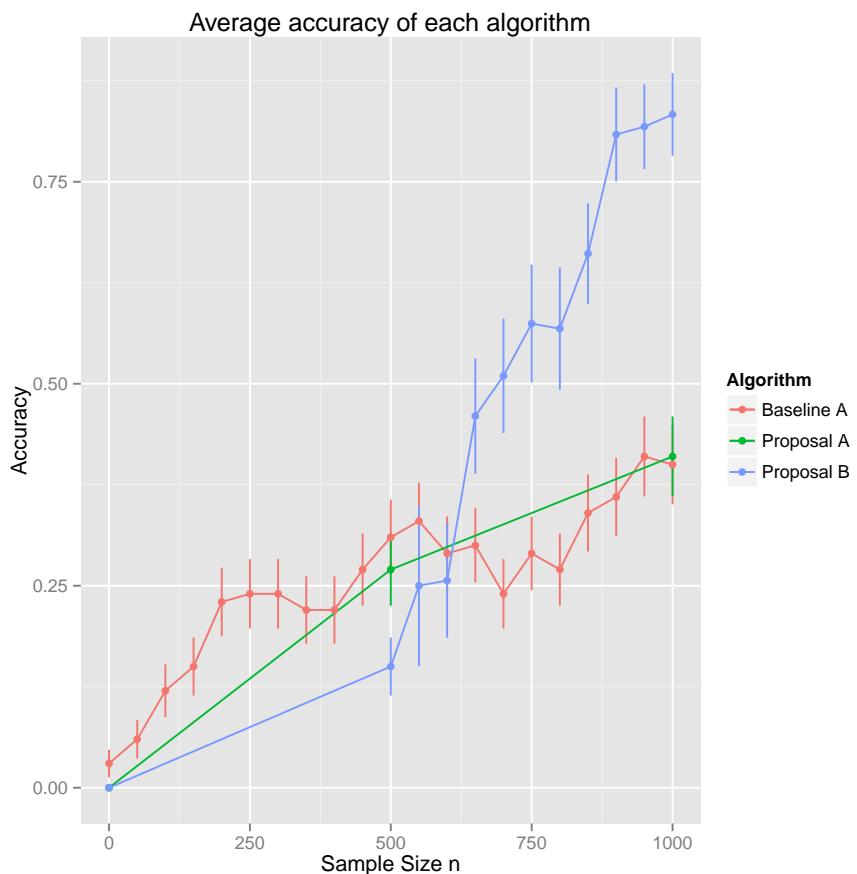


図 5.5 問題 4 における提案アルゴリズム A と提案アルゴリズム B の正確度

提案アルゴリズム A は山登りによる解の改善の速度が遅く、ベースラインアルゴリズム A とくらべても大きな改善が見られない。一方で、観測の短縮を導入した提案アルゴリズム B は解の比較の際に有意差が見られた時点で解移動を行う観測の短縮によって、高速に解の更新を行うことに成功している。

最後に、問題 4 に対して提案アルゴリズム C を適応した結果を図 5.6 に示す。提案アルゴリズム C は観測の短縮に加え、敗者解を観測対象から取り除く敗者解のカットオフを導入することでさらなる解探索の高速化を実現している。

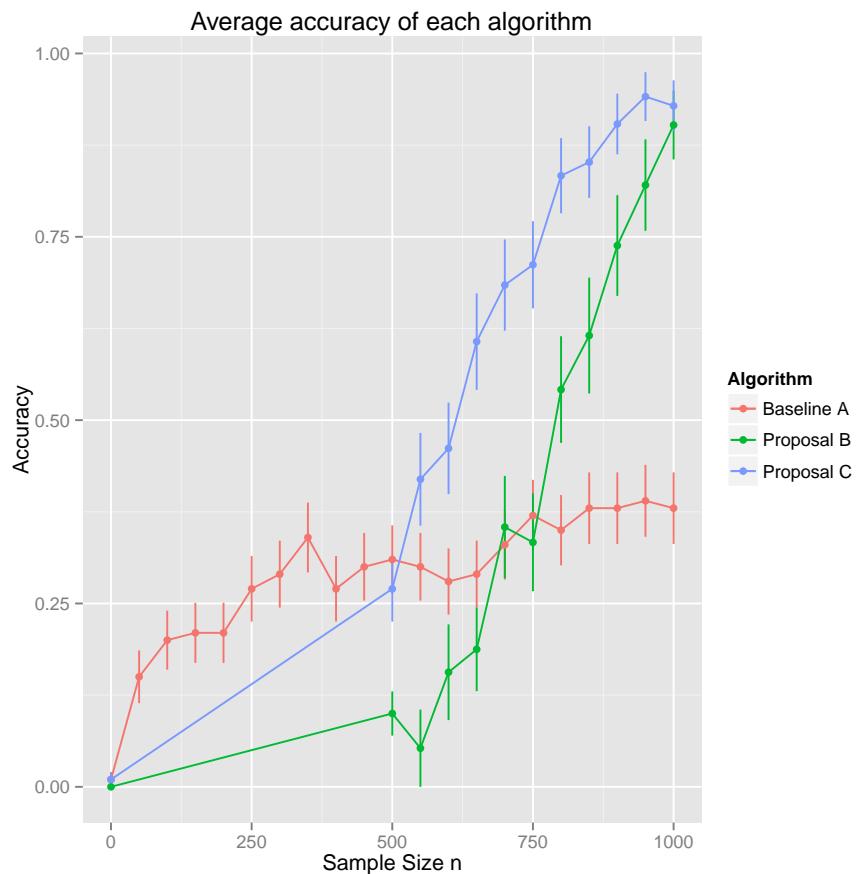


図 5.6 問題 4 における提案アルゴリズム B と提案アルゴリズム C の正確度

5.1.3 まとめ

本節では、著者が独自に設定した仮想の評価関数についてシミュレーションを行なった。評価関数には線形または非線形の特性を持たせ、誤差を含む場合と含まない場合の両方についてシミュレーションを行なった。また、初期値決定フェーズにおいて最適解の予測を行うのに十分な観測回数が与えられる場合と与えられない場合についてもシミュレーションを行い、様々な状況を想定した。

シミュレーションの結果、評価関数と初期値決定フェーズで仮定するモデルが適合する場合には、提案手法によって高い正確度で最適解を推定することができることがわかった。一方で評価関数が非線形性を含み、初期決定フェーズで仮定するモデルと当てはまらない場合には、初期値決定フェーズで最適解を予測する精度が落ち、ベースラインとなる単純な総当たり法と同程度のパフォーマンスになることがわかった。

初期値決定フェーズで高い精度で最適解を予測することが出来なかった場合でも、その後探索フェーズに入ることで、解の更新を行うことができる。提案手法に導入される観測の短縮やカットオフなどの工夫を導入することで、山登りを高速化することができる。予め評価関数に最適な効果量 Δ を設定することは難しいが、この工夫を導入することで必要以上に小さな効果量を設定してしまった場合でも、最適な速度で逐次的な解の更新を行うことができるようになる。

したがって、与えられた観測回数が少ないために初期値決定フェーズにしか観測数を割くことができず、かつ評価関数にもともと仮定していない非線形性が含まれる場合を除いて、提案手法を用いることで限界観測数および解空間のサイズに応じて最適化を行うことが可能であることが示された。

5.2 実際のウェブサイトで得られたデータによるシミュレーション実験

本節では、実際の大規模ウェブサイトでウェブページ最適化を行なった際に得られたデータから評価関数を設計し、シミュレーションを行う。

5.2.1 対象サイト

今回のシミュレーションでは、人物検索サイト「あのひと検索スパイシー^{*2}（以下実サイト A と呼ぶ）」で得たログデータをもとに評価関数を設定し、シミュレーションを行なった。実サイト A は人物を人名やキーワードによって検索し、その人物のプロフィールやつながりを閲覧することができるウェブサービスである。実サイト A は広告収益によって運営されており、ウェブページに広告を配置することによって収益を得ている。特に人物のつながりを表示する「相関図」ページは実サイト A のメインコンテンツである。実サイト A にとってはこのページの閲覧者に満足度の高いコンテンツを提供したうえで、広告に目を向けて興味を持つてもらえるようなウェブページのレイアウトや機能を追求することが増収上の課題となる（図 5.7 参照）。

