

NAIST-IS-DD1161013

## 博士論文

効率のよいコンテキストウェアシステム実現のため  
の最適化アルゴリズムに関する研究

水本 旭洋

2014年 3月 13日

奈良先端科学技術大学院大学  
情報科学研究科 情報処理学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に  
博士(工学) 授与の要件として提出した博士論文である。

水本 旭洋

審査委員：

安本 慶一 教授 (主指導教員)

伊藤 実 教授 (副指導教員)

関 浩之 教授 (副指導教員)

# 効率のよいコンテキストウェアシステム実現のための 最適化アルゴリズムに関する研究\*

水本 旭洋

## 内容梗概

近年、安価なセンサ・モバイルデバイスの普及により、様々な情報を実時間で取得可能になってきた。これらの情報から実世界の現在の状況（コンテキスト）を認識し、コンテキストの変化に適応するように動作するコンテキストウェアシステムを実現する研究が行われている。

コンテキストウェアシステムを実現する研究として、様々なセンサの情報から抽象度の高いコンテキストを推定するコンテキスト推定、コンテキストウェアシステムの実装を容易にするために位置、種類、粒度などが異なるセンサの情報を形式化するコンテキストモデリング、現実の環境で動作する様々なコンテキストウェアシステムを開発する研究などが行われている。コンテキストウェアシステムは、ユーザの好み、時間、エネルギーなど、提供するサービスに関する様々な制約を考慮した上で、効率良く最適にサービスを提供することが望まれる。しかしながら、コンテキストウェアシステムにおけるサービスの効率と実用性を実現することを目的とした研究はほとんど行われていない。本論文では、災害時の医療支援や家庭での省エネ支援という近年関心が高まっている2つの分野において、効率的かつ実用的なコンテキストウェアシステムを実現するために、それぞれのシステムで提供されるサービスを最適化問題として定式化する方法およびそれを解くアルゴリズムを提案する。

---

\* 奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文, NAIST-IS-DD1161013, 2014年3月13日.

第一に、災害時の医療支援に対して、効率的なコンテキストウェアシステムを実現するために、生体センサと通信機能を備えた電子トリアージタグを利用した搬送計画手法の提案を行う。災害時の既存の搬送計画では、傷病者の容態変化に対応できない。また、最も重症な傷病者から搬送されるとは限らないという問題がある。そこで、実時間で傷病者の情報を取得できる電子トリアージタグと傷病者の生存率予測法を組み合わせ、傷病者の容態変化に自動適応する搬送計画を行い、傷病者の救命率を向上させるコンテキストウェアシステムの実現を目指す。上記のコンテキストウェアシステムを実現するために、容体の悪化とともに減少して行く「生存率」という指標に着目し、傷病者が命をとりとめる可能性が高い生存率を $\alpha$ として、傷病者の予測生存率が $\alpha$ 以上の間に限られた医療資源を用いて医療機関に到着する傷病者数を最大化するような傷病者搬送計画問題を定義し、問題を解決するような計画が行える最適化アルゴリズムを提案する。提案手法は、算出した予測生存率を救命救急で一般的に用いられるカーラーの救命曲線に当てはめ、各傷病者の予測生存率曲線を作成する。そして作成した予測生存率曲線や傷病者の位置情報、救急車数、医療機関の病床数などの情報を利用して搬送計画を行う。しかしながら、傷病者搬送計画問題は、傷病者数や救急車数、救護所の数などが増加する程、解の数が爆発的に増加するNP困難問題であるため、実用時間内に最適解を求めることができない。そこで、提案手法では、予測生存率推定曲線から予測生存率が $\alpha$ になる時間を算出し、その時間が早い順にグリーディに救急車を割り当てるようなヒューリスティックなアルゴリズムで短時間で搬送計画を行う。しかし、単純にグリーディに決定するだけでは、1人を助けることにより、より多数の傷病者が救命できなくなる可能性が発生する可能性がある。このようなケースを発生させないように、各順位の傷病者に対して、搬送する場合としない場合の両方を探索し、より救命者数の多くなる計画を行う。提案手法と既存の搬送計画を比較するため、大規模災害を想定し傷病者数や救急車数などを変化させたシミュレーションを行った。その結果、提案手法が比較手法と比べてより多くの傷病者を生存できる搬送計画を行えていることを確認した。

第二に、近年関心が高まり導入も進むスマートホームやスマートスペースにおいて効率的なコンテキストウェアシステムを実現するために、少ない消費電力

量で好みのコンテキストに遷移するよう家電を自動で制御する方法を提案する。既存のスマートスペースで動作するコンテキストウェアシステムの大半は、コンテキストを遷移させる時の消費電力を考慮しておらず、消費電力を考慮したシステムも、消費電力が基準値を超えた場合に警告を行うようなものであるため、消費電力を考慮してコンテキストを遷移させるようなシステムは存在しない。そこで、省エネルギーに様々なデバイスを自動操作し目標のコンテキストに遷移できるようなコンテキストウェアシステムを実現するため、あるコンテキストから任意の別のコンテキストへ遷移する最小コストのデバイスの操作系列を導出する問題を定義し、その問題を解く最適化アルゴリズムを実装した操作系列導出ツール *PathSim* を提案する。スマートスペースのコンテキストの遷移は、コンテキストをノード、コンテキストの各遷移をエッジとした有向グラフとして表すことができるため、対象とする問題は最短経路問題として扱うことができる。そこで、*PathSim* では、A\*アルゴリズムを基にしたアルゴリズムにより準最適なコストの操作系列を導出する。しかしながら、対象とするコンテキストの遷移を表すグラフでは、グラフの各辺が存在するか、また、存在する場合のコストがいくらか、といった情報が予め与えられていないため、A\*アルゴリズムをそのまま適用することはできない。そこで、提案手法では、探索を行う各コンテキストにおいて、隣接するコンテキストのエッジの存在およびそのコストをシミュレーション技術を用いて動的に導出し、A\*アルゴリズムに基づきコンテキスト空間の探索を繰り返すことで準最適なコストで目的のコンテキストに到達するパス（操作系列）を導出する。典型的なスマートホームの要求仕様を対象としたケーススタディにより、*PathSim* の有用性を評価した。その結果、*PathSim* で導出した操作系列は、単純なハイブリッドシステムの動作を想定して導出した操作系列より約26%少ない消費電力量で遷移できることを確認した。また、*PathSim* を実装したコンテキストウェアシステムを実装し実環境で評価を行った結果、*PathSim* で導出した操作系列で任意のコンテキストに実際に遷移できることを確認した。

## キーワード

ユビキタスコンピューティング, コンテキストウェアシステム, 最適化アルゴリ

ズム

# Study of Optimization Algorithm to Realize Efficient Context-Aware System\*

Teruhiro Mizumoto

## Abstract

Recently, it is getting possible to collect various information in real time owing to wide dissemination of inexpensive sensors and mobile devices. Many studies have been conducted regarding to *context-aware systems* which observe the current environment of real world (context) with sensors and operate while adapting to the context changes.

There are several research fields to realize context-aware systems such as *context recognition* for recognizing the context by analyzing sensor information, *context modeling* for formalizing factors comprising context and their structure such as locations, types, and granularity that are different among services provided, and development of various context-aware systems intended for actual deployment. Since context-aware systems are closely related to users' daily lives, it is important to provide each of provided services efficiently and optimally taking into account restrictions related to the service such as user's preference and comfort, operation time, and energy consumption. However, there are few studies that focus on efficiency and optimality in services provided by the context-aware system. This thesis describes two methods to realize efficient context-aware systems for disaster medical support and home energy-saving through formalization of optimization problems and development of algorithms to solve them.

---

\* Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1161013, March 13, 2014.

First, we propose a transportation scheduling method using electronic triage tag to realize an efficient context-aware system for disaster medical support. Existing transportation scheduling methods in disaster situations cannot adapt to patient's condition changes. In addition, the existing method cannot always transport patients in the order of seriousness of their damage. Therefore, we aim to realize a context-aware system improving lifesaving rate of patients by the transportation scheduling method that automatically reflects patient's condition changes by electronic triage tag for collecting patient information in real-time, and the survival rate prediction method. In order to realize this context-aware system, we define the minimum survival rate threshold,  $\alpha$ , and formalize the patient transportation scheduling problem that maximizes the number of patients arriving at a hospital while the patient survival rate is higher than  $\alpha$ . Then, we propose an optimization algorithm for the problem. Our method predicts the survival rate curve by fitting the survival rate calculated from the vital signs sensed from the e-triage tag to Golden Hour Principle that is commonly used in emergency medical support. Then, we schedule patients transportation by using information such as predicted survival rate curves, patient locations, the number of ambulances, and the number of acceptable patients in hospitals. However, since the patient transportation scheduling problem is NP-hard, we cannot derive the optimal solution within practical time. Therefore, our algorithm schedules the patient transportation in short time by calculating the time that the survival rate becomes  $\alpha$  from the predicted survival rate curve and allocating the patients to the ambulances in the increasing order of their time. In order to prevent the case that rescuing a patient earlier results in the death of two or more patients, our proposed algorithm also considers, for each low survival probability patient, the two cases of rescuing the patient or not and derives the transportation order that keeps the most patients alive. Through simulations assuming probable disaster situations, we confirmed that the proposed method can transport larger number of patients to the hospital before their expected survival probability gets



lower than a marginal probability than conventional methods.

Secondly, we propose a method that automatically operates devices in smartspace for moving between two arbitrary contexts with the minimal energy consumption in order to realize the energy-efficient context-aware system for smarthomes and smartspaces. Most existing context-aware systems operating in smartspaces do not consider the optimality of energy consumption to move the context. In addition, existing context-aware systems taking into account energy consumption is too simple, such as only warning. As far as we know, there is no system which can move to the target context with the minimum energy consumption. In order to realize the context-aware system that can move to target context by operating various devices with low energy, we define a problem that derives the device operation sequence with minimal cost for moving from a context to a target context, and propose a tool called *PathSim* which derives a minimal cost sequence of device operations for moving between two arbitrary contexts. Since we can represent the behavior of a smartspace by a directed graph with nodes corresponding to contexts, and edges connecting contexts represented by transition between these contexts, the target problem is regarded as a *single-pair shortest path problem*. Therefore, we derive the minimal cost sequence by an algorithm based on A\* algorithm that can efficiently solve the problem. However, we cannot directly apply A\* algorithm to the problem because we do not know the existence of each edge and the cost in the graph in advance. Therefore, our method dynamically explores the existence of the edge and the cost to each of neighboring nodes and derives the minimal cost path to the target context, by searching the context space based on A\* algorithm. Through a computer simulation assuming winter and summer in Japan, we confirmed that PathSim's device operations consume about 26% less energy consumption than a simple hybrid dynamical system. In addition, through a real environment experiment, we confirmed that PathSim's device operations can move the context in real space to the target context with almost similar energy consumption to simulation results.

