

テスト熟練者知見を基にした優先度付きテストケース 自動生成手法の提案

齋藤英美¹ 中川雄一郎¹

概要：複数機器を接続可能で機能も豊富である監視/集中制御装置では、検証する機器、機能の組合せが多く、テストケースが膨大になる。このためテスト設計の熟練者は、自身の知見に基づいて重大な不具合の検出効率が良い組合せを優先してテストケースを作成している場合が多い。しかし知見は暗黙知であるため、他者との分担や自動化が困難であり、テストケース作成に時間がかかる。本報告では、テスト対象製品の仕様情報を入力することで自動的に優先度付きテストケースを生成する手法を提案する。手法では、仕様情報をテスト熟練者が持つ知見に沿って形式化を行い、人による個別判断を排除して自動化を可能にする。提案手法を空調機の集中制御装置に適用し、作業者の熟練度によらず、従来と比較して1/10程度の時間で不具合検出効率の良いテストケースを生成できることを確認した。

キーワード：組合せテスト、優先度、テストケース生成、システムテスト

Proposal of the test case automatic generation method with the priority based on test expert knowledge

EMI SAITO^{†1} YUICHIRO NAKAGAWA^{†1}

1. はじめに

近年の組込みシステムでは、機器の多機能化や、IoT化による接続機器の増加などで、検証する接続機器、機能の組合せが増大している。組込みシステムでは、ソフトウェアを実際の機器上で動作させないと不具合がわからないことも多い。しかし機器の準備や機器上でのソフトウェア実行に時間がかかるため、手戻りによる負担を最小化するためにも、重大な不具合を早期に検出したいという要求がある。そこでテスト設計の熟練者は、自身の知見に基づき、重大な不具合を早期に検出可能である組合せを優先してテストケースを作成している場合も多い。しかし知見は暗黙知であるため、他者との分担や自動化が困難であり作業に時間がかかる。

テスト設計の熟練者に頼らずに重大な不具合を早期に検出可能にする方法の1つとして、テストケースに優先度を付与する手法が報告されている[1][2][3][4]。既存研究の多くは回帰テストに対する手法であり、ソースコードの情報に基づき優先度を算出する方法[1][2]や、以前のテスト結果から優先度を決定する方法などが報告されている[3]。これらは、回帰テストを対象とするためテストケースが与えられている前提であり、新規開発案件への適用は難しい。ほかに、組合せテスト[5]に優先度を付与する手法[4]があり、重みが設定された組合せテストの入力モデルからテストケースの生成と優先度付けをする。しかし重み付きの入力モデルは与えられている前提であり、モデルへの重み付与にはテスト設計の熟練者等による判断が必要とみられる。

本報告では、テスト熟練者の知見を反映した優先度付き

テストケース生成手法を提案する。テスト熟練者知見を汎化した手法となっており、新規開発にも活用可能である。これまでに接続機器が1つの場合において優先度を付与する手法を報告した[6]が、本報告では、複数機器接続時の優先度も含めて付与できるよう改良を行い、提案手法を評価した。2章では、テスト対象の想定、課題を述べる。3章はアプローチを説明する。4章では、提案手法を説明する。5章では、提案手法を実際のシステムテストに用いて評価を行う。6章はまとめである。

2. テスト対象と課題

2.1 対象の想定

本報告では、空調の集中制御装置を対象に、システムテストの説明を行う。空調は室外機、室内機、リモコンで構成される。加えて、複数室内機を一括管理するために、図1に示すように集中制御装置を用いることがある。集中制御装置は、図のように、接続されている機器に対して個別あるいは複数の機器に同時に温度調節や運転/停止などの操作を行うことができる。ビル、病院、ホテルなど大型施設では、空調を一括管理するために、集中制御装置を導入している場合が多い。

集中制御装置の機能の一つに、ユーザからの操作に応じて冷房運転、暖房運転などの運転を切り替える「運転モード機能」がある。通常は、図1のように暖房運転操作を行うと、室内機は暖房運転を開始する。しかし、暖房が使用できない機器（ここでは冷専機と呼ぶ）が接続されている場合、集中制御装置は暖房運転操作を受け付けられないよう動

作する。また、暖房運転操作を受け付けないためのソフトウェア的な設定（ここでは冷専固定と呼ぶ）を集中制御装置から指定した場合、室内機が冷専機であるかに関わらず、集中制御装置は暖房運転を受け付けないように動作する。集中制御装置は、接続される空調が持つ制約に応じて提供する機能の振る舞いを変更する必要がある。