実サイト A は一日の訪問者数が数万規模の大規模サイトである。実サイト A の訪問者はほとんどが検索エンジンの検索結果からの流入であり、以前に実サイト A に訪れたことのない新規ユーザが過半数を占めている。したがってほとんどの訪問者は「検索結果からはじめて来

^{*2} あのひと検索スパイシー <http://spysee.jp/>



図 5.7 実サイト A の相関図ページ

た」という文脈を共有していることになり、訪問者間の行動特性の差は小さいと考えられる。また、相関図ページ一枚がメインコンテンツであるため、相関図ページを訪問して広告をクリックするか否かという非常にシンプルな構図でウェブページのパフォーマンスを捉えることができる。

実サイト A では積極的にウェブページ最適化を開発に取り入れており、訪問者にデザインや機能が一部異なる相関図ページを表示することによって施策の価値を評価している。そのため、実サイト A ではそれぞれの施策が広告のクリック率に与える効果が蓄積されており、施策の有無によるクリック率の違いをシミュレートするための環境が整っている。以上の理由から対象サイトとして実サイト A を用いることが適切と判断し、シミュレーションを行うことにした。

5.2.2 問題設定

ウェブサイトの目的を顧みて、相関図ページにおける広告クリック率を最大化する評価指標として設定する。2013年5月14日から18日にかけて行なった実験で、相関図の広告の配置

を変えることによって広告のクリック率をテスト前のパターンに比べて 18% 引き上げることに成功した。同時期に他に 2 つの施策の実験も行なっていたが、これらの施策は逆にクリック率を減少させる方向に働いた。以上の施策による効果を表 5.2 にまとめる。「 χ^2 」列と「 p 値」列には同期間のオリジナルパターンによるクリック率と各施策を施したパターンによるクリック率でカイ二乗検定を行なった結果を掲載した。いずれの施策も p 値は 0.01 を下回っており、施策によって広告のクリック率に有意に差が生じたということができる。

表 5.2 実サイト A における各施策の効果

施策名	非クリック	クリック	クリック率	χ^2	p 値
施策 A	14,068	855	5.73%	9.34	2.24×10^{-3}
施策 B	9,002	737	7.57%	18.9	1.39×10^{-5}
施策 C	5,572	297	5.06%	16.3	5.44×10^{-5}
オリジナル	66,883	4,574	6.40%		

それぞれの施策を変数とし、各変数は施策を行うか否かの二値をとるものとする。この期間に得られたデータを基にすると、ある解 x を訪問者に表示した時にクリックされる確率は

$$q(x) = 0.064 - 0.0067x_1 + 0.0117x_2 - 0.0134x_3$$

と表される。ただし、評価関数 $f(x)$ は解 x に対して確率 $q(x)$ で 1 を、確率 $1 - q(x)$ で 0 を返すものとする。

シミュレーションに用いる各パラメータは以下のように設定する。

- 限界観測数: $N = 10000$
- シミュレーション回数: 100
- 有意水準: $\alpha = 0.05$
- 検出力: $1 - \beta = 0.8$
- 効果量: $\Delta = 0.05$

ここから算出される各サンプルサイズは以下のようになる。

- 初期値決定フェーズに用いるサンプルサイズ: $n_1 = 5500$
- 解探索フェーズの解の比較に用いるサンプルサイズ: $n_2 = 6500$

アルゴリズムは 5.1 節と同様に 4.3 節で説明したアルゴリズムを適用した。

5.2.3 実験結果

それぞれのアルゴリズムを適用した結果を図 5.8 に示す。ベースラインアルゴリズム A（単純総当たり）は解空間からランダムに解を抽出して観測を行い、最も期待値が大きい解を最適解とする。観測回数を経るごとに観測データが増え、着実に正確度を向上することに成功している。

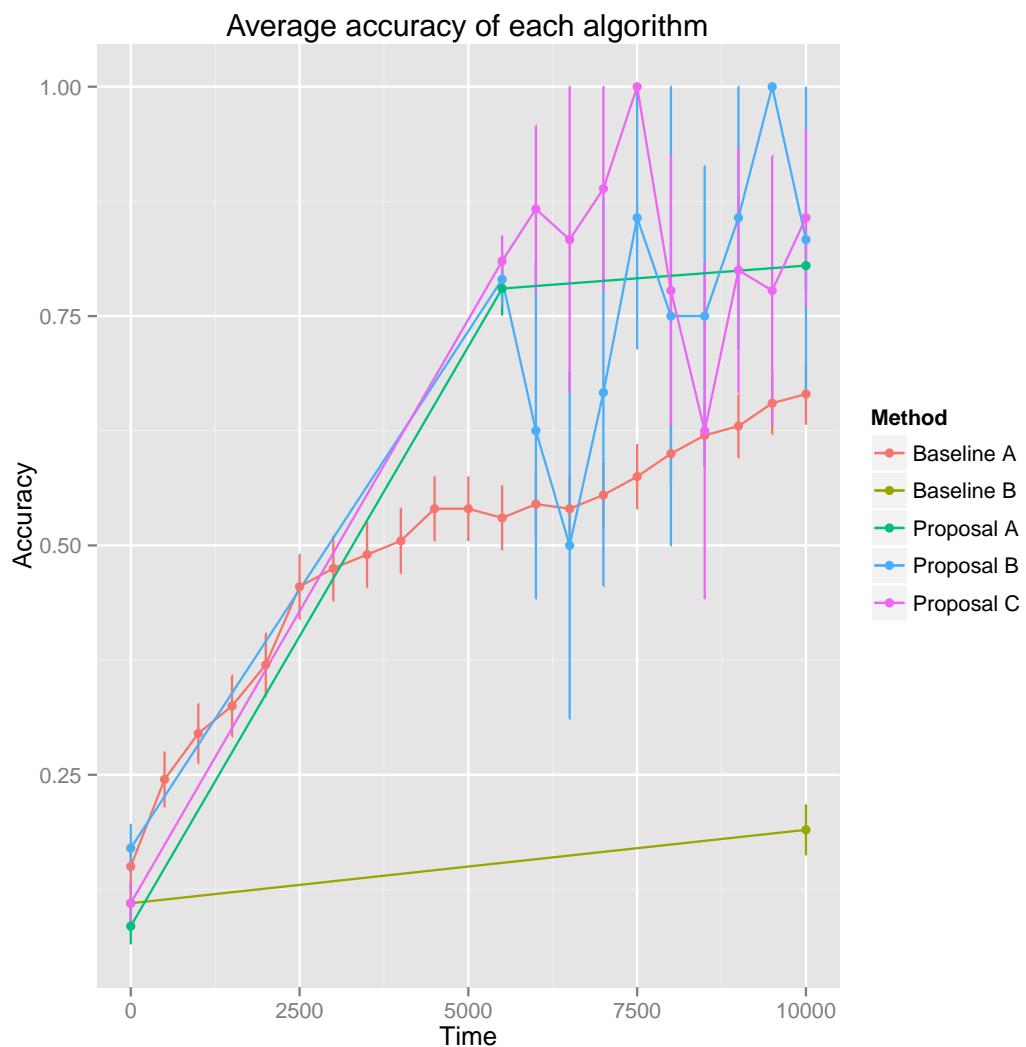


図 5.8 実サイト A から得た評価関数における各アルゴリズムの正確度

ベースラインアルゴリズム B（単純山登り）は解の比較を繰り返すには与えられた限界観測数が不十分であり、一回しか解の移動を行うことが出来ず、他のアルゴリズムに比べると正確度が低くなっている。