**Keywords:**

Ubiquitous Computing, Context-Aware System, Optimization Algorithm

# 目次

1. 序論	1
2. 災害時における傷病者搬送計画を効果的に行うコンテキストウェアシステムの実現法	8
2.1 はじめに	8
2.2 関連研究	8
2.3 多数傷病者搬送計画問題	12
2.3.1 前提条件	12
2.3.2 問題設定	12
2.4 搬送計画システム	15
2.5 搬送計画手法	17
2.5.1 予測生存率推定関数	17
2.5.2 基本アルゴリズム	19
2.5.3 深さ制限付全探索 ( <i>DkBFS</i> ) アルゴリズム	20
2.6 シミュレーション実験と評価	21
2.6.1 シミュレーションの設定	21
2.6.2 評価項目と比較手法	23
2.6.3 実験結果と評価	25
2.7 まとめ	29
3. スマートハウスにおけるコンテキスト遷移を効率的に行うコンテキストウェアシステムの実現法	31
3.1 はじめに	31
3.2 スマートスペースにおける最小コストデバイス操作問題	31
3.2.1 スマートスペースのコンテキスト	32
3.2.2 コンテキスト遷移	33
3.2.3 問題定義	34
3.3 PathSim : 準最適デバイス操作シーケンス導出ツール	35

3.3.1	非環境イベントシーケンスの導出法 . . . . .	36
3.3.2	環境イベントシーケンスの導出法 . . . . .	36
3.3.3	イベントジェネレータ . . . . .	38
3.3.4	コンテキスト遷移シミュレータ . . . . .	39
3.3.5	コスト関数 . . . . .	41
3.3.6	PathFinder : 環境イベントシーケンス導出アルゴリズム . . . . .	42
3.3.7	非環境/環境イベントシーケンスの結合 . . . . .	46
3.4	シミュレーションによる評価 . . . . .	46
3.4.1	対象とするスマートホームおよび使用するデバイス . . . . .	46
3.4.2	要求仕様およびソース/対象コンテキスト . . . . .	48
3.4.3	実験方法 . . . . .	49
3.4.4	実験結果 . . . . .	52
3.5	実環境実験による評価 . . . . .	54
3.5.1	開発したコンテキストウェアシステム . . . . .	54
3.5.2	有効性の評価 . . . . .	55
3.6	関連研究 . . . . .	57
3.7	まとめ . . . . .	59
<b>4.</b>	<b>結論</b>	<b>60</b>
	謝辞	62
	参考文献	63

## 目 次

1	トリアージタグ . . . . .	9
2	e-Triage (フル機能版タグ) . . . . .	9
3	e-Triage (小型軽量版タグ) . . . . .	9
4	傷病者搬送計画問題のネットワークモデル . . . . .	13
5	ハミルトン路問題から傷病者搬送計画問題への変換 . . . . .	15
6	搬送計画システムの構成と処理 . . . . .	16
7	傷病者の予測生存率減少例 . . . . .	19
8	CASE1: 急変確率を変化させた結果 . . . . .	25
9	CASE2: 各傷病者に割り当てる $a$ の割合を変化させた結果 . . . . .	25
10	CASE3: 追加する傷病者数を変化させた結果 . . . . .	26
11	CASE4: 救急車の数を変化させた結果 . . . . .	26
12	$v_{nei}$ の導出例と $v_{cur}$ の選択例 . . . . .	44
13	$v_{nei}$ が既に <i>OPEN</i> または <i>CLOSE</i> に含まれているケース . . . . .	44
14	対象とするスマートホーム . . . . .	47
15	比較手法における各デバイスのオートマトン . . . . .	51
16	冬想定時のデバイス操作シーケンスの消費電力量および遷移時間 . . . . .	52
17	夏想定時のデバイス操作シーケンスの消費電力量および遷移時間 . . . . .	53

## 表 目 次

1	TRISS 係数 . . . . .	18
2	想定する災害状況に関する設定 . . . . .	21
3	傷病者に関する設定 . . . . .	21
4	各ケースに関する設定 . . . . .	22
5	統計検定の結果 (CASE1) . . . . .	27
6	統計検定の結果 (CASE2) . . . . .	27
7	統計検定の結果 (CASE3) . . . . .	28
8	統計検定の結果 (CASE4) . . . . .	28

9	実験で使用するデバイスの設定 . . . . .	48
10	HDS に対する消費電力量の削減率 . . . . .	54
11	実環境実験で生成されたコンテキスト (デバイスの状態は全て OFF に設定したため省略) . . . . .	56
12	実環境実験の結果 . . . . .	56

# 1. 序論

本論文は、筆者が奈良先端科学技術大学院大学情報科学研究科に在学中に行った効率の良いコンテキストウェアシステムを実現する最適化アルゴリズムに関する研究をまとめたものである。

近年、センサ技術の進歩やデバイスの低価格化により、実世界の情報を実時間で収集できるようになってきた。これらの情報を基に実世界の環境やユーザの活動状況といった**コンテキスト**の変化（遷移）を認識し、適応するように動作（処理）するシステムは**コンテキストウェアシステム** [1] と呼ばれている。コンテキストウェアシステムは、情報を収集するために実世界に埋め込まれた多数のセンサと、情報を収集しコンテキストを生成・認識し、ユーザにサービスを提供するソフトウェア、そして、ユーザにサービスを提供するために動作する様々なデバイスで構成される。コンテキストウェアシステムにおいて、各センサ・デバイスはネットワークに接続され、提供するサービスによって情報の送信や操作が行われる。そして、サービスを提供するソフトウェアは、情報を収集し、実時間で現在のコンテキストを認識し、コンテキストに適応するようにサービスを提供する。単純なコンテキストウェアシステムの例として、カーナビゲーションシステムが挙げられる。カーナビゲーションシステムは、現在の位置情報（コンテキスト）をGPSセンサで収集し、実時間で変化する現在地を把握しながら、目的地へのルートを提供し続けるため、コンテキストウェアシステムの一つである。

近年、このようなコンテキストウェアシステムを実現するための様々な研究が行われており、実環境の状態（コンテキスト）をモデル化するコンテキストモデリング [2, 3, 4, 5, 6, 7, 8]、実環境上に存在するモバイルデバイスやセンサデバイスから取得した情報を基に現在のコンテキストの状態を推定するコンテキスト推定 [9, 11, 10]、コンテキストに合わせユーザにサービスを提供するコンテキストウェアシステム [12, 13, 14, 15, 16, 17, 18, 19] に関する研究などが主に行われている。

コンテキストモデリングに関する研究では、収集できるセンサデータの形式の差を緩和し、コンテキストウェアシステムの開発を容易にするために、実環境のコンテキストを形式的に表現する様々な方式が提案されている。これらの研究

では基本的なモデルとして、コンテキストを構成する各属性をオブジェクトで表現するオブジェクトロールベースモデル [3, 4], 場所 (空間) ごとにコンテキストを表現する空間モデル [5, 6], 記述論理によって表現を行うオントロジーベースモデル [7, 8] が提案されている。

コンテキスト推定に関する研究では, 加速度センサのデータを基にユーザの現在の状態を推定する手法 [9], Kinect センサを利用して人の行動を推定する手法 [11], 複数のスマートフォン (およびセンサ) を利用してユーザのコンテキストを推定する手法 [10], のようにセンサデータから直接認識できないコンテキストを推定する手法などが研究されている。

コンテキストウェアシステムの研究として, ナビゲーションシステムやスマートホーム上で動作するシステムなど様々な応用システムを提案する研究が行われている。ナビゲーションシステムに関する研究では, 近年, ルートを提示し続けるような単純なナビゲーションシステムだけではなく, ユーザの嗜好やルート上に設置された環境センサや気象情報からユーザにとって快適なルートをコンテキストウェアに選択するナビゲーションシステムが複数提案されている [14, 15]. スマートホームに関する研究として, 在宅医療支援システムや高齢者の見守りシステム [16, 17], 実時間でスマートホーム内の消費電力量や電気料金を提示するシステム [19] やスマートホーム内の環境を調整するシステム [18] などが提案されている。このようなコンテキストウェアシステムは, ユーザの日常生活と密接に関わっているため, ユーザの嗜好や快適さ, 消費される時間やエネルギーなどの提供を行うサービスに関連した制約を考慮し, 効率良く最適にサービスを提供できるようにすることが重要である。しかしながら, 既存のコンテキストウェアシステムに関する研究では, ナビゲーションシステムを除き, 本論文で対象とする最適化を取り扱うものはほとんどなく, 提案されているシステムが効率の良いシステムとはいえない。また, 最適化手法に関しては, 現実の問題が既知の最適化問題と完全に一致することは珍しいため, 本研究で対象とするコンテキストウェアシステムを直接最適化する手法は存在しない。そこで本研究では, コンテキストウェアシステムが有用な現実問題のモデル化と解決する最適化アルゴリズムの検討を行うことで, 実用的かつ効果的なコンテキストウェアシステム



の実現を目指す。

コンテキストウェアシステムを適用する分野として、本研究では、災害時の医療支援と家庭での省エネ支援を対象とする。近年、東日本大震災を発端に、災害支援や電力使用についての問題が浮き彫りになった。災害支援に関しては、災害直後の傷病者が多数発生する状況において、限られた医療資源の中で最大多数の人を救命しなければならないため、リアルタイムに変化する災害現場の状況に応じた医療支援システムが望まれている。また、大震災後の電力問題から国民の節電意識が高まり、家庭内での電力使用を最適に行えるようなシステムが望まれている。そこで本研究では、上記の災害時の医療支援、家庭での省エネ支援を対象とした効率の良いコンテキストウェアシステムを最適化アルゴリズムにより実現する研究を行った。

第一に、近年注目が高まっている災害時支援に対して実時間で変化する傷病者の容態に基づいてコンテキストウェアに効果的な傷病者の搬送計画を行えるコンテキストウェアシステムを実現するための手法について研究を行った。近年、多数の死傷者が発生する自然災害、テロ事件、大規模事故などが世界中で頻発している。これら大災害や事件、事故では、医療従事者の人数、救急車の数、医療機関の収容能力といった医療資源の数を上回る傷病者が発生し、指揮・救援・医療系統の混乱を引き起こす。その結果、二次三次の被害として「避けられた死」が少なからず発生する。こういった災害、事件や事故を総じて**多数傷病者事故** (Mass Casualty Incident, 以下 **MCI** と記す) と呼ぶ [20]。MCI 状況下では、医療資源が不足するため、全ての傷病者に医療資源を平等に割り当てることができない。そこで、各傷病者に対して、治療や搬送の優先順位を短時間で決める**トリアージ**が行われる。トリアージは、傷病者の重症度を1人当たり30秒以内に4つのカテゴリに分類し、紙製のトリアージタグに記載し、傷病者につけて結果を表示する。しかし、現在の、紙製のトリアージタグでは、判定時の人為的なミスや容態の変化に対応できないことや傷病者の位置が把握できないといった大きな問題がある。また、標準のトリアージでは4段階にしか重症度を分類できないため、同一カテゴリに分類された傷病者の間で優先順位を決定できない。従って、MCI という傷病者の容態を把握しづらい状況において、同一カテゴリの傷病者が多い場合、