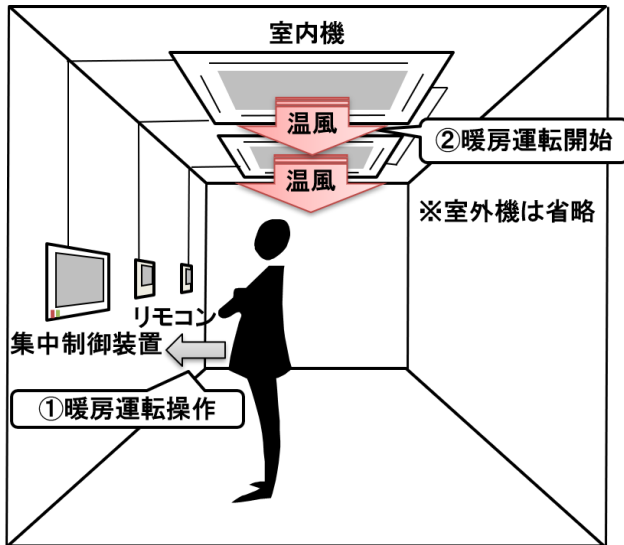


図 1 集中制御装置の使用例

Figure 1 Example of using central control unit

2.2 課題

集中制御装置に接続可能な空調機器、空調機器に設定可能な機能は多数あり、集中制御装置が提供する機器操作の機能も多数あるため、すべての組合せについて考えるとテストケースが大量になる。

集中制御装置が提供する機能 1 つにおいて、その機能に影響を与える機器および関連機能の組合せ数を概算する。

集中制御装置のテスト対象の特定機能に影響を与える接続機器の状態数を n とおく。接続機器の状態数 n は、機器の種類数 u と、機器に設定可能な関連機能数 f で表現できる。1 つの関連機能は、有効（設定されている）と無効（設定されていない）の 2 状態を持つとみなすと、接続機器の状態数 n は式(1)のように表現できる。

$$n = u \times 2^f \quad (1)$$

集中制御装置に同時に接続する機器の数を r とおく。異なる状態を持つ機器どうしを組合せると仮定すると、集中制御装置の最大接続数 $> n$ ならば、 r は、1 から n まで変化する。よって、組合せ数 N_{mul} は、式(2)、(3)を用いて計算できる。

$$N_{mul} = \sum_{r=1}^n n C_r = n C_1 + n C_2 + \dots + n C_n \quad (2)$$

$$n C_r = \frac{n!}{(n-r)! r!} \quad (3)$$

機器の状態数や組合せの考え方を図 2 に示す。空調の暖房運転、冷房運転の切り替えなどを操作する運転モードが

テスト対象機能の場合、テスト対象に影響を与える機器の例として冷専機がある。機器に設定可能な関連機能とは、前述の冷専固定や、冷房/暖房の自動切換え運転を許可する冷暖自動である。機器 1 つに対して状態数が n 個あり、それが接続機器数 r 分存在するときの組合せが N_{mul} である。

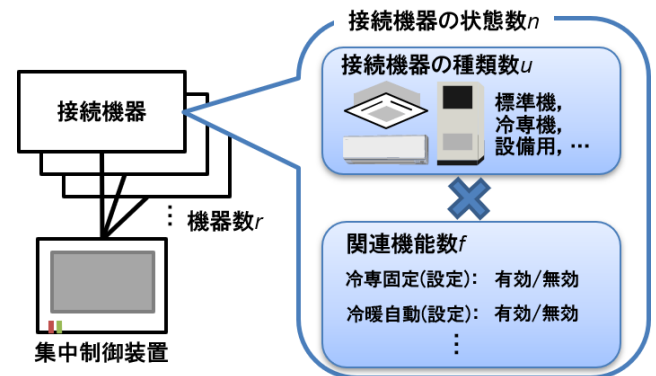


図 2 機器の状態と組合せの考え方

Figure 2 Image of combination and the state of unit.

ある集中制御装置において、接続機器のうちテスト対象とする機能に影響のある機器の種類は 4 つ ($u = 4$)、関連機能は 3 つ ($f = 3$) であった。式(1)を使うと、 $n = 32$ となる。よって組合せ数 N_{mul} は式(4)のように計算できる。

$$N_{mul} = \sum_{r=1}^{32} {}_{32}C_r = 4,294,967,295 \quad (4)$$

以上から、テストする 1 つの機能に対して影響のある機器に絞っても約 40 億通りの組合せが存在する。

なお、不具合の多くは 2 状態までの組合せで起きることが知られている[7]。そこで $r = 1 \sim 2$ に範囲を絞った場合は、 $N_{mul} = 528$ 通りの組合せに限定できる。それでも 1 機能に対して件数が多く、実際の機器を用いて行うテストの件数はさらに絞り込む必要がある。しかし絞り込み方によっては不具合の見逃し原因となるため、品質を維持したままテストの件数を絞り込むことが求められる。その手段として、テスト設計の熟練者が毎回重大な不具合を検出可能な組み合わせを判断してテストケースを作成する方法が存在するが、作業に時間がかかる。不具合検出効率の良いテストケースを短時間で用意する方法が必要とされる。