提案手法はいずれもベースラインを上回る結果となった。今回の評価関数は施策を行う・行わないの二値変数の線形結合であり非線形性が無いため、 $n_1 = 5500$ の初期値決定フェーズ完了時点でいずれのアルゴリズムも高い正確度で最適解を推定することができている。

解探索フェーズでは、提案手法 B および C が観測の短縮を用いて頻繁な解移動を行っていることがわかる。ただしその変化はばらつきが大きく、評価値が低い解に移動してしまう様子も見て取れる。

5.3 実際のウェブサイトにおける実験

本節では、実際のウェブサイトに提案アルゴリズムに基づいて実装したウェブページ最適化プログラムを導入し、実際にウェブページ最適化を行うことで提案アルゴリズムを評価する。

5.3.1 実験内容

今回の実験は、以下のことを目的としている。

- 提案手法が実際のウェブサイトに最適化プログラムとして組み込むことができ、実際に機能することを評価する。
- 実サイト A のような大規模サイトだけでなく、中規模サイトにおいても提案アルゴリズムが有効であることを評価する。

そのため一日の訪問者数が数百～数千程度の中規模サイトを選出し、提案アルゴリズムに基いて実装したウェブページ最適化プログラムを導入した。

対象サイト

今回は実験を行う対象のウェブサイトとして壁紙画像検索サービス「Imagerous^{**3}（以下実サイト B と呼ぶ）」を採用した。実サイト B はパソコンやスマートフォンの壁紙に使える画像を検索したり、アイドルや女優のグラビア画像を検索したりして楽しむことを目的としたウェブサイトである。検索結果で表示されるページのひとつひとつがテーマに則した記事になっており、記事を閲覧すると記事のテーマに関連する画像がタイル上に並べられて表示される（図 5.9 参照）。この記事ページが検索エンジンからのランディングページになっており、実サイト B のメインコンテンツとなるページである。

^{**3} Imagerous* <http://imagero.us/>

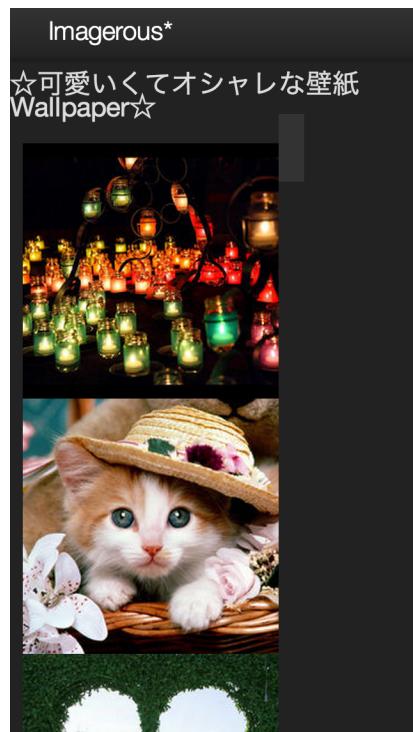


図 5.9 実サイト B の記事ページ例

今回実サイト B を実験対象として選択した理由について説明する。ひとつはウェブサイトの構造が非常に単純なため、評価指標を設けやすい点である。SNS やメールクライアントなど、ツールとしての側面が強いウェブサイトは機能が充実している分ユーザの使い方や目的も多様であるため指標化できる箇所が複数あり、どの指標をウェブサイト全体の評価指標として設定するかが難しい。一方で、実サイト B は記事ページの写真をクリックして写真を拡大表示して、また記事ページに戻るというように、ユーザの行動は記事ページと写真ページを往復するのみである。利用目的および利用方法が単純であるため、ウェブページ最適化によるウェブサイト全体への効果を測定することが容易と判断し、実サイト B を実験の対象とした。

もうひとつの理由は実サイト B の訪問者がほとんど新規ユーザであり、多くが検索エンジンの検索結果から流入している点である。リピートユーザが多いサイトではウェブページ最適化のためにウェブページに一部変更を加えた際に 2.5.2 項で説明した目新しさ効果が生じてしまい、極端にユーザのアクション数が増える可能性がある。一方で実サイト B の訪問者の多くは以前に実サイト B を閲覧したことが無いユーザであるため、目新しさ効果が生まれることはなく、目新しさ効果が収束する期間を待つことなく実験を繰り返すことができる。

今回はウェブページのデザインにかかる変数が多いためユーザが使用するデバイスによる影響が大きいと考え、パソコンから閲覧された際の観測値は使用せず、スマートフォンからの

アクセスによる観測値に絞って最適化を行なった。

評価指標

実サイト B はウェブページに張られたバナー広告による広告収益によって運営されている。広告収益を増加させるには、ウェブサイトにランディングしたユーザを着実に引き込んでできるだけ多くのページ数を見てもらい、ユーザに広告をクリックしてもらうチャンスを増やすことが重要になる。ユーザをウェブサイトにひきつける度合いを表す評価指標には

- 訪問あたり閲覧ページ数
- 直帰率
- 訪問あたり滞在時間

などが考えられる。今回はこの中から訪問あたり閲覧ページ数を最適化する評価指標として設定した。ユーザの中には一回の訪問で数百ものページを閲覧する、ロボットとみられる動きをするものもあったため、評価を行う際には上限値を設定した。

解候補

今回はこのページを構成する要素として表 5.3 に示す変数と値を設定し、これらの組み合わせを解候補として実験を行うことにした。参考のために解候補の一例を付録 A に示す。

表 5.3 実サイト B を構成する変数と値

変数	対応する要素	取りうる値
x_1	画像の枠線の太さ	0px, 5px
x_2	画像の間の間隔	0px, 5px, 10px
x_3	画像のサイズ	100px, 200px, 300px
x_4	画像の切り抜き	正方形, 正円

ここでは各変数の値は 0 以上の整数で表すこととし、整数の組み合わせによって解を表現する。たとえば、 $x = (0, 2, 1, 0)$ は下記のような設定を表す。

- 画像の枠線の太さ:0px
- 画像の間の間隔:10px
- 画像のサイズ:200px
- 画像の切り抜き:正方形

評価するアルゴリズム

今回の実験では表 4.1 で説明したベースラインアルゴリズム B(単純山登り)と提案アルゴリズム A を適用した。同じ期間で提案アルゴリズムがベースラインアルゴリズムよりも良い解を見つけることができることを示す。

パラメータ設定

有意水準 $\alpha = 0.05$ 、検出力 $1 - \beta = 0.8$ とした。効果量は $\Delta = 0.3$ 程度を検出することとした。これらのパラメータから各サンプルサイズを以下のように算出した。