最も重症な傷病者から搬送されるとは限らない。災害医療における救命率および救助効率を改善するために、世界各国で数多くの研究が行われている。電子トリアージ・システムプロジェクト [21] では、センサ機器が埋め込まれたトリアージタグ（電子トリアージタグ）が開発されている。電子トリアージタグを用いれば、心拍数、呼吸数、血中酸素濃度といった人体の生体情報をリアルタイムに取得でき、さらに無線により送信することで、医療サーバで一括管理できる。これにより、タグを装着する際の判定ミスを避け、タグ装着後の容態変化を知ることができるようになり、トリアージ段階のミスを最小限に抑えることができる。一方、傷病者の容態及び外傷部位から、すぐに治療が施される場合の生存率を予測する TRISS 法 [22, 23] などの手法が救命救急に応用されている [24]。この予測生存率は、実際の死亡率と関連しており、医療機関での治療時に予測生存率が 30% 未満であった傷病者の 75% 以上が、実際に死亡していることが明らかになっている。そのため、電子トリアージタグと予測生存率を組み合わせることで、より救命率を高める傷病者の搬送計画を実現できる可能性がある。本研究では、限られた数の救急車を用いて、多数の傷病者を医療機関に搬送する傷病者搬送計画問題を解決する。予測生存率と実際の死亡率の関係から、傷病者を救命するためには、医療機関での治療時における予測生存率がある程度高い状態で搬送する必要がある。傷病者の容態は医療機関で治療を受けるまでの時間が長くなるほど悪化するため、予測生存率も時間と共に低下していくと考えられる。そこで、本研究では傷病者が命をとりとめる可能性がある生存率の下限を  $\alpha$  と定義し、傷病者の予測生存率が  $\alpha$  未満になる時間（搬送限界時刻）までに医療機関に到着しなければ生存の可能性がないと考える。従って傷病者搬送計画問題は、医療機関の収容能力の制約を満たしながら、予測生存率が  $\alpha$  以上の間に医療機関に到着する傷病者の数を最大化することを目的とする。この問題の解決のため、電子トリアージタグを用いた傷病者搬送計画システム及び、搬送計画手法の提案を行う。MCI のように搬送する傷病者が多くなる状況では、傷病者の搬送待ち時間が長くなることから、予測生存率の低下が早い傷病者を先に搬送したい。そのため、各傷病者の予測生存率がどのように減少するかモデルを作成する必要がある。生存率が時間と共に変化するモデルとしてカーラーの救命曲線 [25, 26] やドリンカーの生存

曲線 [27] などが存在する。これらのモデルに近似する曲線を予測生存率曲線として作成し搬送計画に用いる。提案システムでは、まず応急救護所でのトリアージ時に記録された医療従事者の所見と、電子トリアージタグからリアルタイムに得られる生体情報を TRISS 法に当てはめ、予測生存率を算出する。算出された予測生存率と傷病者の搬送待ち時間、そして搬送待ちの間に計測、算出された予測生存率をカーラーの救命曲線に当てはめ、各傷病者の予測生存率曲線（**予測生存率推定関数**）を作成する。そして、傷病者の予測生存率推定関数や位置情報、救急車や医療機関の情報などを利用して傷病者の搬送計画を算出する。傷病者搬送計画問題は、医療機関、救急車、現場救護所、各現場における傷病者数が増加する程、組み合わせ数が爆発的に多くなる NP 困難問題であるため、実用時間内に最適解を算出できない。提案手法では、予測生存率推定関数を用いて予測生存率がちょうど  $\alpha$  になる搬送限界時刻を算出し、その時間が早い傷病者から順にグリーディに救急車を割り当てることで、短時間で傷病者の搬送リストを作成する。しかし、単純にグリーディな手法で搬送リストを決定するだけでは、1 人を助けることにより、より多数の傷病者が救命できなくなるケースが発生する可能性がある。このようなケースを発生させないように、提案手法では各順位の傷病者に対し、搬送する場合としない場合の両方を探索し、より救命者数の多くなる搬送リストを作成する。提案手法は、傷病者の容態の変化や追加など実時間で変化する被災地のコンテキストに適応するように搬送計画の修正を繰り返す。提案手法と既存の搬送計画手法とを比較するため、大規模災害を想定したシミュレーションを行った。その結果、提案手法は他の手法と比べてより多くの傷病者を、生存に最低限必要な予測生存率として設定した 30% 以上を保った状態で搬送できることを確認した。

第二に、近年関心が高まり導入も進むスマートホームにおいて省エネルギーにユーザが望むコンテキストに遷移するように家電を操作するようなコンテキストウェアシステムを実現する手法について研究を行った。ユビキタスコンピューティングシステム [42, 43] が実装された空間や家はスマートスペース（またはスマートホーム） [44] と呼ばれている。スマートスペースには、センサデバイスや家電デバイスなどが分散して設置されている。センサデバイスはスマートスペー

スを構成する各空間の温度や湿度といった空間の物理的特性，ユーザの位置情報や行動などの状況をセンシングする．また各デバイスは，スマートスペース内のユーザにサービスを提供するためにネットワークを通して操作される．これらの物理的特性，ユーザの状況，そして家電デバイスの状態などのそれぞれを変数として表したとき，これら変数の値の組はスマートスペースのコンテキストとして扱うことができる．スマートスペースがユーザに適切なサービスを提供できるように，ユーザは自分の好みに合わせた要求仕様を設定する必要がある．ユーザの要求はコンテキストを構成する各変数の値の範囲を用いて定義される．要求仕様は，“ユーザAが部屋Bに居る場合に，温度を 25℃，湿度を 50%に保つ”のようなシステムが動作するトリガーとなる条件（以下，**事前条件**と表す）とシステムの動作によって提供または維持される状態の条件（以下，**事後条件**と表す）の組（性質）で与えられる．要求仕様に基づいたサービスを実現するために，スマートスペースは，事前条件を満たすコンテキストから，事後条件を満たすコンテキストに到達するようにデバイスを操作する必要がある．一般的に，複数のデバイスがスマートスペース内に設置されている場合，あるコンテキストから異なるコンテキストにスマートスペースを遷移させるデバイスの操作シーケンスは多数存在する．そして，シーケンスごとに操作するデバイスや操作する順番などが異なるため，消費電力量や動作時間がそれぞれ異なる．例えば，サービスが要求仕様に基づき，温度 15℃と湿度 30% (15℃, 30%) のコンテキストから，(24℃, 50%) のような異なるコンテキストに遷移させる場合を考える．この時，スマートスペース内に，エアコン，オイルヒータ，そして加湿器が設置されていると，温度を上げるために，エアコンやオイルヒータが使用できるため，エアコンのみを使用する場合や，オイルヒータのみを使用する場合，また両方使用する場合といった異なる制御シーケンスが考えられる．また，デバイスを制御する順番や動作時間によっても異なる制御シーケンスが存在する．実際に，スマートスペースには多くのデバイスが設置されるため，膨大な数のシーケンスが存在する．そのため，この中から最適または準最適なコストのシーケンスを自動的に見つけることが出来れば，ユーザにとって非常に有益だと考えられる．本研究では，省エネルギーにコンテキストアウェアなデバイス操作を実現することを目的とし，スマートスペース

において、あるコンテキストから任意の別のコンテキストへ遷移する準最適コストのデバイスの操作シーケンスを導出するツール *PathSim* を提案する。PathSim は、A\*アルゴリズムを基としたアルゴリズムにより準最適コストのシーケンスを導出する。A\*アルゴリズムは有限個のノードとコストが付されたエッジで構成されたグラフに対してのみ適用可能なアルゴリズムである。スマートスペースのコンテキストの遷移は、各コンテキストをノードの集合  $V$ 、コンテキスト間の遷移をエッジの集合  $E$  とする有向グラフとして表すことができる。しかしながら、一般的にコンテキストの各変数を取りうる値の範囲は連続または無限であるため、そのままではA\*アルゴリズムを適用することができない。そこで、A\*アルゴリズムを適用できるように、コンテキストを構成する各属性(変数)の値域を有限個の範囲に分割し、離散的なコンテキスト遷移モデルとして表す。また、A\*アルゴリズムが適用されるグラフでは、ノード間を結ぶエッジ(およびコスト)の集合が予め与えられる必要があるが、本研究が対象とするコンテキストの遷移では、エッジの存在およびエッジのコストが未知である。そのため、探索を行う各コンテキストにおいて、隣接するコンテキストへのエッジの存在およびそのコストを動的に調べる必要がある。PathSim は、各コンテキストにおいて実行可能なデバイスの操作(イベントと呼ぶ)をシミュレーションにより模擬実行することで、イベントによって遷移する隣接コンテキストと遷移に必要なコストを導出する。そして、A\*アルゴリズムに基づきコンテキストの探索を繰り返すことで準最適コストで目的のコンテキストに到達するパス(操作シーケンス)を導出する。日本の夏と冬を想定した計算機シミュレーションにより PathSim の有用性を評価した。結果、PathSim で導出したシーケンスは、単純なハイブリッドシステムでコンテキストを遷移させる場合と比べ、消費電力量を約 26%削減できることを確認した。また、PathSim を導入したコンテキストウェアシステムを開発し、実環境実験を行った結果、導出した操作シーケンスを用いて様々なコンテキストに実際に遷移できることを確認した。

## 2. 災害時における傷病者搬送計画を効果的に行うコンテキストウェアシステムの実現法

### 2.1 はじめに

本章では、大規模災害や事故などの多数の傷病者が発生する状況において、実時間で変動する被災地の状況にコンテキストウェアに効果的な搬送計画を行える搬送計画システムおよび搬送計画手法について述べる。本システムでは電子トリアージタグを用いてリアルタイムに取得した傷病者の生体情報と医者の方見などから、生存率の時間減少モデルを作成し、搬送計画アルゴリズムを用いて計画を行う。提案アルゴリズムでは、救命のための限界時間が早い傷病者から順に、最も早く搬送できる救急車と搬送先医療機関を割り当てる。ある傷病者を先に搬送することにより、他の多数の傷病者が救命できないケースを避けるため、各傷病者に対し搬送する場合と搬送しない場合の両方を効率良く探索する。大規模災害を想定したシミュレーション実験を行うことで、提案手法と既存の搬送手法を比較し、提案手法の有用性を確認した。