3. 解決アプローチ

3.1 解決方針

テストケースを短時間で用意するには、人の判断の介在を極力排除し、作業分担や自動化を可能にする必要がある。重大な不具合の検出効率が良いテストケースの判断に人を介在させないようにする方法として、優先度を付与したテストケースを自動生成する方法が考えられる。機器状態の組合せのテストに限定する場合、重みを設定した組み合わせテストの入力モデルから優先度付きの組合せテストを自

動生成する手法が報告されている[4]。このような手法ではモデルに付与した重みが優先度の判定に大きく影響を与えるが、重みを付与する方法に関しての言及はない。そのためテスト設計の熟練者が判断して重みを付与しなければ、所望の優先度を持つテストケースの生成が困難となってしまう、人の判断を排除することができない。

そこで本報告では、優先度を付与したテストケースを自動生成する手法を新規に構築する。手法が満たすべき要件を以下に示す。

要件(1) 期待結果を含むテストケースを生成する

組合せテスト手法は多数存在するが[5][7][8]、期待結果の生成は対象としていない。期待結果は製品仕様依存するため、手法で自動的に生成することが困難とされる。そのため、テストにおいて期待結果の作成は時間がかかる要因の一つとなっている。提案手法では、期待結果も自動的に生成可能とする。

要件(2) 生成するテストケース数を限定したうえで優先度の付与も行う

優先度を用いると、複数段階の優先度を付与できるため、テストの期間、内容、状況に応じて実施するテストケース数の調整が容易である。例えば最初に優先度の最も高いものを実施して重大な不具合を検出し、重大な不具合の修正後に次に優先度が高いテストを実施するといった段階分けを可能とする。しかし優先度を付与しても、テストケースの総数が多ければ、テストケースごとの優先度を確認することに時間がかかる。そこで、生成するテストケースの数を減らすことも同時に行う。

要件(3) 熟練者の知見を反映する

組み合わせ数を限定する組み合わせテスト手法の研究は多数あるが[5][7][8]、これらの手法は、どの組み合わせも不具合の検出率あるいは検出する不具合の重要性は同じであるとの前提条件がある。しかし、実際には組み合わせの重要性には差があると考えられる。テスト設計の熟練者は、重大な不具合を検出するテストケースについて知見を持っており、その知見に基づいてテストケースを作成することで、少ないテスト件数でも品質を維持していた。熟練者の知見を手法に取り入れることで、熟練者による作業をなくしても、テストケースの品質を保つことができる。

3.2 要件の実現方針

3.2.1 要件(1)の実現

集中制御装置の期待結果は、テスト対象機能を操作した際に動作可能な範囲と定義する。この範囲とは、空調の運転モードなら冷房運転や暖房運転、風量なら強、中、弱などが相当する。この場合に期待結果は、機器や設定が持つ

動作範囲への制約の組合せで決められる。動作範囲の拡大・縮小に絞って期待結果を考えると、期待結果の法則を単純にでき、手法で期待結果を生成可能になる。

なお、テストケースは、{事前条件、イベント、期待結果}で構成する。事前条件は、接続する機器や関連機能の組合せである。この事前条件は、組合せテスト[5]形式で表現する。組合せテストとは、テストの入力条件を因子と水準(因子の具体的な選択肢)[5]で表現し、異なる因子間の水準を組合せる手法である。因子間に関連性がない場合には組合せテスト手法のうちペアワイズ法[7]が有効だが、因子の組合せによって提供機能の動作範囲が変わる集中制御装置の例では、因子間に関連性があるためデジジョンテーブル[5]の活用が有効である。このとき、水準は FALSE, TRUE の 2 値で表現する。イベントは、期待結果を得るためのトリガーであり、ユーザによる「テスト対象機能の操作」が該当する。例えば、暖房から冷房へ運転を切り替えるために、運転モード画面を開いて運転モードを選択する操作である。本報告では自明なものとして、イベントは手法による生成対象から除外している。

3.2.2 要件(2)の実現

重みで優先度を調整しやすいため、重み付けによる優先度付与の方式を用いる。特定のルールに該当するごとに重みを付与し、その重み合計値を評価することで優先度を決定する。

なお組合せ数の増大を防ぐ方法の 1 つとして、因子を限定して組合せテストを生成し、生成結果と残りの因子を組合せて再度組合せテストを生成する手法が報告されている[8]。提案する手法でも 2 段階に分けて組合せテストを生成することで生成されるテストケース数を絞り込む。具体的には、集中制御装置への接続機器が 1 つの場合と接続機器が複数の場合で 2 段階に分ける。提案手法では、接続機器が 1 つのテストケースを単数機器テストケース、接続機器が複数のテストケースを複数機器テストケースと呼ぶ。単数機器テストケースを絞り込み、再度組み合わせることで複数機器テストケースを生成することで、テストケース生成数を削減する。単数機器テストケースの絞り込みは、単数機器テストケースの優先度に基づいて行うとする。