- 初期値決定フェーズに用いるサンプルサイズ: $n_1 = 120$
- 解探索フェーズの解の比較に用いるサンプルサイズ: $n_2 = 200$

5.3.2 実験装置

ウェブページ最適化プログラムはクライアントサイド方式によって実装した。下記にプログラムの仕組みの全体像を図 5.10 に示す。

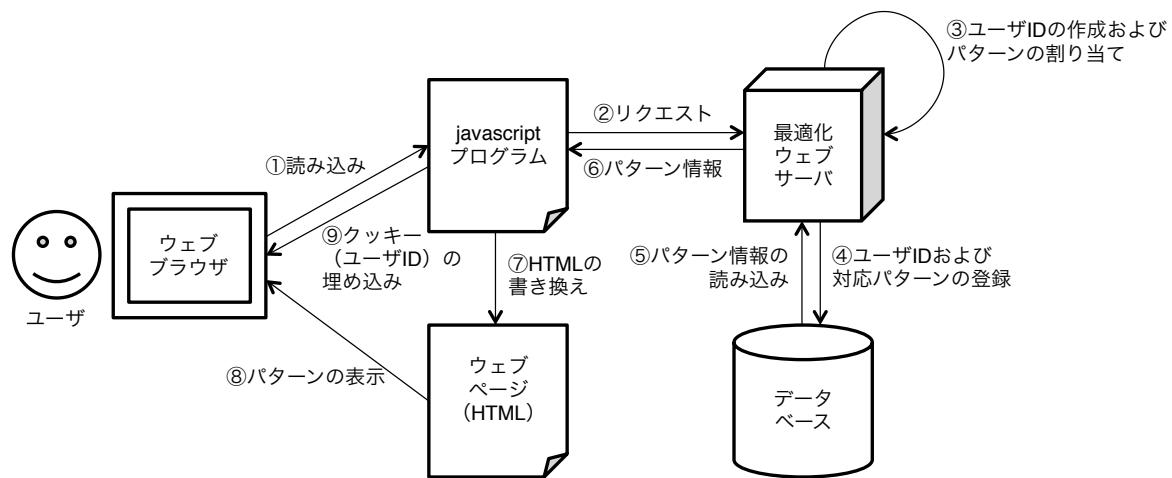


図 5.10 ウェブページ最適化プログラムの全体像

ユーザは実サイト B にアクセスすると、ウェブサイトをホストするウェブサーバからウェブページを表す HTML と javascript プログラムを受け取る。ウェブブラウザがこの javascript プログラムを実行すると、ウェブページ最適化のために用意された専用のサーバ（最適化ウェブサーバと呼ぶ）にリクエストを送信する。最適化ウェブサーバはそのリクエストに対して

ユーザごとにユニークな ID を発行し、最適化アルゴリズムに基いてそのユーザに表示するパターンを決定する。ここで決定されたユーザ ID とパターンの対応付けはユーザが再度実サイト B に訪問した際にも同じパターンを表示できるようにデータベースに記録される。

データベースにはユーザ情報に加えて、パターンを表示するために変更を加えるべき HTML の DOM 要素の style 属性とその値（パターン情報と呼ぶ）が記録されている。ユーザに表示するパターンが決定されると、必要なパターン情報がデータベースから最適化ウェブサーバに受け渡される。

最適化ウェブサーバはそのパターン情報を javascript プログラムにレスポンスとして受け渡す。javascript プログラムはパターン情報に応じて HTML の DOM 要素の書き換えを行い、ウェブブラウザ上に目的のパターンを表示する。この際、javascript プログラムはウェブブラウザのクッキーにユーザ ID を埋め込むことで、同じウェブブラウザから再度実サイト B にアクセスがあった際に同じパターンを表示できるようにする。

ユーザがページの閲覧を開始したり、ウェブサイト内の他のページに移ったりした時は、その情報は即時に javascript プログラムによって捕捉されて最適化ウェブサーバに送信され、ログデータとしてデータベースに記録される。また、ユーザがウェブサイトに滞在している間は 5 秒ごとに javascript プログラムがユーザのページ滞在時間をチェックし、最適化ウェブサーバにログを送信する。このようにして、最適化ウェブサーバはユーザの行動を捕捉する。

また、ユーザの訪問があるたびに解の評価を行うとウェブページの読み込みが遅くなってしまうため、30 分に 1 回実行されるバッチプログラムによって解の評価を行うことにした。

5.3.3 実験結果

2014 年 1 月 12 日から 19 日にかけて実サイト B で行なった実験の結果を表 5.4 に示す。

表 5.4 実サイト B における実験結果

	\mathbf{x}^*	$\mathbb{E}[y \mathbf{x}^*]$	$ Y_{\mathbf{x}^*} $
ベースライン B	(1,0,0,0)	2.792	178
提案 A	(0,0,0,0)	2.896	182

ベースラインアルゴリズム B で見つけられた最適解 $\mathbf{x}^* = (1, 0, 0, 0)$ は下記のような組み合せである。

- 画像の枠線の太さ:5px

- 画像の間の間隔:0px
- 画像のサイズ:100px
- 画像の切り抜き:正方形

最適解 x^* の評価値の期待値は $\mathbb{E}[y|x^*] = 2.792$ であった。ベースラインアルゴリズム B は期間中に解 x^* を 178 回観測した。

これに対して提案アルゴリズム A で見つけられた最適解 $x^* = (0, 0, 0, 0)$ は下記のような組み合わせである。

- 画像の枠線の太さ:0px
- 画像の間の間隔:0px
- 画像のサイズ:100px
- 画像の切り抜き:正方形

最適解の評価値の期待値は $\mathbb{E}[y|x^*] = 2.896$ であり、ベースラインアルゴリズム B によって発見されたものを上回った。 t 検定の結果、二つの期待値の差には有意差が見られなかった。

図 5.11 および表 5.5 にそれぞれの探索アルゴリズムが探索を行なった推移を示す。図 5.11 では、最適化を行う以前 1 ヶ月の評価指標の平均値を算出した結果得られた $\mathbb{E}[y|x^*] = 2.710$ を始点とした。エラーバーは暫定解の期待値の標準誤差を表す。

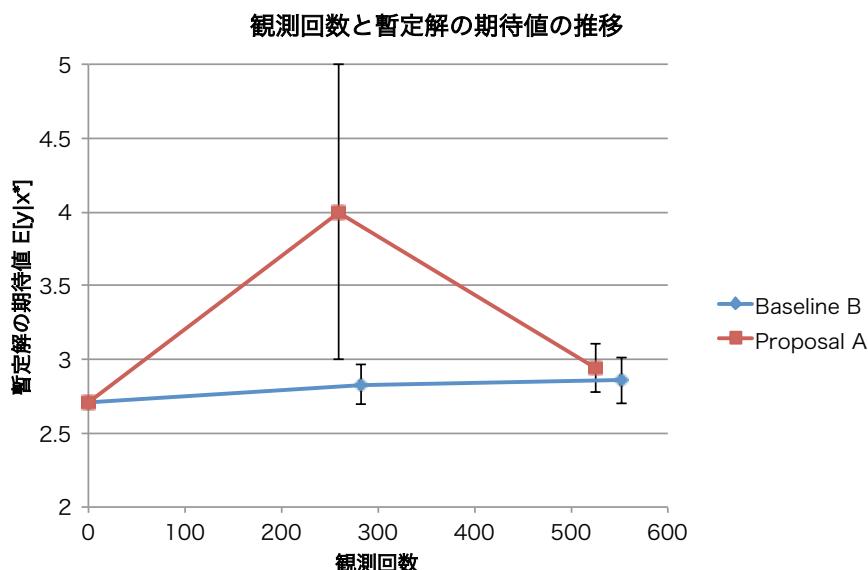


図 5.11 観測回数と暫定解の期待値の推移

表 5.5 暫定解と期待値の推移

アルゴリズム	観測数	暫定解	期待値	期待値の標準誤差
ベースライン B	0	(0,0,0,1)	N/A	N/A
ベースライン B	282	(1,0,0,1)	2.832	0.137
ベースライン B	552	(1,0,0,0)	2.860	0.156
提案 A	0	N/A	N/A	N/A
提案 A	259	(1,0,0,0)	4.000	1.000
提案 A	525	(0,0,0,0)	2.942	0.164

ベースラインアルゴリズム B の単純山登り法では、解空間から無作為に抽出した解を初期値として近傍解との比較を繰り返して探索を行う。一回の解比較にはそれぞれの解に n_2 のサンプルサイズを割いて観測値を収集する。その結果、着実に解の改善を行っていることがわかる。

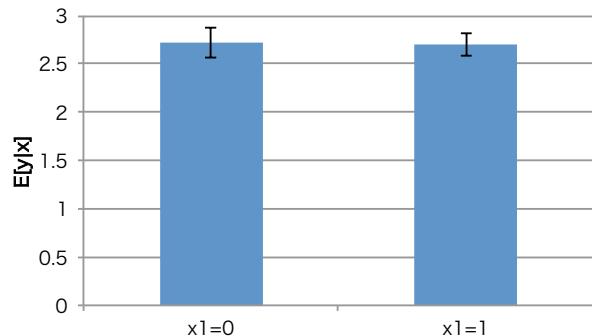
一方、提案アルゴリズム A では観測数 $n < n_0$ の間を初期解決定フェーズとし、解空間全体から解を抽出して観測を繰り返す。このフェーズではひとつひとつの解の観測回数が少ないため、予測された $x = (1, 0, 0, 0)$ は初期解決定フェーズ完了時点で 2 回しか観測されておらず、標準誤差も 1.000 と大きくなっている。初期値決定フェーズ終了後、 $x = (1, 0, 0, 0)$ を初期解として局所探索を行い、最適解 $x^* = (0, 0, 0, 0)$ に到達する。