### 2.2 関連研究

医療資源が傷病者の数を上回る MCI では、傷病者の緊急度・重症度を評価し、搬送や治療の優先順位を決定するトリアージが行われる。トリアージでは、START 法 [28] に基づいて傷病者の容態を、緑（優先度：低、軽傷で専門的な治療が必要無い）、黄（優先度：中、多少治療が遅れても生存率に影響が無い）、赤（優先度：高、生命に関わる重篤な状態で、生存の可能性はある）、黒（すでに死亡しているか、生存の可能性が無い）という 4 つのカテゴリに分類し、簡単な所見と共に 1 のような紙のタグ（トリアージタグ）に記録し傷病者に取り付ける。また、トリアージは応急救護所での治療優先度を決定する 1 次トリアージと、応急処置後に医療機関への搬送順位を決定する 2 次トリアージに分けられる。現在は紙で作られたトリアージタグが使用されているため、(1) 傷病者の容体変化への対応

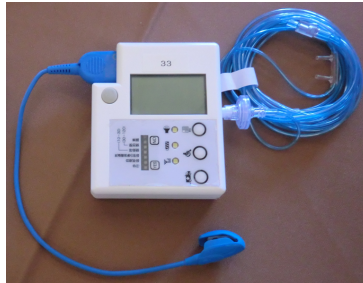


図1 トリアージタグ 図2 e-Triage (フル機能版) 図3 e-Triage (小型軽量版タグ)

が困難, (2) 同カテゴリー内で優先順位が不確定, (3) 短時間で判断するため誤判定が発生, (4) 医療従事者の重圧が大きい, などの問題がある.

現在の災害医療における傷病者搬送は, 紙のトリアージを使用していることから重傷者 (赤カテゴリ) から手当たり次第に搬送するのが一般的である. これは本質的に同一カテゴリ内でランダムに搬送することと同じである. 傷病者の搬送を指揮する指揮所は, 電話やトランシーバ, メモなどを使って傷病者の情報や医療機関の空き情報を収集しており, これらの情報に基づいて救急車の派遣先を決定する. しかし, 各応急救護所から傷病者の情報が逐次報告されるため, 計画を行う傷病者が増加するほど, 再計画や調整を行うことが難しく, 応急救護所の医師の判断や, 報告された順番などに従って傷病者を搬送することになる. そのため, 必ずしも救命率を最大化する順番にはなっていないケースが多いと考えられる. Jotshi らは, 収集する情報に誤りがあることを考慮した搬送計画手法を提案している [29]. この手法では, 地図上の傷病者が集中している地域をクラスタと定義し, クラスタごとの傷病者の人数, 救急車からの距離, クラスタから医療機関までの距離という3つの要素を考慮して救急車を派遣するクラスタを決定する. 傷病者にはそれぞれ, 赤か黄色のトリアージ結果が与えられている, すなわち, 重みが設定されているものとする. これにより, 赤色の傷病者が多いクラスタに救急車が派遣されやすくなっている. この手法は個々の傷病者の情報に誤りがあ

る場合に、無駄な救急車の派遣を防ぐことができる。しかし、傷病者が少ないクラスタに対して、救急車が派遣される可能性が少ないため、人数が少ないクラスタに所属する赤色の傷病者は放置される可能性が高くなったり、救急車から遠いクラスタへ派遣される可能性も低くなる。また、クラスタ内の傷病者の搬送順序の決定まで行われていない。

近年、紙のトリアージタグの問題を解決するために、トリアージタグの電子化に関する研究が世界中で活発に行われている。Gao らは AID-N プロジェクトにおいて、ETag というセンシング機器を実装した電子トリアージタグを開発している [30, 31]。また日本国内でも、災害時救命救急支援を目指した人間情報センシングシステムプロジェクトにおいて、電子トリアージタグを開発している [21, 32, 33, 34]。このプロジェクトで開発されている電子トリアージタグ e-Triage は、ETag と比べ小型の機器で生体情報を収集可能である。2 のフル機能版タグでは、呼吸数、脈拍、血中酸素濃度の測定が、3 の小型軽量版タグでは、脈拍、血中酸素濃度の測定が可能である。特に小型軽量版タグは指先に挟み込むだけで簡単に生体情報を測定でき、傷病者の装着負担を軽減することができる。e-Triage は IEEE802.15.4 規格の ZigBee を用いて、サーバにデータを送信できる。サーバで傷病者の情報を管理することで、全ての傷病者の生体情報を一目で確認でき、傷病者の生体情報の変化を監視することができる。このような、電子トリアージを用いたシステムの研究も行われており、Gunawan らは化学災害を想定し、電子トリアージタグを所持した歩行可能な傷病者（緑トリアージ）の移動を追跡し、適切な場所へと傷病者を避難させるシステムを提案している [35]。また、栖関らは電子トリアージから得られた生体情報から、傷病者の呼吸数や脈拍数が、START 法で定められている各色に分類するための呼吸数、脈拍数からどれだけ離れているかで、同じカテゴリに分類される傷病者に対して治療や搬送の優先度を決定する手法を提案している [36]。

医学の分野では、傷病者の重症度を評価するための手法が研究されており、これらを用いて傷病者の救命率や死亡率を予想することが可能になっている。評価手法として、生理学的指標と、解剖学的指標、そしてそれらを組み合わせた指標が存在する。その中でも生理学的指標である RTS[37] と解剖学的指標である ISS[38]



を組み合わせた TRISS 法 [24, 22, 23] が有名である。TRISS 法を用いることで傷病者の予測生存率が算出可能であり、この予測生存率が 30% 未満になった傷病者の 75% が実際に死亡していることが日本外傷データベース (JTDB) のデータから明らかになっている [39]。また、TRISS 法より正確な予測生存率を算出するための研究も行われており、33,950 例の外傷データベースを用いて提唱された HARM[40] や、NTDB に登録されている 702,229 人の患者のデータを用いて提唱された TMPM[41] などがある。これらの研究で用いられている外傷データベースのように、世界中で外傷データベースへの外傷データの登録が進んでおり、今後より多くの外傷データを使って評価手法が提唱されることが予想される。また、時間経過と死亡率の関係を表した評価指標が以前より提唱されており、カーラーの救命曲線がよく参考にされている。これによると、心臓停止は 3 分、呼吸停止は 10 分、多量出血は 30 分放置すると死亡率が 50% に達するとされている。福田らはカーラーの救命曲線を表す関数式を算出し、医療情報デジタル伝送システム導入時の人的損失額の算出や傷病者の情報を基地局まで搬送するために必要なアンテナの設置場所の検討に利用している [25, 26]。カーラーの救命曲線では、3 つの傷病に対する救命曲線しか見ることができない。しかし、今後電子トリアージタグの実用化が進み、時間経過に伴う生体情報の変化や予測生存率の変化を大量に記録することができれば、各傷病や生体情報に応じた救命曲線を導出できる可能性が高い。

本研究では、救急車による傷病者の搬送計画問題に着目し、電子トリアージタグから得られる傷病者の生体情報をパラメタとして、予測生存率や予測生存率の時間変化を推定する関数を算出し、最大多数の傷病者を搬送・救命できる搬送計画をリアルタイムに策定する手法を提案する。既存の搬送計画手法では、紙のトリアージタグを利用することを前提としているため、多人数の傷病者の容態をリアルタイムに監視することはできなかった。そのため、傷病者ごとの予測生存率の時間変化を考慮した搬送計画手法は存在しなかった。提案手法では、電子トリアージタグが計測した生体情報の変化から、時間経過に対する予測生存率推定関数を導出し利用することで、個々の傷病者の容態を考慮した搬送計画を行うことができる。また、脱線事故のような応急救護所が 1 箇所には存在しない場合だ

けではなく、複数の応急救護所が距離的に離れた位置に存在する地震災害の場合にも対応できるように、災害現場全体を見通した搬送計画を考える。

## 2.3 多数傷病者搬送計画問題

本節では、対象とする搬送計画問題に関して、前提条件を示したあと問題設定を行う。

### 2.3.1 前提条件

各傷病者  $p$  に対し、ある時刻  $t$  での予測生存率を算出できる関数  $P_s(p, t)$  が与えられるとする（関数の詳細は 2.5.1 項で述べる）。

日本外傷データバンクによると、傷病者の医療機関到着時刻での予測生存率が 10% 未満では 95%、20% 未満では 80%、30% 未満では 75% が実際に死亡してしまう。そのため、傷病者が最終的に生存するために最低限必要な予測生存率の閾値を  $\alpha$  とし、閾値未満に予測生存率が減少した傷病者については生存可能性がないものとして扱う。

### 2.3.2 問題設定

本研究で対象とする問題は、傷病者の予測生存率が  $\alpha$  以上の間に医療機関へ搬送できる人数を最大化する搬送スケジュールを算出する問題である。傷病者の集合を  $P$ 、現場の応急救護所の集合を  $S$ 、医療機関の集合を  $H$ 、救急車の集合を  $Am$  とする。被災地内には、多数の傷病者  $P$  が発生し、複数の応急救護所  $S$  のいずれかにそれぞれ搬送済とする。傷病者は応急救護所で処置された後、複数の医療機関  $H$  の中から最適と思われる場所へ搬送される。傷病者を搬送するために、複数の救急車  $Am$  が、応急救護所と医療機関の間を往復している。救護所  $s$  から医療機関  $h$  への片道搬送時間は、関数  $Tt(s, h)(s \in S, h \in H)$  で求めることができるとする。傷病者を現場から医療機関に搬送する過程は 4 のようなネットワークモデルで表現できる。ある時刻  $t$  に医療機関  $h_1$  に位置する救急車  $am$  が、救護所  $s$  の傷病者  $p$