3.2.3 要件(3)の実現

テスト設計の熟練者の知見は暗黙知であり、ヒアリングだけで明らかにすることは難しい。そこで、過去にテスト熟練者が作成した理想的なテストケースから、以下の知見抽出手順に従うことで知見を抽出し、ポリシーという形で汎化する。手法への反映を容易化する。

Step(1): 集中制御装置に対する接続機器と関連機能の組合せに対して、過去のテスト熟練者が実際に作成していたテストケースをマッピングし、差分を抽出する。

Step(2) : Step(1)の差分発生理由の仮説を立て、テストの熟練者にヒアリングを行う。

Step(3) : 差分発生理由を基に、優先するテストのポリシーを導く。

4. 優先度付きテストケース生成手法の構築

4.1 手法の概要

要件とその実現方針を反映した手法として、図3に示す優先度付きテストケース生成手法を提案する。製品仕様を因子と水準、制約等で表現しなおすことで、期待結果を含むテストケースを自動生成可能にし、要件(1)を実現する。単数機器テストケースと複数機器テストケースの2段階に分けて組み合わせ生成と優先度付与を行うことで、テストケースの生成数を限定し、要件(2)を実現する。知見抽出手順を実施して導いたポリシーを重み付与のルールと重み合計値の評価基準に反映することで要件(3)を実現する。

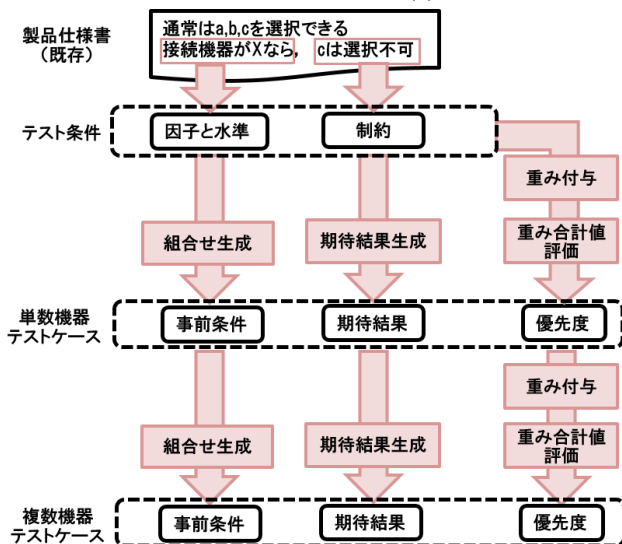


図3 優先度付テストケース生成手法概要

Figure 3 Overview of prioritized test case generation method.

要件(3)の実現のためには、テスト設計の熟練者の知見を抽出しポリシーを導く必要がある。テスト設計の熟練者に、過去のテストケースの中から理想的なテストケースを選定してもらい、熟練者の知見抽出手順を実施すると、以下のポリシーが明らかになった。

● 単数機器テストケースのポリシー

- (A) 利用頻度が高い利用形態におけるテストを優先する
- (B) (A)に該当しない場合でも、不具合発生可能性が高いテストは優先する

● 複数機器テストケースのポリシー

単数機器テストケースのポリシーを踏襲するが、さらに以下のポリシーを追加する

- (C) 単数機器で確認できない内容を優先する
- (D) 単数機器テストケースのテストに利用した構成を優先

する

(A)は、よく利用される形態での不具合は利用者への影響が大きく、重大な不具合であると判断している。利用形態とは、集中制御装置に接続する機器やその機器に設定されている機能の組合せで表現できる。例えば、標準機種で、特殊な設定を付けていない状態が最も利用頻度が高い。

(B)に該当するケースとしては、矛盾する制約の組合せが挙げられる。冷専機では暖房が利用できないため、冷房と暖房を切り替える冷暖自動を設定するケースはほぼ存在しない。しかし、実際に冷専機に冷暖自動を設定してしまった場合、互いの動作が矛盾するので、不具合にならないか確認する必要がある。

複数機器テストケースでは基本的に単数機器テストケースのポリシー(A)(B)を踏襲するが、複数機器テストケースに特有のポリシーとして(C)(D)が存在する。

(C)は、例えば異なる機器を同時に接続するテストは単数機器では行えないので、異なる機器同士の組合せはなるべく優先することをさす。

(D)は、テスト環境の用意に時間がかかるため、単数機器テストケースの確認に利用した機器を複数機器テストケースでも再利用するという内容である。(C)で異なる機器同士の組合せを優先としたが、単数機器テストケースで非優先とみなしているテストケースにおける機器を利用するテストケースは、複数機器テストケースでも非優先である。