最後に二つのアルゴリズムで合わせて得られた観測データから算出された、各変数における変数値の期待値を図 5.12 から図 5.15 に表す。各図のエラーバーは 95% 信頼区間を表している。どの変数 x_i についても変数値 $v = 0$ を取ったときが最も期待値が高くなっていること、提案アルゴリズム A は良い推定ができていることが伺える。

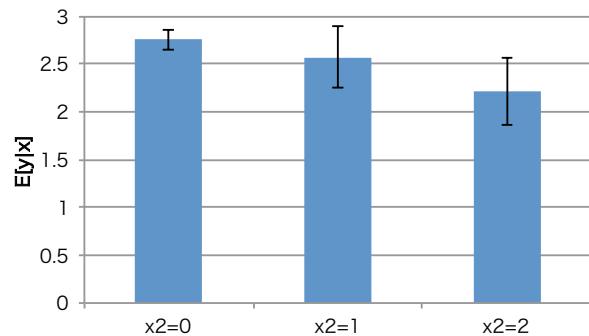
ここで変数 x_1 と変数 x_2 の間で各変数値の期待値を比較したものを図 5.16 に示す。変数 x_1 が変数値 $v = 1, 2$ をとる限りは変数 x_2 で変数値 $v = 1$ を取ったほうが期待値が高いが、変数 x_1 が変数値 $v = 0$ を取る場合に限り、変数 x_2 で変数値 $v = 2$ を取ったほうが期待値が高いことがわかる。この変数 x_1 と変数 x_2 の間の非線形性が初期値決定フェーズで最適解 $x^* = (0, 0, 0, 0)$ が推定されなかった一因と考えられる。

また、表 5.5 からわかるように、提案アルゴリズム A はベースラインアルゴリズム B より先にベースラインアルゴリズム B の最適解 $x = (1, 0, 0, 0)$ に到達していたという点で一足早い探索を行うことが出来たと考えられる。

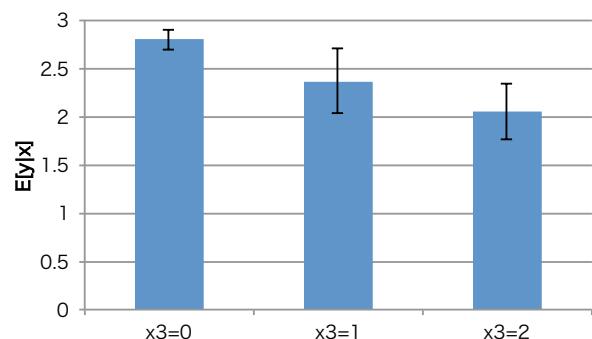
x1: 期待値の比較

図 5.12 x_1 : 期待値の比較

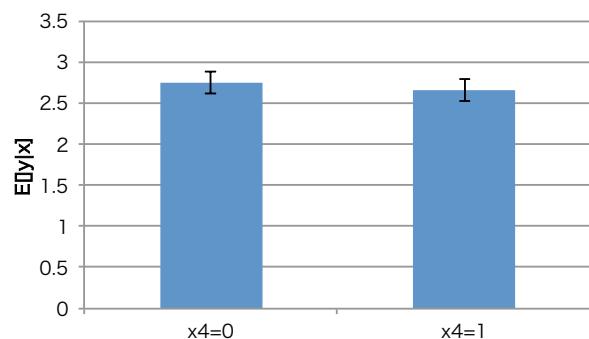
x2: 期待値の比較

図 5.13 x_2 : 期待値の比較

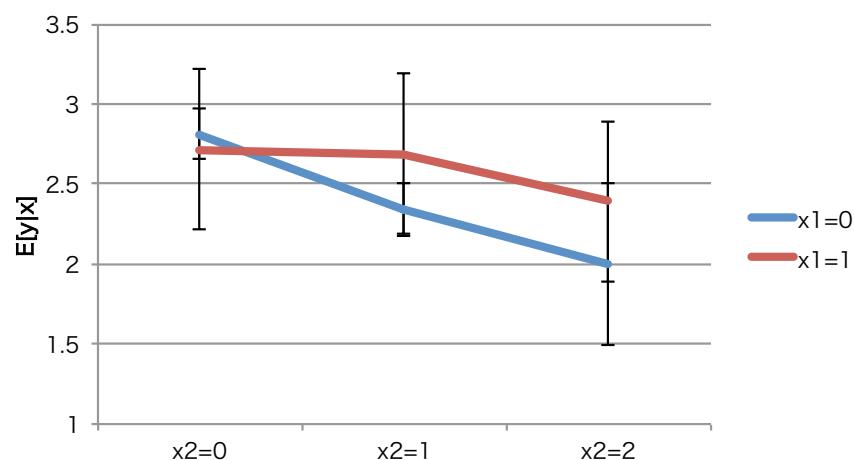
x3: 期待値の比較

図 5.14 x_3 : 期待値の比較

x4: 期待値の比較

図 5.15 x_4 : 期待値の比較

x1-x2: 期待値の比較

図 5.16 $x_1 \times x_2$: 期待値の比較

第 6 章

考察

本章では、第 3 章から第 5 章までを振り返り、考察を行う。6.1 節では提案アルゴリズムの適用可能性について議論する。6.2 節ではウェブページ最適化問題における変数の抽出の重要性について議論する。6.3 節では実験に用いたウェブサイトの妥当性について議論する。6.4 節ではウェブページ最適化問題と従来の実験計画法の関係について議論する。6.5 節では提案手法のウェブ以外の分野での応用可能性について議論する。

6.1 提案アルゴリズムの適用可能性

本節ではウェブページ最適化を取り巻く様々な状況を想定し、提案アルゴリズムの適用可能性について議論する。

6.1.1 限界観測数 N が非常に小さい場合

ウェブサイトによっては訪問者数が少なく、非常に小さい限界観測数 N の中で最適化を行わなければならない状況もありうるが、提案手法は限界観測数 N が小さい場合でも最適化を行うことができる。

提案手法では戦略判断フェーズで最適化に必要最低限なサンプルサイズ n_0 を算出することで、最適化を行うか否かの判断を行う。それぞれの変数が取りうる値の数を一定とした場合、解空間のサイズは変数の数 m に対して指数関数オーダー $O(c^m)$ で増えていくのに対し、最適化を行うのに最低限必要なサンプルサイズ n_0 は式 4.1 に示すように線形関数オーダー $O(m)$ で増えていく。したがって、提案手法で最適化を行うのに必要なサンプルサイズ n_0 は試すべき変数の増加に対して強く、少ない訪問者数でも最適化を行うことができる事を示して

いる。

6.1.2 解空間 X が非常に大きい場合

解空間 X が非常に大きい場合に局所探索法を用いると、限界観測数 N のなかで十分な数の解を評価できずに最適化が終了してしまう可能性がある。しかし、提案手法では初期解決決定フェーズで解空間全体から収集した観測値から最適解を推定するため、解空間の中で大域的に評価値が高い解から探索をスタートすることができる。

予測モデルによっては評価関数への当てはまりが悪く、評価値の高い解から探索を行うことが出来ない可能性もあるが、その後の解探索フェーズで局所探索法を用いて探索を行うことで、最終的には評価値の高い解に到達することができる。

また、本研究では局所探索法の一例として山登り法のみを取り上げたが、大域解の探索に対応できる擬似焼きなまし法やタブーサーチなどのメタヒューリстиクスを用いた局所探索法を用いることで、大域的な最適解を探索する工夫も考えられる [25]。加えて 4.2 で紹介した解の類似度の利用やレーシングアルゴリズムも、探索空間を絞り込むための工夫として活用することができる。