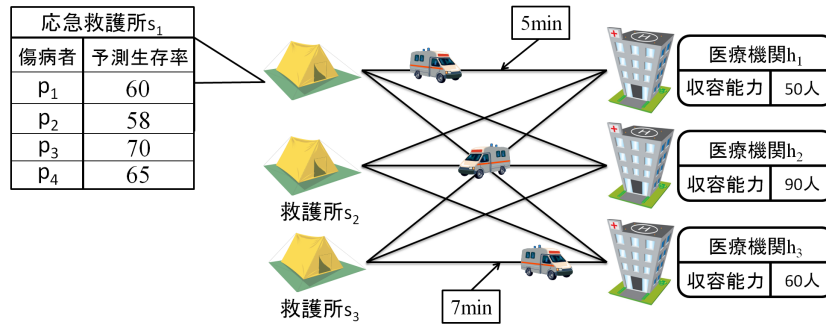


図 4 傷病者搬送計画問題のネットワークモデル

を医療機関  $h_2$  に搬送する場合の搬送完了時刻  $at$  は、 $at = t + Tt(h_1, s) + Tt(s, h_2)$  である。また前節の仮定より、傷病者  $p$  の搬送完了時刻  $at$  での予測生存率  $ps$  は関数  $Ps(p, at)$  を用いて算出できる。搬送計画により、傷病者の搬送情報リスト  $TL$  を作成する。搬送情報リスト  $TL$  は、どの救急車が、どの傷病者を、どの医療機関に搬送するかを表す搬送情報  $tl$  の集合である。搬送情報  $tl$  を、 $\langle$  傷病者  $p$ , 救急車  $am$ , 救急車の派遣元医療機関  $h_{start}$ , 派遣先救護所  $s$ , 搬送先医療機関  $h_{end}$ , 出発時刻  $st$ , 到着時刻  $at$  $\rangle$  と表記する。搬送情報  $tl$  で搬送される傷病者  $tl.p$  が閾値  $\alpha$  以上の予測生存率で医療機関に到着するかどうか（すなわち、治療後生存可能かどうか）を 1, 0 で表す関数を  $Survive(tl)$  とする。関数  $Survive(tl)$  を式 (1) に定義する。

$$Survive(tl) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } Ps(tl.p, tl.at) < \alpha \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

各医療機関  $h \in H$  に対し、收容可能傷病者数を  $h.cap$  と表記する。搬送情報リストの中で医療機関  $h$  が搬送先になっている搬送情報  $tl$  の数、すなわち、 $h$  に運ばれる傷病者数を関数  $count(TL, h)$  で表す。各医療機関  $h$  に運ばれる傷病者は、收容可能傷病者数  $h.cap$  以下でなければならない、この制約を式 (2) で表す。

$$\forall_h \in H, \text{count}(Am, h) \leq h.\text{cap} \quad (2)$$

式 (2) を満たしながら，予測生存率が  $\alpha$  以上の間に医療機関に搬送される傷病者数（以後，**救命者数**と呼ぶ）を最大化する搬送情報リスト  $TL$  を求めることが本問題の目的である．よって，本問題の目的関数を以下の式 (3) で定義する．

$$\begin{aligned} \text{Maximize : } & \sum_{tl \in TL} \text{Survive}(tl) \\ \text{subject to } & (2) \end{aligned} \quad (3)$$

定義した搬送計画問題が NP 困難であることを，NP 困難問題であることが分かっている最短ハミルトン路問題を本問題に帰着することで証明する．

最短ハミルトン路問題は，与えられた重み付きグラフ  $G$  において，全ての頂点を一回だけ通る重みの和が最小のパスを求める問題である．

$G = (V, E, \text{cost})$  を重み付き完全無向グラフとする．ここで， $V, E, \text{cost}$  は，それぞれ，頂点の集合，辺の集合，辺へのコスト関数  $\text{cost} : E \rightarrow \mathbf{R}$  である．

以下， $G$  を傷病者搬送計画問題のインスタンスに変形する．頂点集合  $V'$ ，辺集合  $E'$  を変数として用いる．最初， $V', E'$  は空集合である．

グラフ  $G$  において，各辺  $(u, v) \in E$  に対し，新たな頂点  $w$  を用いた 2 つの辺  $(u, w), (w, v)$  を作成し，辺集合  $E'$  の要素とする．新たな頂点  $w$  は，辺ごとに異なるものを用意し，それぞれ頂点集合  $V'$  に加える．これらから新たなグラフ  $G' = (V \cup V', E \cup E', \text{cost}')$  を構成する．ここで， $\text{cost}'$  は以下の条件が成立するよう  $E'$  の辺に重みを付与するコスト関数である．

$$\forall (u, v) \in E \exists w \in V' \text{cost}(u, v) = \text{cost}'(u, w) + \text{cost}'(w, v) \quad (4)$$

$$\forall u \in V \left( \prod_{(u, w) \in E'} \text{cost}'(u, w) \right) = 0 \quad (5)$$

式 (4) は，元の辺の重みが，新たに作成した 2 つの辺の重みの和となること，式 (5) は， $V$  の各頂点  $u$  を端点とする辺のうち，少なくとも一つの辺は重みが 0 でなければならないことを表している．

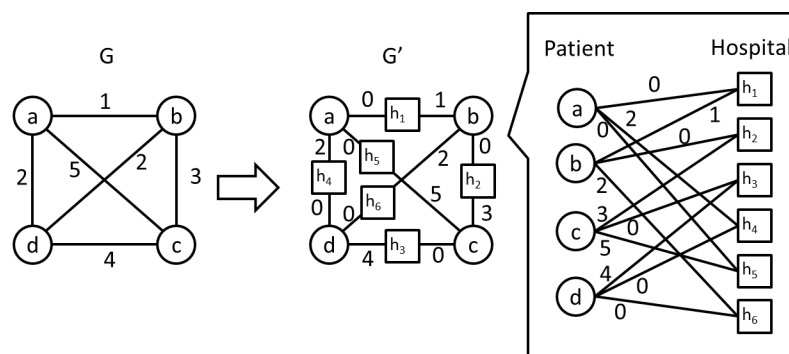


図 5 ハミルトン路問題から傷病者搬送計画問題への変換

図5に変換例を示す. 変換後のグラフ  $G'$  において, 頂点集合  $V, V'$  をそれぞれ, 傷病者, 病院に相当するとする (図では,  $V = \{a, b, c, d\}, V' = \{h_1, h_2, h_3, h_4, h_5, h_6\}$ ). また,  $E'$  の各辺の重みは, 救急車の移動時間とする. 各傷病者の予測生存率は経過時間に応じて単調に減少するものとし, 各傷病者の搬送完了時刻の予測生存率は  $\alpha$  未満にはならないとする.

図5より, グラフ  $G'$  において, 救急車 1 台を用いて, 搬送時刻での予測生存率の和を最大化する全ての傷病者 ( $V$ ) の病院への搬送順序を求める問題は, グラフ  $G$  において最短ハミルトン路を求める問題と等価であることは自明である. 以上より, 最短ハミルトン路問題は傷病者搬送問題の特殊な場合であり, 後者の問題は NP 困難である.

## 2.4 搬送計画システム

本節では, 傷病者搬送計画問題を解決するための, 電子トリアージを用いた搬送計画システムの構成について述べる.

搬送計画システムは指揮所で機能し, 応急救護所の電子トリアージから得られた傷病者情報, 医療機関から得られる収容可能人数の情報に基づき, 傷病者の搬送リストを作成し, 救急車に提供する. システムの構成と実行される処理を図6

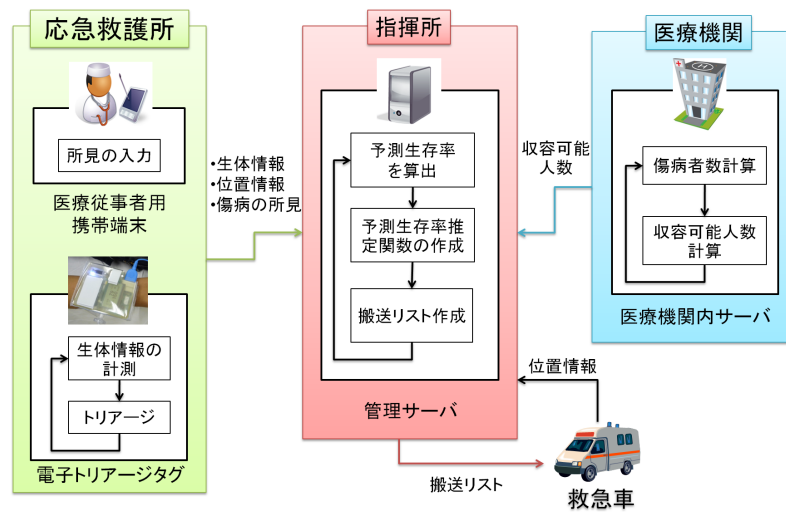


図 6 搬送計画システムの構成と処理

に示す。

各医療機関は収容可能人数が変化した場合にサーバに通知を行う。救急車は現在位置及び状況（傷病者を搬送中かどうか）、目的地のいずれかに変更があった場合に更新情報を指揮所に送信する。指揮所は救急車に対して次に搬送する傷病者の情報、応急救護所や医療機関などの情報を送信する。応急救護所では、傷病者に取り付けられた電子トリアージタグによって、生理学的指標を算出するための生体情報をリアルタイムに測定する。また、応急救護所内の医療従事者は治療または二次トリアージ時に、意識レベルなどの一部の生理学的指標に必要な情報および、解剖学的指標を算出するために必要な情報を、携帯端末を用いて入力する。そしてそれぞれサーバに情報を送信する。サーバは受信した情報を基に傷病者の予測生存率を算出する。予測生存率が算出された後、カーラーの救命曲線に近似する予測生存率推定関数を作成し、搬送計画を行う。予測生存率の算出、予測生存率推定関数の作成、そして搬送計画は、救急車が医療機関に到着する直前、極端に予測生存率が減少した傷病者を検出した場合、新たに傷病者が搬送待ち状態

に変わった場合、そして、現場の医療従事者の判断で搬送する傷病者が変わった場合などの被災地内の搬送計画に関するコンテキストが変化した場合に行う。

## 2.5 搬送計画手法

傷病者搬送計画問題はNP 困難であるため、最適解を実用時間で求めることは難しい。そのため、提案手法では、準最適解を短時間で求めるヒューリスティックアルゴリズムを用いて搬送計画を行う。本節では、傷病者の予測生存率の減少を推定する関数の作成方法とヒューリスティックアルゴリズムを提案する。まず、傷病者の予測生存率の減少を推定する予測生存率推定関数の作成方法を提案する。次に、予測生存率推定関数を用いて搬送限界時間を算出し早い傷病者から順に搬送するグリーディ法に基づいた**基本アルゴリズム**を提案する。最後に、基本アルゴリズムで算出した搬送順序に対して、一部の傷病者を搬送しないことによる救命数の向上を試みる**深さ制限付全探索アルゴリズム (DkBFS)**を提案する。これらの詳細を以下で順に説明する。