4.2 手法の内容と利用方法

図3に示す手法のそれぞれの要素について、以下に説明を行う。

4.2.1 テスト条件

手法では製品の仕様書を基に、テスト条件を定義する。この作業に関しては手作業で行う。しかし図4に示すようなフォーマットに則ることで、人による判断の差異はほとんど発生せず、作業時間も短縮できる。

テスト条件表				テスト対象					
				運転モード					
				範囲					
				冷房	ドライ	送風	暖房	冷暖自動	
				基準範囲					
				○	○	○	○		
#	分類	因子	水準	制約					
1	F	冷専機	TRUE				▲	▲	
not 1			FALSE						
2	P	冷暖自動	TRUE					◎	
not 2			FALSE						
3	P	冷専固定	TRUE				▲	▲	
not 3			FALSE						

図4 テスト条件

Figure 4 Test condition.

テスト条件には以下を含む。

(a) テスト対象

テスト対象の機能名である。便宜上設けているが手法として必須ではない。

(b) 範囲

テスト対象機能の最大動作範囲である。図の例ではテスト対象の運転モードに対し、冷房、ドライ、送風、暖房、冷暖自動が最大範囲である。

(c) 基準範囲

すべての因子の水準が FALSE である場合の動作範囲を基準として記載する。空調の標準機種では、冷房、ドライ、送風、暖房が利用できるとして、その欄に○を記載する。

(d) 分類

機種に該当する因子には F, 事前に機器に設定しておく関連機能の因子は P, 運転中に集中制御装置から変更可能な関連機能の因子は V に分類する。冷専機などの機種は F, 冷専固定、冷暖自動などの事前に機種に設定する関連機能は P、テスト時に風向や風量などの運転中に集中制御装置等から変更する必要がある機能は V に分類する。

(e) 因子と水準

因子は、テスト対象機能の動作を制約する要因の名称である。例えば、冷専機などの機器名、冷専固定などの関連機能名である。因子が TRUE か FALSE の 2 値の水準をとるように定義する。TRUE だとその因子が有効、すなわち設定されている/接続されている状態である。冷専機が TRUE の場合、集中制御装置の接続機器が冷専機である状態をさす。

(f) 制約

各因子に対して、水準が TRUE/FALSE のときにテスト対象機能の動作範囲に与える影響を定義する。▲を書いた欄は動作範囲から除外する、◎を記載した欄は動作範囲に追加するという意味を持つ。因子が冷専機で水準が TRUE なら、暖房運転及び冷暖自動運転が利用できないため、範囲のうち暖房と冷暖自動の欄に▲を記載することで、範囲からの除外を定義する。

4.2.2 単数機器テストケース

単数機器テストケースは図 5 に示す形態で定義する。本テストケースは、テスト条件に対し、組合せ生成、期待結果生成、重み付与、重み合計値評価を実施して生成する。

	事前条件			期待結果					重み	優先度
	#1 F	#2 P	#3 P	冷房	ドライ	送風	暖房	冷暖自動		
1	FALSE	FALSE	FALSE	●	●	●	●		0	高
2	FALSE	FALSE	TRUE	●	●	●	●		-2	高
3	FALSE	TRUE	FALSE	●	●	●	●	●	-2	高
4	FALSE	TRUE	TRUE	●	●	●	●		-5	中
5	TRUE	FALSE	FALSE	●	●	●	●		-2	高
6	TRUE	FALSE	TRUE	●	●	●	●		-6	低
7	TRUE	TRUE	FALSE	●	●	●	●		-5	中
8	TRUE	TRUE	TRUE	●	●	●	●		-9	低

図 5 単数機器テストケース
 Figure 5 Single unit test cases.

(1) 組合せ生成

単数機器のテストケース生成では、異なる因子間で水準の組合せを網羅し、事前条件を生成する。なお、分類 F が複数ある場合、1 つの機器に対するテストで同時に分類 F が TRUE のテストはできないので、分類 F 同士の組合せは禁止である。そのような因子間の組合せの禁則を事前に定義しておき、禁則を除いて組合せを生成する。

(2) 期待結果生成

制約を基に、因子間の水準組合せに対する期待結果を生成する。期待結果の生成は、制約と基準範囲を組合せ、▲に該当する範囲は基準範囲から除外し、◎に該当する範囲は基準範囲に追加することで実現する。異なる制約が重なる場合には、制約の優位性に従って優先する制約を決める。制約は▲と◎の 2 種類で表現しているの、▲と◎が重なった場合にどちらを優先するか、制約の優位性を製品仕様を基に定義しておく。本報告で対象としている集中制御装置では範囲を削除する制約のほうが追加よりも強いため、▲と◎が重なる範囲は▲を優先するよう手法で規定している。テスト対象製品の仕様に応じて制約の優位性は変更してもよい。

(3) 重み付与と合計値評価

単数機器の優先度付与方法を説明する。優先度は図 6 のような仕組みで、(a)~(c)のルールに該当するたびにテストケースに重みを付与し、重みの合計値を評価し優先度を付与する。