6.1.3 時間によって評価関数が変化する場合

2.5.4 項でも説明したように、ウェブページ最適化にまつわる大きな外的要因のひとつに、時間による評価関数の変化がある。提案手法では時間による評価関数の変化を考慮していないため、観測データを活用する際に誤った判断を行なってしまう可能性がある。

たとえば、ウェブページ最適化を行うあるウェブページが仕事でよく使われる傾向にあるために、休日はウェブページの評価指標がすべての解について下落するケースが考えられる。このとき、平日に観測されたデータを持つ解 x_d と休日に観測されたデータを持つ解 x_e を比較してしまうと、解本来のパフォーマンスとは関係なく解 x_d の方が勝ちやすくなってしまう。このように時間要因によって解の評価を誤ることを防ぐには、実験を行う期間を絞る、常に同じ時間帯で実験を行うといった工夫が考えられる。

この例からもわかるように、観測データは解空間の特性を予測して効率的な探索を行う上で有用である一方、時間を経た観測データは誤判断を生む要因となってしまう可能性がある。これに対しては、観測データのサイズを固定する工夫が考えられる。評価関数が時間変化するウェブページ最適化問題では観測データのサイズも効率的な探索のために重要なパラメータに

なると考えられる。

6.2 最適化問題における変数の抽出の重要性

5.1 節のシミュレーションにおいて、ウェブページ最適化のパフォーマンスはその評価関数が含む非線形性によって大きく変わることを示した。もし初期解決定フェーズにおいて評価関数が精度よく線形モデルに回帰すれば高い精度で最適解を予測することができるが、線形モデルのあてはまりが悪かった場合は有利な解から局所探索を行うことができなくなる。評価関数に含まれる非線形性は変数の取り方に依るため、ウェブページから要素を切り出す方法も効率的な探索を行うためには欠かせない要素である。

たとえば図 6.1 に示すように、お申込みボタンのデザインを最適化する場面を考えてみる。このとき変数としてページの背景色 x_1 とお申込みボタンの色 x_2 を取り出し、それぞれが $V_1 = V_2 = \{ \text{白, 黒} \}$ から値をひとつ取るものとする。



図 6.1 評価関数に非線形性が生じる変数の抽出例

この実験によってユーザが好む背景色およびボタンの色を試すことができるが、白い背景色に白いボタンを載せた場合と黒い背景色に黒いボタンを載せた場合は、背景色とボタンの色が同化してしまい、ボタンの可視性自体が損なわれてしまっている。このような組み合わせではユーザがボタンの存在に気づかず、ボタンがクリックされる回数が減ってしまうかもしれない。このとき、評価関数の中に「ボタンの見やすさ」を表す非線形項 $x_1 x_2$ が入り込んでしまっていることになる。

また、ボタンの文言のように名義尺度をとる変数について一度にたくさん値を試すと評価

関数に多峰性が生じることが予想されるため、ダミー変数を用いてそれぞれを独立の変数とする、一度にテストする値の数を制限するといった工夫が考えられる。

6.3 実験に用いたウェブサイトの妥当性

本研究では一日の訪問数が数万規模の大規模サイトとして実サイト A、一日の訪問数が数百～数千規模の中規模サイトとして実サイト B の 2 つを取り上げて実験を行い、ウェブサイトの規模に関しては幅広く提案手法が機能することを示すことができた。

また、実サイト A では広告のクリック率という観測値が 0 または 1 の二値で与えられる評価関数について実験を行ったのに対し、実サイト B では訪問あたりのページビュー数という、観測値が二値ではなくある程度の幅を持って与えられる評価関数について実験を行なった。このことから、様々な確率分布を持つ評価関数について提案手法が機能することを評価できたと考えている。

ただ、SNS や E コマースサイトのようにリピートユーザが多いウェブサイトでは 2.5.4 項で述べた目新しさ効果のような別の要因によって異なる挙動が観測されるかもしれない。このようにリピートユーザが多いウェブサイトでの実験は今後の課題としたい。

6.4 ウェブページ最適化問題と実験計画法の関係

実験計画法は、研究対象の最適な組み合わせを見つけるために事前に実験を設計する手法である。たとえば農作物において肥料や土の栄養分の組み合わせが作物の成長に与える影響や、製造業において製造ラインの室温や回転数などにパラメータの組み合わせが製品の品質のばらつきに与える影響を測るための手法として発展してきた [40][41]。

これらの問題は一種の組み合わせ最適化問題だということができるが、実験計画法で取り扱う分野は解の生成が容易でないものが多い。たとえば農業において解を生成するには対応する肥料や栄養分を組み合わせた圃場を新たに用意する必要がある。製造業においても、任意の解を生成することはプラグの抜き差しや温度調整を要するため容易ではない。そのため、局所探索のように任意の解の生成を要請する解法をこれらの分野で用いることは困難である。

そこで広く活用されてきたのが直交計画である。直交計画では直交表を用いて解の割り当てを行うことで、生成すべき解の数を絞りながら解空間全域からまんべんなく観測値を収集することを可能にする。直交計画では得られた観測データに分散分析を施すことで各変数の影響度を算出して最適な組み合わせを求めるが、これはウェブページ最適化問題の線形モデルによる

最適解の予測に相当する。つまり、直交計画は任意の解の生成が容易でない分野で、線形モデルによる最適解の予測を行うための実験を事前に設計する手法だということができる。

それに対してウェブページの場合は任意の解の生成にコストがかからない。そのため直交表を用いることなく、解空間から無作為に解を抽出して観測を行うことで、同様の結果を得ることができる。

6.5 ウェブ以外の分野への応用可能性の検討

ここで、ウェブページ最適化問題を再掲する。

ウェブページ最適化問題

解 $x \in X$ について得られる観測値を y 、限界観測数を N 、観測データを Y とするとき、下式を満たす解 x^* を求めよ。

$$x^* = \arg \max_{x \in X} \mathbb{E}[y|x] \text{ s.t. } \sum_{x \in X} |Y| \leq N$$

今回の提案手法は、この形をとる問題ならばウェブ以外の分野にも適用することができる。つまり下記の条件を満たしさえすれば、ウェブページ最適化問題に落としこんで最適解を探索することができるようになる。

- 解 x が組み合わせによって表現される。
- 解 x の生成が容易である。
- 解 x に対する観測データ Y が得られる。
- 制約として限界観測数 N を与えても良い。

たとえば、ファッショニもシャツやパンツなどのアイテムの組み合わせと考えれば、ひとつの解 x として表現することができる。服装は自由に変えることができるので、容易に任意の解 x を生成することができる。その解を写真に撮ってアップロードすることで、ブログやソーシャルメディアを介したコメントや「いいね！」などの反応 y を観測することができれば、ファッショニもウェブページ最適化問題として捉えて、最適化することができる。

その他にも様々な媒体のデザイン (x : 用いる素材とレイアウト, y : それに見た人の反応) 料理 (x : 具材の組み合わせ, y : 食べた人の反応、口コミ) など、様々な応用が考えられる。ウェブ技術の発展によって、解に対する人々の反応を広く計測できるようになったところ、任意の解の生成が容易になったところにブレイクスルーがある。ここからはいくつか特徴的な分

野を取り上げて、ウェブページ最適化問題の応用可能性について検討する。

6.5.1 メディア製作への応用可能性

今回提案した最適化手法のひとつの応用分野として、映画や音楽などのメディア製作への応用が考えられる。近年では YouTube^{*1} やニコニコ動画^{*2}などの動画共有サービスが普及したことによって、発信者がネットユーザに作品を評価してもらうための地盤がしてきた。そのためクリエイティブ製作者は作品の再生回数や再生時間といった評価指標を設定することで、データを収集して実験を行うことができる。実際に Brightcove^{*3} や Treepodia^{*4} をはじめとする動画の再生状況を解析する動画配信プラットフォームが登場して始めており、E コマースサイトに導入する動画の A/B テストも実際に行われている^{*5}。