### 2.5.1 予測生存率推定関数

本研究では福田らが算出したカーラーの救命曲線式 (6) を参考に予測生存率推定関数を作成する。

$$R = 1 - \left( \frac{1}{1 + \exp(4.80861 - a\beta t)} \right) \quad (6)$$

ここで、 $R$  : 生存率を表す関数、 $\beta$  : 救命救急士の応急救護処置効果、 $t$  : 搬送時間、 $a$  : 定数項である。本研究では一般の救急医療とは違い、搬送前に応急救護所で応急処置が行われることを想定する。そのため、定数項  $a$  の値に救命救急士の応急救護処置効果が含まれると想定する。また、カーラーの救命曲線は受傷時点からの生存率を表しているが、本研究では応急救護所での治療の際に予測生存率が算出されており、既に応急処置も済まされていることを想定する。そのため、受傷した時点での予測生存率からではなく、応急処置が完了した時点（搬送

表 1 TRISS 係数

	$b_0$	$b_1$	$b_2$	$b_3$
鈍的外傷	0.4499	0.8085	0.0835	-1.734
鋭的外傷	2.5355	0.9934	0.0651	-1.136

待ち開始時点)での予測生存率から減少するように、パラメタを変更する。作成した予測生存率推定関数  $P_s(p, t)$  を式 (7) のように定義する。

$$P_s(p, t) = p.ps_0 \left( 1 - \frac{1}{1 + \exp(4.80861 - a \cdot t)} \right) \quad (7)$$

ここで、 $P_s(p, t)$  : 予測生存率推定関数、 $p.ps_0$  : 搬送待ち開始時点の予測生存率、 $a$  : 定数項、 $t$  : 時間である。定数項  $a$  を変化させると、曲線の下降速度が変化し、搬送限界時刻も変化する。そのため、各傷病者ごとの定数項  $a$  を算出することで、予測生存率推定関数を作成する。予測生存率推定関数の作成処理では、現在時刻や、現在の予測生存率、そして傷病者の搬送待ち開始時の予測生存率をそれぞれ式 (7) に代入し、定数項  $a$  を算出する。

現在の予測生存率は、1 の係数 ( $b_0, b_1, b_2, b_3$ ) と電子トリアージタグが取得した生体情報から算出した RTS, そして医者 の 所見 から 算出 した ISS を TRISS の式 (8) に代入することで算出する。

$$P_s = \left[ \frac{1}{1 + e^{-b}} \right] \quad (8)$$

$$b = b_0 + b_1 \times RTS + b_2 \times ISS + b_3 \times \text{Age}$$

予測生存率推定関数は、時々刻々と変化する傷病者の容態に対応するために、定期的に更新を行う。



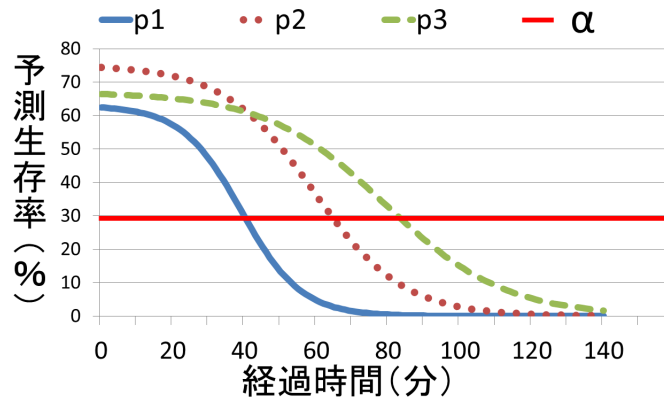


図 7 傷病者の予測生存率減少例

### 2.5.2 基本アルゴリズム

救命者数を増加させるため、傷病者の予測生存率が、救命のために最低限必要な予測生存率  $\alpha$  以上である間に医療機関に搬送したい。そのため、傷病者  $p$  の予測生存率が  $\alpha$  未満になる搬送限界時刻が早い傷病者から順に搬送を行う。この搬送限界時刻を算出するため、2.4 節で算出した予測生存率推定関数を用いて、搬送限界時刻  $t$  を各傷病者  $p$  ごとに求め、この時間の昇順に傷病者をソートした搬送順位リスト  $PL$  を作成する。このリストの上位ほど、早く搬送しなければ死亡してしまう可能性が高い。傷病者  $p_1, p_2, p_3$  の予測生存率推定関数が図 7 のように与えられた場合、 $\alpha$  を 30% と設定すると、傷病者の搬送順位リスト  $PL$  は  $[p_1, p_2, p_3]$  となる。

基本アルゴリズムは、 $PL$  の順序で、各傷病者に対し、空きのある最寄医療機関を選択し、最も搬送が早い救急車を割当てる。また、それぞれの傷病者が医療機関に搬送される時刻を計算し、その時刻における予測生存率を計算する。基本アルゴリズムでは、搬送順位リストの先頭から順に計画を行うため、1 人を助けることで、搬送順位が遅い 2 人以上が救命されないケースが発生する。これを解

決し、できるだけ多くの傷病者を救命されるように最適化を行うアルゴリズムを次節で提案する。

### 2.5.3 深さ制限付全探索 ( $DkBFS$ ) アルゴリズム

改善の方針は、「一人をあきらめることにより、二人以上多く助かる」可能性を探ることである。本アルゴリズムでは、予測生存率が  $\alpha$  未満になる時刻が早い順に全傷病者をソートした搬送順位リスト  $PL$  に対して、それぞれの傷病者が搬送される/されないの 2 通りを考慮する。この場合、全傷病者のそれぞれに対し 2 通り、すなわち、 $n$  人の傷病者に対し  $2^n$  通りの組合せについて救命者数を全て計算すれば、最適な搬送情報リストを求めることができる。しかし、 $n$  の数が大きくなると、実用時間内に最適解を求められない。そこで、搬送順位リスト  $PL$  の  $n$  人中  $k$  人ずつ全探索を行う深さ制限付き全探索アルゴリズムを示す。

(1)  $PL$  中の先頭の  $k$  ( $|PL| < k$  の時は、 $|PL|$ ) 人のそれぞれの傷病者が搬送される/されない場合、すなわち、 $2^k$  通りの組合せに対する救命者数を求める。その際、 $k + 1$  番目以降の傷病者については、基本アルゴリズムに従って搬送するかどうかを決める。

(2)  $2^k$  通りの中で最も良い解をもつ組合せに含まれる傷病者およびその順位を搬送順位として確定し、各傷病者の搬送情報を作成し搬送情報リスト  $TL$  に加える。

(3)  $PL$  から先頭の  $k$  人 ( $|PL| < k$  の時は、 $|PL|$  人) を取り除き、(1) から繰り返す。  $PL$  が空になったら終了する。

例えば、 $PL = [p_0, p_1, p_2, p_3, p_4, p_5]$ ,  $k = 3$  の時、 $p_0, p_1, p_2$  がそれぞれ搬送される/されない場合の全ての組合せ、 $\{\{p_0\}, \{p_1\}, \{p_2\}, \{p_0, p_1\}, \{p_0, p_2\}, \{p_1, p_2\}, \{p_0, p_1, p_2\}\}$  について考える。 $\{p_0\}$  の時、 $p_0$  は先頭  $k$  人に対する全ての組み合わせを探索するために、2.3.2 項の式 (1) を満たさなくても必ず搬送し、 $p_1, p_2$  は必ず搬送しない (黒カテゴリに仕分けされる)、残りの  $p_3, p_4, p_5$  に対しては、基本アルゴリズムに基づいて搬送計画を行い、予測生存率が  $\alpha$  以上で搬送される傷病者の人数と、予測生存率の平均を求める。同様の処理を、他の組合せに対しても行い、式 (3) の目的関数の値が最も高い組合せを、搬送情報リスト  $TL$  に追加する。そして、次の  $k$  人に対しても、組合せを作成し、同様の処理を行う。これ

表 2 想定する災害状況に関する設定

応急救護所	医療機関	各医療機関の収容能力	傷病者
10 箇所	5 箇所	80 人 / 医療機関	400 人

表 3 傷病者に関する設定

閾値 $\alpha$	初期 $P_s$	定数項 $a$	急変確率	急変時増加定数
30 (%)	60 – 80 (%)	0.06 – 0.12	0 – 20 (%)	0.01

を傷病者リストの最後になるまで繰り返す。

## 2.6 シミュレーション実験と評価

提案手法の性能を評価するため、大規模災害を想定したシミュレーション実験を行い、既存の搬送計画手法と比較を行う。

### 2.6.1 シミュレーションの設定

本実験では、近畿地方で大規模地震が発生した場合を想定し、奈良県生駒市内における傷病者の搬送をシミュレーションする。想定する災害状況の設定を表 2 に示す。

本実験では、複数の応急救護所が離れた位置に設置される地震災害を想定する。また医療機関 5 箇所については、奈良県生駒市周辺の救急科を持つ医療機関の位置情報を、応急救護所については、生駒市内に存在する 10 箇所の学校・総合公園の位置情報を参考に、救急車の片道移動時間を最短 3 分、最長 31 分に設定する。また、傷病者数を阪神淡路大震災時のある地方自治体における傷病者数を参考に 400 人と設定する。医療機関の収容能力は全ての傷病者が搬送可能であり、かつ、一つの医療機関では収容能力が不足するよう均等に 80 人と設定する。

表 4 各ケースに関する設定

	変化させた項目	設定した値
CASE1	急変確率	0%, 5%, 10%, 15%, 20% (他のケースでは 10%)
CASE2	各傷病者に割り当てる $a(0.12,0.10,0.08,0.06)$ の割合	{100,0,0,0},{50,50,0,0},{25,25,25,25}, {0,0,50,50}, {0,0,0,100} (他のケースでは {25,25,25,25})
CASE3	追加する傷病者数	2, 4, 6, 8, 10 人 / 20 分 (CASE1,2 では 4 人, CASE4 では 10 人)
CASE4	救急車数	10, 20, 30, 40, 50 台 (他のケースでは 30 台)

傷病者に関しては表 3 に示す設定を用いる。本実験では、救命のために最低限必要な予測生存率  $\alpha$  を 30% と設定する。これは JTDB の調査結果 [39] により 30% 未満で搬送された傷病者のほとんどが医療機関で治療を受けても死亡しているからである。傷病者の予測生存率の初期値は、搬送待ち状態になった時点（搬送待ち開始時点）とした場合、受傷から時間が経過しているため予測生存率が 100% であるとは考えられない。そこで、傷病者の初期予測生存率について、60% から 80% が妥当だと考え、各傷病者にこの範囲でランダムに設定する。本実験では、傷病者はカーラーの救命曲線に近似して時間と共に予測生存率が減少すると仮定する。そこで、応急処置を受けても血液を失い続けている多量出血の傷病者を最も緊急な傷病者と考え、30 分前後で予測生存率が 50% 未満に減少するように、式 (7) の定数項  $a$  を 0.12 と設定する。また、トリアージにおいて 1 時間以上搬送しなくても救命に支障がない傷病者は赤タグに分類されないため、赤タグ傷病者の中で緊急度の低い傷病者の定数項を、最低でも 1 時間前後で予測生存率が 50% 未満になる 0.06 と設定する。0.12 から 0.06 まで 0.02 ずつ減少させた 4 種類の定数項  $a$  (0.12,0.1,0.08,0.06) を設定した予測生存率算出関数を傷病者に設定する。

本実験では、様々な設定での提案手法の性能を評価するために、(1) 急変確率、

(2) 定数項  $a$  に関する分布, (3) 時間当たりに追加される傷病者数, (4) 救急車数, を変化させた複数のケースで実験を行い, 各ケースに対して救命者数 (予測生存率が  $\alpha$  以上の間に搬送された傷病者数) を評価する. 各ケースに関する設定を表 4 に示す.