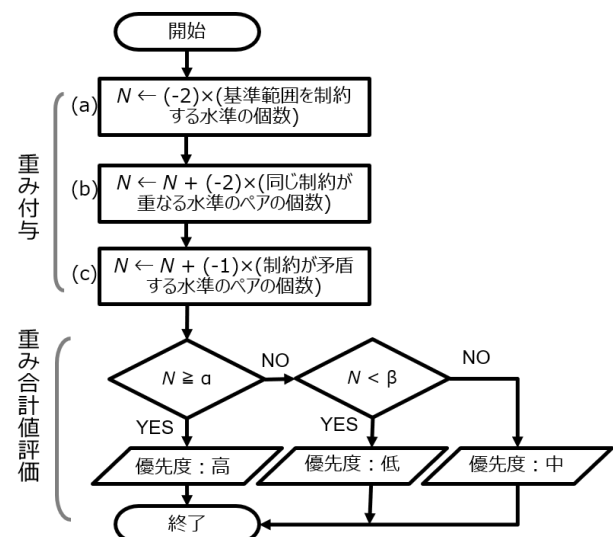


図 6 単数機器テストケースへの優先度付与方式

Figure 6 Method to add priority to single unit test cases.

図の N は重みの合計値である。重み付与のルール(a)~(c)について説明する。

(a) 標準からずれるほど重みを減らす. 具体的には, 事前条件に True の数が多いほど重みを減らす.

(b) 組合せに同じ制約が重複すると重みを減らす. 組合せる制約の同じ範囲に▲がある場合に該当する.

(c) 矛盾する制約の数だけ重みを減らす. 例えば, 組合せる制約どうしで同じ範囲に▲と◎がある場合に減らす.

(a),(b),(c)は単数機器テストケースのポリシー(A)を反映している. ポリシー(B)は(c)で反映しており, 矛盾する制約の組合せでは, (a)や(b)に対してより重みの減らし方を少なくする. よって, 図のように(a),(b)では-2, (c)では-1の重みを付与することで差をつけている.

優先度は, 重みの合計値があらかじめ定義した閾値を超えたかどうかで評価する. 閾値を α と β で表現する. 重みの合計値が α 以上なら優先度を高, β 未満なら優先度を低, α 未満で β 以上なら優先度を中と評価する. 3段階の優先度は, 優先度“高”を最優先で実施して重大な不具合を検出して修正を行い, 次に優先度“中”のテストを実施して各機能のより細かい不具合調査を行うという使い方を可能にしている. α と β の目安は, $\alpha=-2$, $\beta=-5$ としている. この数値はテスト対象機能やテストの規模に応じて調整してよい. 優先度を2段階にする場合には, α と β に同一の値を設定する.

4.2.3 複数機器テストケース

複数機器テストケースは図7に示す形態で定義する. 本テストケースは, 単数機器テストケースに対して, 組合せ生成, 期待結果生成, 重み付与, 重み合計値評価を実施して生成する.

	事前条件						期待結果				重み	優先度	
	機器1			機器2			冷房	ドライ	送風	暖房			冷暖自動
1	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	●	●	●	●		0	高
2	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	●	●	●			0	高
3	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	●	●	●			-2	低

図7 複数機器テストケース

Figure 7 Multiple unit test cases

(1) 組合せ生成

複数機器の組合せ件数は多いため, 組合せを生成する段階で件数を絞り込む. 抽出した熟練者の知見をもとに, 以下の手順によって組合せ数を絞り込むことを実現した.

Step(1): 単数機器の優先度“高”(重み合計値 α 以上)のテストケースを選出する.

Step(2): 単数機器テストケースで期待結果が同じであるテストケースは同値クラスとおき, 同値クラスの中から代表のテストケースを選出する. 代表テストケースは, 同値クラスで最も重み大きいテストケースを選出する. 重み大きいテストケースが複数あれば, 事前条件の分類Fの因子がTRUEになっているテストケースを選出する. それでも複数存在する場合は, 複数の代表テストケースを選出してよい.

Step(3): 異なる同値クラスから選出した代表テストケースを2つずつ組合せて, テストの事前条件を生成する.

上記の手順は, 複数機器テストケースのポリシー(C)(D)を反映している.

(2) 期待結果生成

複数機器を集中制御装置から操作する場合, 複数機器の動作範囲の共通範囲だけ選択可能にすることが多い. よって, 組合せる単数機器テストケースの期待結果どうしのANDをとることで, 複数機器テストケースの期待結果を導出する.

なお, 集中制御装置の設計思想によっては, 複数機器の動作範囲を合算した範囲を選択可能にする場合もある. その場合には, 組合せる単数機器テストケースの期待結果どうしのORをとることで, 期待結果を生成可能である.