クリエイティブを作成する側も高機能の音楽編集ソフトウェアや動画編集ソフトウェアが普及したことによって、録音しなおしたり、撮影しなおしたりする手間をかけることなく作品を編集して解を生成できるようになった。「ナレーションの有無」や「効果音 A/B/C」などの変数とそれが取りうる値からなる解空間を用意しておけば、本研究で提案したアルゴリズムを導入することが可能だと考えている。

6.5.2 ものづくりへの応用可能性

近年 3D プリンタが普及したことによって、コンピュータ上の CAD ソフトウェアを用いてモデリングを行うことさえできれば、実際に手に触れることができる製品を出力できるようになった。このような動きはメーカーズムーブメントやファブラボと呼ばれており、「ビットからアトムへ」という標語のもと世界中で盛り上がりを見せている。以前は工場の製造ラインを使って大きなバッチ単位で生産しなければならなかつたような製品も、個人用 3D プリンタでひとつひとつ生産することができるようになる [42]。製品の修正も CAD ソフトウェア上で行うことができるため、新たな機能を導入したり、製品のバリエーションを増やしたりすることが容易になる。したがって、ものづくりにおいてもウェブと同じように容易に任意の解を生成できるようになる。

^{*1} YouTube <http://www.youtube.com/>

^{*2} ニコニコ動画 <http://www.nicovideo.jp/>

^{*3} Brightcove <http://www.brightcove.com/>

^{*4} Treepodia <http://treepodia.com/>

^{*5} Ecommerce Video A/B testing - Test Your Intuitions! : Treepodia <http://blog.treepodia.com/2010/07/ecommerce-video-ab-testing-test-your-intuitions/>

3D プリンタが普及した世界では製品を生産して配送する必要はなく、CAD ソフトウェアで作成したデザインファイルをウェブで受け渡しして自宅の 3D プリンタで生成すればよい。実際に Ponoko^{*6}や Shapeways^{*7}を始めとする、デザインファイルをやりとりするためのプラットフォームが登場し始めている。今後このようなプラットフォームを介したウェブ上でのモノのやりとりが増えていけば、自分の製品を多くのユーザに評価してもらうことができる。そこで製品の利用状況に関する適切な評価指標を設定することができれば、今回提案した最適化手法をものづくりにおいても活用することが可能だと考えている。

^{*6} Ponoko <https://www.ponoko.com/>

^{*7} Shapeways <http://www.shapeways.com/>

第 7 章

まとめ

本研究では、ウェブサイトの訪問者数とテストする要素の数に合わせて汎用的に最適化を行うことができるウェブページ最適化手法を提案した。

まず第 3 章ではウェブページ最適化と組み合わせ最適化問題の類似性に着目し、ウェブページ最適化問題を定式化した。ウェブページ最適化問題は解の評価値がばらつきを持った値として与えられ、観測数に制約がある中で最適解を探索する組み合わせ最適化問題である。この問題に対して局所探索法を用いた解法および効率的な探索を行うための工夫を示した。

それを踏まえて第 4 章では、ウェブサイトの訪問者数とテストする要因の数に応じて最適化を行うウェブページ最適化手法を提案した。提案手法は戦略判断フェーズ、初期解決決定フェーズ、解探索フェーズの 3 つからなる探索手法である。戦略判断フェーズでは、それぞれのフェーズで解の評価を行うのに十分な観測数を算出する。初期解決決定フェーズでは、解空間全体から得た観測データを活用することで、最適解の予測を行う。解探索フェーズでは、初期解決決定フェーズで予測した最適解を初期解として局所探索を行う。

第 5 章では著者が設定した仮想の評価関数でのシミュレーション実験、実際の大規模サイトで得られたデータにおけるシミュレーション実験、実際の中規模サイトに提案アルゴリズムを実装したウェブページ最適化プログラムを導入する実験によって、提案手法の評価を行なった。その結果、大規模サイトでも中規模サイトでも提案手法が有効であることが示された。また、実際のウェブサイトに導入した場合でも機能することが示された。

第 6 章では考察として提案手法の適用可能性、従来の実験計画法との関係性、ウェブ以外の分野での応用可能性について主に議論した。ウェブページ最適化問題は、解が組み合わせによって表現され、解の生成が容易であり、解に対する観測値が与えられる問題であれば、ウェブ以外の問題にも当てはめることができる。今後はものづくりの世界でも 3D プリンタや 3D

スキヤナの登場によって、ウェブサービス開発と同じように低コストかつ高速に製品開発ができるようになると考えられる。本研究で提案したウェブページ最適化問題および提案した手法は、将来ウェブだけではなく様々な分野で適用することができると考える。

付録 A

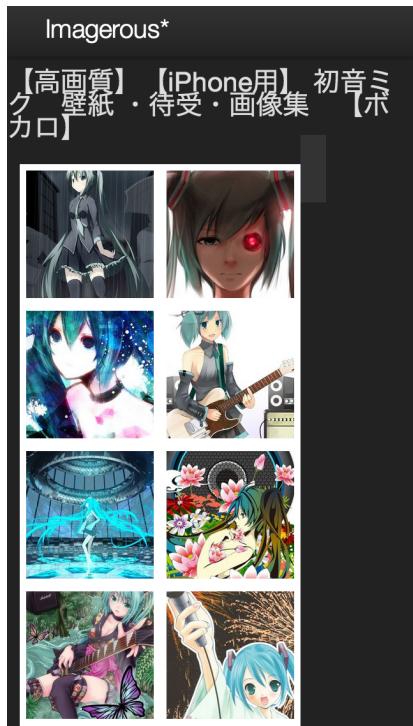
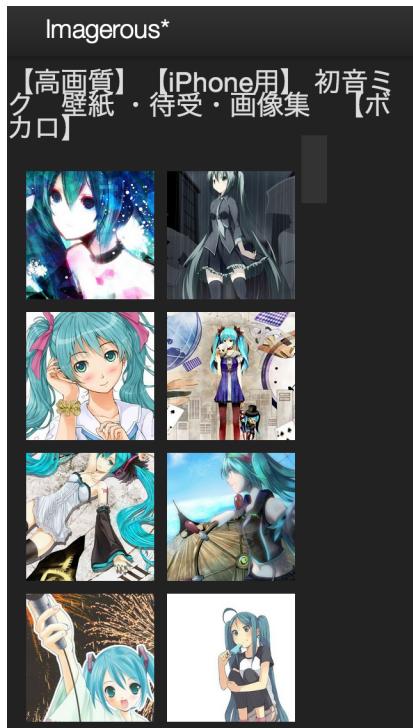
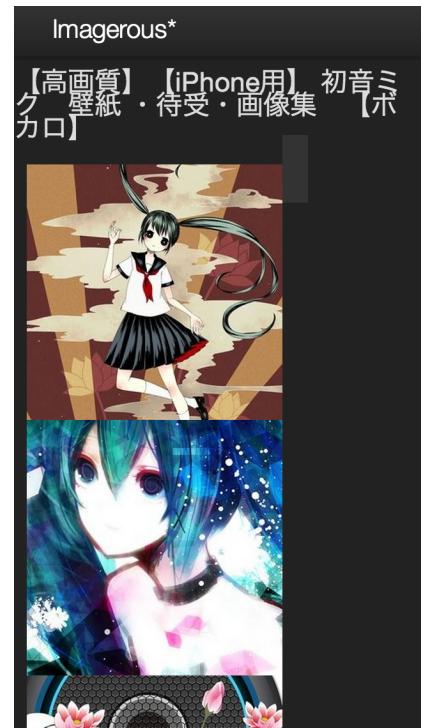
実サイト B 仮説パターン例



図 A.1 仮説パターン $x = (0, 0, 0, 0)$



図 A.2 仮説パターン $x = (0, 0, 0, 1)$

図 A.3 仮説パターン $x = (1, 0, 0, 0)$ 図 A.4 仮説パターン $x = (1, 2, 2, 1)$ 図 A.5 仮説パターン $x = (0, 1, 0, 0)$ 図 A.6 仮説パターン $x = (0, 0, 1, 0)$