傷病者は時間の経過と共にショック状態など傷病者の容態が急変する場合があります, 急変した場合は搬送限界時刻も早くなる. CASE1 では, 急変する傷病者が多い場合や存在しない場合での提案手法の性能を評価するため, 10 分ごとに急変確率 0% – 20% で急変を発生させる. また, CASE1 以外では急変確率を 10% とし設定する. 急変した傷病者に対して予測生存率推定関数の定数項  $a$  に 0.01 加算することで搬送限界時刻を早める. CASE2 では, 予測生存率の減少が早い (搬送限界時刻が短い) 傷病者が多い場合や少ない場合での提案手法の性能を評価するため, 予測生存率推定関数 4 種類 ( $a \in \{0.12, 0.10, 0.08, 0.06\}$ ) を  $\{100, 0, 0, 0\}$ ,  $\{50, 50, 0, 0\}$ ,  $\{25, 25, 25, 25\}$ ,  $\{0, 0, 50, 50\}$ ,  $\{0, 0, 0, 100\}$  という割合で割り当てる. CASE2 以外では, 25% ずつ均等に割り当てる. 傷病者は応急救護所で応急処置を受けた後, 医療機関に搬送されるために救急車を待つ, 搬送待ち状態になる. 搬送待ち状態の傷病者が多くなればなるほど, 救急車で効率よく搬送しなければ, 搬送が間に合わなくなり, 救命者数が低下する. CASE3 では, 各応急救護所に対して搬送待ち状態として追加する傷病者数を 2, 4, 6, 8, 10 人と変化させることにより, 提案手法の効率を評価する. また, 20 分ごとに設定した人数を搬送待ち状態として各応急救護所に追加する. CASE4 では 10 人ずつ, CASE1 と 2 では 4 人ずつ追加されるように設定する. 同様に救急車の台数が変化した時の提案手法の性能を評価するために, CASE4 では救急車を各医療機関に 2, 4, 6, 8, 10 台 (各合計 10, 20, 30, 40, 50 台) 設定する.

各ケースに対して 100 回実行した後, 評価項目の, 最大, 最小, 平均を算出し既存手法との比較を行う.

## 2.6.2 評価項目と比較手法

以下に実験で使用する比較手法の処理を示す.

**基本アルゴリズム (Base) および提案手法 (DkBFS)** : BASE は 2.5.2 項, 提

案手法 (DkBFS) は 2.5.3 項で定義したアルゴリズムを用いて計画を行う。BASE は単純に搬送順位リストの先頭から順に計画を行い、DkBFS は、深さ  $k$  で搬送される/されない場合を探索する。DkBFS は、搬送計画の対象となる傷病者数  $n$  人に対して  $2^n$  の組み合わせを全て計算すれば最適解を算出できる。しかし、傷病者数が多くなるほど計算時間が長くなることや、傷病者の人数や容態が変化するたびに再計画が必要であることから、搬送順位が遅い傷病者の計画が無駄になる。また、搬送計画は救急車が救護所や医療機関に到着するまでに行えば良いが、傷病者の追加や容態変化が急に起こった場合にすぐに再計画を行えるように、計算時間を短くしたい。そのため DkBFS では、搬送順位リストの先頭  $k$  人に対し全探索を行い、 $k$  人以降は BASE で一意に計画し、 $2^k$  個の組み合わせの中から最適解を求めることで、短時間で計画を行っている。予備実験として傷病者 100 人の搬送計画に対し  $k$  の値を 20 まで変化させ、搬送計画を 1 回行うために必要な計算時間とその計画によって搬送する場合の傷病者の合計予測生存率を確認した。予備実験の結果から  $k$  の増加とともに予測生存率も増加したが、 $k = 10$  以降は計画時間の方が大きく増加し、 $k = 16$  以降で 1 分以上、 $k = 20$  以降では 10 分以上必要とした。また、合計予測生存率が  $5 \leq k \leq 15$  では大きな差が無かったため、本実験では最も計画時間が短い 5 を  $k$  として設定した。

**Jotshi's method** : Jotshi らの手法 [29] は、クラスタ (傷病者が密集したある地域) への救急車の派遣と、医療機関への傷病者の搬送という 2 つの処理を行うことで搬送計画を行う。派遣先応急救護所の決定には、傷病者の人数、救急車からクラスタまでの距離、クラスタから全医療機関への距離の合計を用いて選択する。搬送先医療機関の決定には、医療機関の収容能力、医療機関に搬送された傷病者の人数、クラスタから医療機関への距離を用いて選択する。本実験ではクラスタを応急救護所、応急救護所から医療機関までの距離を時間と置き換えそれぞれの処理を行う。Jotshi らのアルゴリズムは応急救護所の選択は行っているが、その中からどの傷病者を搬送するかを決定していない。そのため、本実験では、救急車が応急救護所に到着した時点での、応急救護所内で予測生存率が  $\alpha$  以上で最も低い予測生存率の傷病者を選択する。

**Greedy method** : 救急車が医療機関に到着した時点で、最も近い応急救護所に

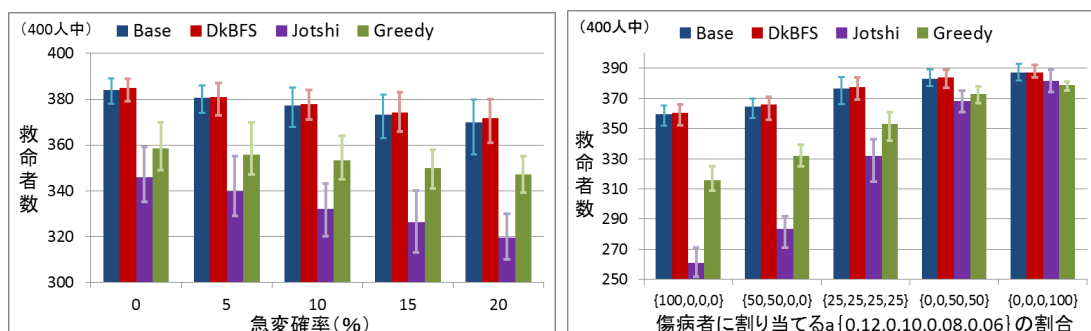


図 8 CASE1:急変確率を変化させた結果 図 9 CASE2:各傷病者に割り当てる  $a$  の割合を変化させた結果

位置する傷病者の中から、予測生存率が  $\alpha$  以上で最も低い予測生存率の傷病者を選択する。最も近い応急救護所に傷病者が存在しない場合は、次に近い応急救護所から傷病者を選択する。搬送先の医療機関は搬送する傷病者が位置する応急救護所から最も近い医療機関を選択する。医療機関の収容能力を超えている場合は、次に近い医療機関に搬送する。

### 2.6.3 実験結果と評価

実験結果において、基本アルゴリズムは Base, 提案手法は DkBFS, Jotshi's method は Jotshi's, Greedy method は Greedy と表記する。各ケースの実験結果を図 8, 9, 10, 11 に示す。

各表の縦軸は救命者数, 横軸は CASE ごとに変化させた値を表しており, 各手法での最大, 最小, 平均を表示している。

図 8 (CASE1) の結果から, 急変確率が増加すると, 全ての手法で救命者数が減少していることが分かる。また, 最大・最小値の幅も長くなる傾向にあることが分かる。急変確率によらず DkBFS は他の手法と比べて最も救命者数が多いことが分かる。

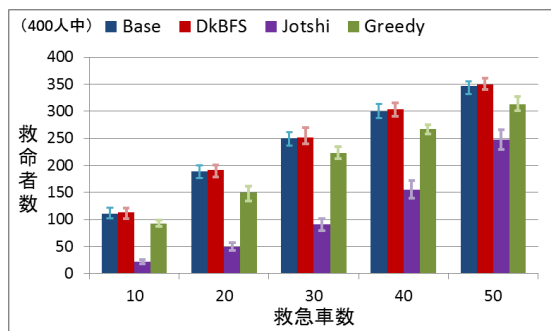
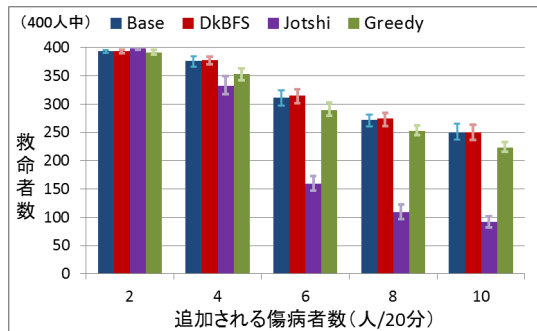


図 10 CASE3 : 追加する傷病者数を変化させた結果 図 11 CASE4 : 救急車の数を変化させた結果

図 9 (CASE2) の結果では、既存手法は搬送限界時刻の長い傷病者が多くなると、救命者数が大きく増加することがわかる。一方、DkBFS でも搬送限界時刻の長い傷病者が多くなるごとに、救命者数が増加するが、他の手法と比べ大きく増加していない。このことから、DkBFS は搬送限界時刻が短い傷病者が多い場合でも効果的な搬送計画が行えているといえる。また、全ての手法で搬送限界時刻の長い傷病者が多くなると、最大・最小値の幅が短くなることが分かる。

図 10 (CASE3) の結果では、追加される傷病者数が最も少ない 10 人 (各応急救護所に 2 人追加) の場合には、DkBFS と他の手法との間に救命者数の差が少ない、これは、救急車の台数が十分足りているからだと考えられる。同様に、追加される傷病者が多くなるごとに、全ての手法で救命者数が減少しているのは、救急車の台数が足りなくなっているからだと考えられる。Jotshi's は特に救命者数が減少している。これは、救急車を派遣する応急救護所を決定する際に傷病者の予測生存確率を考慮していないため、効率的な搬送計画を行えていないからだと考えられる。また、DkBFS は Greedy よりも救命者数が多いことから、単純に搬送時間が短い応急救護所と医療機関の間を往復するよりも、効果的な搬送計画を行えていると言える。