(3) 重み付与と合計値評価

優先度付与に関するフローチャートを図8に示す.

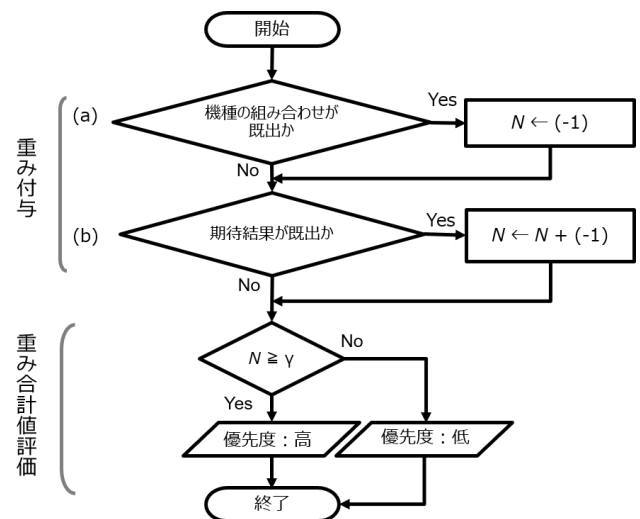


図8 複数機器テストケースへの優先度付与方式

Figure 8 Method to add priority to multiple unit test cases.

重みは, テストケース生成順に以下のように付与する.

(a) 期待結果が既出なら重みに-1を付与する.

(b) 事前条件の機種同士の組合せが既出なら重みに-1を付与する.

(a)と(b)は, 複数機器テストケースの生成順に優先度を付与していくとし, 該当テストケースよりも前に生成した複数機器テストケースにおいて, 該当テストケースと同じ期待結果, 機種組み合わせが出現しているかどうかで重みを減らす. 上記の重み付与ルールは, 複数機器テストケースのポリシー(C)(D)を反映している.

重み合計値が γ 以上の時に優先度“高”, γ 未満を“低”と評価する. 複数機器テストケースでは, 組合せ生成で十分に生成されるテストケースが絞り込まれているため, 優先

度“中”の該当判定は行わず，“高”“低”の2種類で優先度を表現する．本手法では $\gamma = -1$ とする．

5. 評価

5.1 手法の評価方法

実現した手法を新規の集中制御装置開発におけるシステムテストの一部に適用し，提案手法の効果を確認する．手法評価方法を図 9 に示す．評価は，新製品の一部の機能を対象とし，提案手法と従来手法によるシステムテストの結果の比較で行う．従来手法とは，製品仕様書を基に，テスト設計の熟練者が直接テスト仕様書を作成する方法である．提案手法は非熟練者が利用し，従来手法は，テスト設計の熟練者が利用する．なお，従来手法で作成されるテスト成果物に形態を合わせるため，提案手法で生成したテストケースはテスト仕様書の形態に手作業で書き直すことも行った．テスト仕様書とは，テストケースとテストの実行手順が併記されたものである．

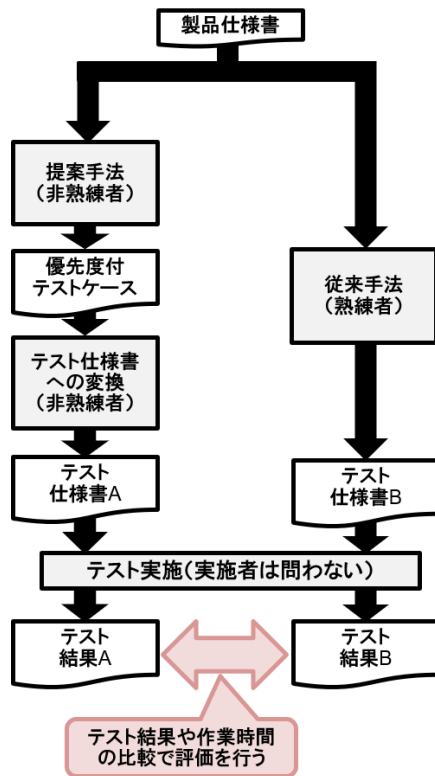


図 9 手法評価実験

Figure 9 Evaluation experiment of the proposed method.