謝辞

本論文を作成するにあたり、指導教官の松尾豊准教授から丁寧かつ熱心なご指導を賜りました。ここに感謝の意を表します。また、ご指導を頂いた副指導教官の阿部力也特任教授に感謝致します。

人物検索サイト「あの人検索スパイシー」での実験にあたりオーマ株式会社の皆様、特にオーマ株式会社取締役 アグチバヤル・アマルサナー氏には多大なご協力をいただきました。ここに深く感謝致します。

最後に、研究生活において苦楽を共にし、支え続けてくれた松尾研究室の皆様に改めて感謝致します。特に同級生の濱野将司君、衛藤貴文君は素晴らしい研究仲間でした。感謝致します。

東京大学大学院工学系研究科
技術経営戦略学専攻
松尾研究室修士 2 年
平成 26 年 2 月 飯塚修平

参考文献

- [1] Eric Ries. *The Lean Startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Random House Digital, Inc., 2011.
- [2] Dan Siroker. How obama raised \$60 million by running a simple experiment <http://blog.optimizely.com/2010/11/29/how-obama-raised-60-million-by-running-a-simple-experiment/>, 2010.
- [3] Thomas Eisenmann, Eric Ries, and Sarah Dillard. Hypothesis-driven entrepreneurship: The lean startup. *Harvard Business School Entrepreneurial Management Case*, No. 812-095, 2012.
- [4] Carmen Nobel. Teaching a'lean startup'strategy, 2012.
- [5] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, Vol. 18, No. 1, pp. 140–181, 2009.
- [6] Thomas Crook, Brian Frasca, Ron Kohavi, and Roger Longbotham. Seven pitfalls to avoid when running controlled experiments on the web. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1105–1114. ACM, 2009.
- [7] Saharon Rosset and Slava Borodovsky. A/b testing using the negative binomial distribution in an internet search application. 2011.
- [8] Zaiqing Nie, Ji-Rong Wen, and Wei-Ying Ma. Webpage understanding: beyond page-level search. *ACM SIGMOD Record*, Vol. 37, No. 4, pp. 48–54, 2009.
- [9] Christian Kohlschütter and Wolfgang Nejdl. A densitometric approach to web page segmentation. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 1173–1182. ACM, 2008.
- [10] Shian-Hua Lin and Jan-Ming Ho. Discovering informative content blocks from web

- documents. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 588–593. ACM, 2002.
- [11] Parag Mulendra Joshi and Sam Liu. Web document text and images extraction using dom analysis and natural language processing. In *Proceedings of the 9th ACM symposium on Document engineering*, pp. 218–221. ACM, 2009.
 - [12] Ranjitha Kumar, Jerry O Talton, Salman Ahmad, and Scott R Klemmer. Bricolage: example-based retargeting for web design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2197–2206. ACM, 2011.
 - [13] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Vips: a visionbased page segmentation algorithm. 2003.
 - [14] Jie Zou, Daniel Le, and George R Thoma. Combining dom tree and geometric layout analysis for online medical journal article segmentation. In *Digital Libraries, 2006. JCDL'06. Proceedings of the 6th ACM/IEEE-CS Joint Conference on*, pp. 119–128. IEEE, 2006.
 - [15] Dave Gehrke and Efraim Turban. Determinants of successful website design: relative importance and recommendations for effectiveness. In *System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, pp. 8–pp. IEEE, 1999.
 - [16] O Byung Kwon, Choong-Ryuhn Kim, and Eun Jong Lee. Impact of website information design factors on consumer ratings of web-based auction sites. *Behaviour & information technology*, Vol. 21, No. 6, pp. 387–402, 2002.
 - [17] Imran Sarwar Bajwa, Imran Siddique, and M Choudhary. Web layout mining (wlm): A new paradigm for intelligent web layout design. In *Proceedings of the ITI 4th International Conference on Information and Communications Technology (ICICT'06)*. IEEE, Los Alamitos, CA, 2006.
 - [18] Allison Thompson. Automated generation of website content and layout. *Michigan Celebration of Women in Computing*, p. 43, 2009.
 - [19] Srinivasan H Sengamedu and Rupesh R Mehta. Web page layout optimization using section importance. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008.
 - [20] T. Ash, M. Ginty, and R. Page. *Landing Page Optimization: The Definitive Guide*

- to Testing and Tuning for Conversions.* ITPro collection. Wiley, 2012.
- [21] 栗原伸一. 入門統計学検定から多変量解析・実験計画法まで. オーム社, 2011.
 - [22] 奥野忠一, 塩見正衛. 直交表による多因子計画のわりつけ. 農業技術研究所報告. A, 物理統計, No. 12, pp. 23–76, 1965.
 - [23] John White. *Bandit Algorithms for Website Optimization.* O'Reilly, 2012.
 - [24] Round M Kohavi R. Front line internet analytics at amazon.com <http://ai.stanford.edu/~ronnyk/emetricsAmazon.pdf>, 2004.
 - [25] 古川正志, 川上敬, 渡辺美知子, 木下正博, 山本雅人, 鈴木育男. メタヒューリスティクスとナチュラルコンピューティング. コロナ社, 2012.
 - [26] Arben Asllani and Alireza Lari. Using genetic algorithm for dynamic and multiple criteria web-site optimizations. *European journal of operational research*, Vol. 176, No. 3, pp. 1767–1777, 2007.
 - [27] Belarmino Adenso-Diaz and Manuel Laguna. Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research*, Vol. 54, No. 1, pp. 99–114, 2006.
 - [28] Steven P Coy, Bruce L Golden, George C Runger, and Edward A Wasil. Using experimental design to find effective parameter settings for heuristics. *Journal of Heuristics*, Vol. 7, No. 1, pp. 77–97, 2001.
 - [29] Verena Heidrich-Meisner and Christian Igel. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 401–408. ACM, 2009.
 - [30] Oded Maron and Andrew W Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. *Robotics Institute*, p. 263, 1993.
 - [31] Mauro Birattari, Thomas Stützle, Luis Paquete, and Klaus Varrentrapp. A racing algorithm for configuring metaheuristics. In *GECCO*, Vol. 2, pp. 11–18. Citeseer, 2002.
 - [32] E. Hansen. System and method for optimizing website visitor actions, November 30 2006. US Patent App. 11/140,288.
 - [33] Ron Kohavi, Randal M Henne, and Dan Sommerfield. Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data*

- mining, pp. 959–967. ACM, 2007.
- [34] 金澤貴彦, 安田恵一郎. 207 組合せ最適化問題における proximate optimality principle を考慮した tabu search の探索効率改善. 最適化シンポジウム講演論文集, Vol. 2002, No. 5, pp. 219–224, 2002.
- [35] 野々部宏司, 柳浦睦憲. 局所探索法とその拡張 タブー探索法を中心として. 計測と制御, Vol. 47, No. 6, pp. 493–499, 2008.
- [36] 菅民郎. Excel で学ぶ実験計画法シックスシグマと重回帰分析第 2 版. オーム社, 2012.
- [37] Gerald van Belle and Steven P Millard. Struts: Statistical rules of thumb. *Departments of Environmental Health and Biostatistics, University of Washington*, 1998.
- [38] 水本篤, 竹内理. 効果量と検定力分析入門 統計的検定を正しく使うために . 2011.
- [39] Jacob Cohen. A power primer. *Psychological bulletin*, Vol. 112, No. 1, p. 155, 1992.
- [40] 大村平. 実験計画と分散分析のはなし: 効率よい計画とデータ解析のコツ. 日科技連出版社, 2013.
- [41] Steven J Spear. Learning to lead at toyota. *harvard business review*, Vol. 82, No. 5, pp. 78–91, 2004.
- [42] C. Anderson. *Makers: The New Industrial Revolution*. Crown Publishing Group, 2012.