提案手法は，Excel VBA を用いてツールとして実装して利用する．ツールは，テスト条件表の入力によって，自動で優先度付きの単数機器テストケース，複数機器テストケースを生成する．

5.2 評価結果

手法評価実験の結果を示す．まず，提案手法によるテ

ストケース生成結果を表 1 に示す．重み合計の評価基準は， $\alpha = -2$ ， $\beta = -5$ ， $\gamma = -1$ としている．

表 1 生成テストケース数

Table 1 The number of generated test cases

優先度	単数機器テスト ケース[件]	複数機器テスト ケース[件]
高	7	6
中	8	-
低	17	9

単数機器テストケースは全 32 件生成され，優先度“高”が 7 件，“中”が 8 件であった．複数機器テストケースは全 15 件生成され，優先度“高”が 6 件であった．

今回，従来手法で作成したテスト仕様書は，提案手法で作成したテスト仕様書とは内容が大きく異なっていた．新規開発のため仕様に変化があること，知見抽出の対象とした熟練者とは異なる者が従来手法を用いたことが理由としてあげられる．テストケースの直接の比較は困難であったため，概数で比較する．従来手法における 1 機能に対するテスト件数は数十件程度であり，今回生成したテストケースと同程度の数である．2.2 節の課題では，接続機器数を 2 つまでに絞っても 528 通りの組合せがあるとしていたが，提案手法によってテスト実施が可能な数十件の数まで絞り込むことができている．

次に，作業時間の比較を表 2 に示す．

表 2 テスト作業時間の比較

Table 2 Comparison of the test design time.

提案手法[h]	テスト仕様書へ の変換[h]	従来手法 [h]
0.5	5	78

提案手法によるテストケース生成時間は 0.5 時間であった．テストケース生成は実装したツールによって自動化ができるので，テスト条件の記載にかかった時間が 0.5 時間となっている．生成したテストケースを基にテスト仕様書を作成する作業には 10 倍の時間を要した．テスト仕様書は，ある程度の自由記述ができる形式のため，テスト手順の記載方法の検討に時間を要した形となった．一方で，従来手法は製品仕様書から直接テスト仕様書を作成しており，78 時間を要している．熟練者がテストケースを考えながら直接テスト仕様書を作成している場合と比べ，提案手法の活用で 1/10 以下にまで作業時間を削減できることが分かった．

提案手法と従来手法それぞれにおける不具合検出数の比較を表 3 に示す．なお提案手法では，優先度“中”まで

のテストケースを実施する想定としたうえで、実際には不具合の検出漏れ有無の確認のためにすべてのテストケースを実施した。

[8]M. Ozcan, "Applications of Practical Combinatorial Testing Methods at Siemens Industry Inc.", IWCT 2017, Mar. 2017.

表 3 不具合検出数

Table 3 Number of defects detected.

利用手法	不具合検出数[件]
提案手法	1
従来手法	0

従来手法では不具合検出数が 0 件なのに対し、提案手法では 1 件不具合を検出した。不具合を検出したのは、優先度“中”の単数機器テストケースであった。検出した不具合の内容は、ユーザへの影響はさほど大きくなかったことから、優先度“中”という設定は正しいといえる。従来手法では検出できていない不具合を検出したことから、過去の理想的なテストケースを設計した熟練者の知見を、提案手法によって非熟練者が他の製品開発で再利用可能になったといえる。以上から、短時間で非熟練者でも高品質なテストを作成でき、本報告の課題を解決できる見込みを得た。

6. おわりに

熟練者の知見を取り入れた優先度を付与するテストケース生成手法を提案した。提案手法を空調の集中制御装置のシステムテストに適用し、従来と比較して短時間で不具合検出効率の良いテストケースを生成できることを確認した。今後は IoT 化に伴い、接続機器の組合せが増大する問題はさらに深刻化すると考えられる。本手法は空調以外にも有効と考えられるため、手法の適用範囲拡大の検討も行っていく。

参考文献

- [1]G. Rothermel, R. H. Untch, C. Chu, and M. J. Harrold, "Prioritizing Test Cases For Regression Testing," IEEE Trans. Software Eng., vol. 27, no. 10, Oct. 2001.
- [2]加藤正恭, 小川秀人, "テスト戦略のためのテストケース優先度付けと可視化手法," 信学技報, vol. 117, no. 136, pp. 151-156, Jul. 2017.
- [3]B. Qu, C. Nie, B. Xu, X. Zhang, "Test Case Prioritization for Black Box Testing," Proc. 31st Annual IEEE International Comput. Software and Appl. Conf., vol. 1, pp. 465-474, Beijing, Jul. 2007.
- [4]E. Choi, T. Kitamura, C. Artho, and Y. Oiwa, "Design of Prioritized N-Wise Testing," Proc. International Conf. on Testing Software and Syst., pp. 186-191., Madrid, Sep. 2014.
- [5]秋山浩一, "ソフトウェアテストの最新動向 3. 組合せテストの作成," 情報処理, vol. 49, No. 2, Feb. 2008.
- [6]斎藤英美, 利國愛, 中川雄一郎, "システムテストに対するユーザの利用形態に基づく優先度付きテストケース生成手法の提案," 信学技法, vol. 117, no. 477, pp. 69 -74, Feb. 2018.
- [7]土屋達弘, 菊野亨, "ペアワイズテスト-ソフトウェアテストの効率化を求めて-", 電子情報通信学会論文誌 D, Vol. J90-D, No. 10, pp. 2663-2674, Oct. 2007.