逆に図 11 (CASE4) では、追加される傷病者に対する救急車の増加による救命者数の変化が確認できる。傷病者が各応急救護所に 10 人 (合計 50 人) 追加さ



表 5 統計検定の結果 (CASE1)

	急変確率				
	0%	5%	10%	15%	20%
$p$ 値	0.01183	6.36e-04	7.40e-03	7.89e-07	2.78e-06
$p < 0.05$	true	true	true	true	true

表 6 統計検定の結果 (CASE2)

	各傷病者に割り当てる $a$ の割合 {0.12,0.10,0.08,0.06}				
	{100%,0%, 0%,0%}	{50%,50%, 0%,0%}	{25%,25%, 25%,25%}	{0%,0%, 50%,50%}	{0%,0%, 0%,100%}
$p$ 値	1.55e-06	2.21e-07	2.61e-04	0.003108	0.05092
$p < 0.05$	true	true	true	true	false

れるのに対して、救急車の台数が各医療機関に2台(10台)と圧倒的に足りない場合でも、DkBFSは他の手法より救命者数が多くなっている。また、救急車の台数を多くしていくごとに救命者数が増加しているが、Greedyより増加率が高くなっているため、救急車を効果的に活用できていると言える。

全てのCASEにおいてBASEよりDkBFSの救命者数が多いことがグラフから確認できる。しかし、他の手法より救命者数の差は少なく、各CASEの100回の実験結果の中には、DkBFSよりBASEの救命者数が多い場合が存在した。筆者は予備実験において、急変が起らず、傷病者も追加されないような、傷病者の搬送限界時刻の順位が変化しない環境での評価を行い、BASEよりDkBFSの方が救命者数が必ず多くなることを確認している。これは、DkBFSはBASEの搬送計画も含めた組み合わせの中から、救命者数の多い搬送計画を選択しているからである。しかし、本実験のように、急変が発生し、傷病者が追加される環境では、搬送限界時刻に基づいた搬送順位が時間によって更新されるため、傷病者追加前に計算されたDkBFSの計画(既に途中まで実行済みで後戻り出来ない)が、

表 7 統計検定の結果 (CASE3)

	各救護所に追加する傷病者数 (20分毎)				
	2人	4人	6人	8人	10人
$p$ 値	0.1165	0.002265	7.34e-09	3.92e-04	0.0284
$p < 0.05$	false	true	true	true	true

表 8 統計検定の結果 (CASE4)

	救急車の台数				
	10台	20台	30台	40台	50台
$p$ 値	1.64e-04	5.04e-06	1.04e-05	6.40e-09	2.09e-12
$p < 0.05$	true	true	true	true	true

傷病者の追加により最適でなくなる（傷病者追加前に計算された BASE の計画の方が良くなる）場合がある．そのため，2.6.1 項，および表 4 で設定した 4CASE と各パラメタ 5 種類の合計 20 種類の実験から得られた実験結果に対して，R 言語を用いて統計的な検定を行い有意差を確認した．検定では有意水準を一般的に用いられる 0.05（5%）に設定し検定を行った．2 群間のデータに対応がある場合には，正規分布に従うかによって検定方法が変わるため，まず，シャピロ-ウィルク検定を行い正規性を確認した．その結果，正規性が認められなかったため，ウィルコクソンの符号付き順位和検定を行い有意差を確認した．ウィルコクソンの符号付き順位和検定は，正規分布では無い対応のある 2 群間のデータの差に対して順位付けを行い，その順位和を検定統計量として有意差の検定を行うノンパラメトリックな検定法である．本検定では帰無仮説を「2 群間に差が無い」，対立仮説を「2 群間の差の中央値が 0 より大きい」と仮定し検定を行った．各検定の結果を表 5，6，7，8 に示す．

表 5 から，CASE1 では全ての結果で，有意確率  $p$  が有意水準 0.05 以下であること，つまり「有意差がある」ことが分かる．そのため，急変確率に関わらず，

DkBFS は BASE より効果的な搬送が行えるといえる。表 6 から、CASE2 では  $\{0\%,0\%,0\%,100\%\}$  の場合（全ての傷病者の  $a$  が 0.06 の場合），つまり全ての傷病者の限界搬送時刻が遅い場合に，有意確率  $p$  が 0.05092 で有意水準を上回るため，「有意差が無い」ことが分かる。しかし，それ以外の場合では有意水準以下であることから，DkBFS は限界搬送時刻が早い傷病者が多い場合に BASE より効果的な搬送が行えるといえる。同様に表 7 から，CASE3 では「各救護所に 20 分ごとに 2 人ずつ傷病者を追加する」ような救急車の搬送能力が十分足りる場合に，有意確率  $p$  が 0.1165 で有意水準を上回るが，それ以外の場合には有意水準以下であるため，DkBFS は傷病者数が救急車の搬送能力を上回る場合に BASE より効果的な搬送が行えるといえる。表 8 から，CASE4 では全ての結果で，有意確率が有意水準以下であり「有意差がある」ことが分かる。CASE4 では追加される傷病者数を 10 人に設定しているため，CASE3 のように救急車の搬送能力が足りる場合が無い。そして，救急車を増加させることで  $p$  値が減少する傾向にあるため，DkBFS は救急車の搬送能力が足りない場合に，救急車数の増加による効果が大きいといえる。

## 2.7 まとめ

本章では，多数傷病者事故における救命率の向上を目的に，傷病者の搬送計画問題に着目し，電子トリアージタグを用いた搬送計画システムを提案した。また，搬送計画問題は NP 困難問題であるため，救命率を向上しながら，短時間で準最適解を算出できるヒューリスティックなアルゴリズムを提案した。提案システムでは，電子トリアージタグを用いることでリアルタイムに得られた傷病者の生体情報から予測生存率を算出し，その予測生存率から時間経過に伴う予測生存率の変化を算出できる予測生存率推定関数を作成する。そして DkBFS アルゴリズムによって，限界搬送時間が短い順に  $k$  人ずつ選択し，それぞれが搬送される場合と搬送されない場合の全ての組み合わせの中から，最も救命者数が多くなる組み合わせの搬送計画を探索する。シミュレーション実験を通して，提案手法は既存の搬送計画手法と比べて，より多くの傷病者が生存できる搬送計画を行えていることを確認した。

本研究で対象とした搬送計画問題は，SMT(Satisfiability Modulo Theories) ソルバなどを用いて解決できる可能性がある．これらのソルバでは，計算時間が長くなる可能性があるため，実時間で動作することが望ましいコンテキストウェアシステムにおいては，提案手法の方が適していると考えられる．提案手法の効率を評価するために，ソルバを使った場合と比較することは今後の課題である．

### 3. スマートハウスにおけるコンテキスト遷移を効率的に行うコンテキストウェアシステムの実現法

#### 3.1 はじめに

本章では，スマートハウスにおいて省エネルギーに様々なデバイスをコンテキストウェア制御することを目的に，任意の2つのコンテキスト間を準最適なコストで遷移させるデバイス操作シーケンスを導出するツール *PathSim* を提案する．*PathSim* は，スマートスペースにおけるコンテキストの各構成要素の値域を有限個の範囲に分割することで，コンテキストを有限個の要素からなる空間として扱う．そして，各コンテキストにおいて任意のデバイスの操作イベントを実行した時に遷移するコンテキストをシミュレーションにより求めていくことで，準最適コストで目的のコンテキストに遷移するデバイスの操作シーケンスを A\* アルゴリズムに基づいて導出する．*PathSim* のプロトタイプとして，デバイスの状態，部屋の温度/湿度，ユーザの位置情報などをスマートスペースのコンテキストとして扱い，コンテキスト遷移やデバイスの消費電力量をシミュレーション可能なシステムを実装した．日本の夏と冬を想定した計算機シミュレーションを通して評価実験を行った結果，*PathSim* によって導出された操作シーケンスが，デバイスがそれぞれ目標のコンテキストに調整するように独立して動作するような単純なハイブリッドシステムと比較して，デバイスの合計消費電力量を約 26%削減できることを確認した．また，*PathSim* を採用したコンテキストウェアシステムを実装し現実の環境で動作させた結果，導出したシーケンスにより目的のコンテキストに正しく遷移できることを確認した．

#### 3.2 スマートスペースにおける最小コストデバイス操作問題

本節では，想定するスマートスペースの定義を行う．そして，任意のコンテキストから特定のコンテキストへスマートスペースを遷移させるための最小コストのデバイス操作シーケンスを導出する問題を定義する．

### 3.2.1 スマートスペースのコンテキスト

スマートスペースは1つまたは複数の（有限個の）物理的な位置（または部屋）の集合  $R = \{r_1, \dots, r_k\}$  を持つとする。スマートスペースには、テレビ、エアコン、加湿器、除湿機など、様々なデバイスが設置されている。これらのデバイスの集合を、 $D = \{d_1, \dots, d_m\}$  と表す。各デバイス  $d_i$  は、それぞれ固有の属性  $A^{d_i}(j)$ ,  $j = 1, \dots, n$  を持つ。例えば、テレビの場合、電源状態 (Status), チャンネル (Current-Channel), 音量 (Volume-Level) などの属性を持ち、それぞれ  $A^{TV}(1) = \text{Status}$ ,  $A^{TV}(2) = \text{Current-Channel}$ , and  $A^{TV}(3) = \text{Volume-Level}$  のように表される。また、デバイスの属性  $A^{d_i}(j)$  は、それぞれ、値域  $Dom(A^{d_i}(j))$  を持つ。例えば、テレビの属性は、 $Dom(A^{TV}(1)) = \{ON, OFF\}$ ,  $Dom(A^{TV}(2)) = \{1, \dots, 300\}$ , and  $Dom(A^{TV}(3)) = \{1, \dots, 100\}$  のような値域をそれぞれ持つ。

デバイス  $d_i$  の現在の状態  $c$  は、デバイスの属性値の組  $v_c^{d_i} = \langle v_c^{d_i}(1), \dots, v_c^{d_i}(n) \rangle$  ( $v_c^{d_i}(j) \in Dom(A^{d_i}(j))$ ,  $j = 1, \dots, n$ ) で表される。例えば、テレビの状態は  $v_c^{TV} = \langle v^{TV}(1) = \text{"ON"}, v^{TV}(2) = 10, v^{TV}(3) = 30 \rangle$  のように表すことができる。この例では、テレビの電源状態が“ON”, チャンネルが10ch, 音量が30である。スマートスペース内に設置されている  $m$  個のデバイスの現在の状態は、 $\langle v_c^{d_1}, \dots, v_c^{d_m} \rangle$  で表される。

加えて、スマートスペースに  $l$  人のユーザ  $u_1, \dots, u_l$  がいるものとする。スマートスペース内のユーザの位置は、部屋単位または空間単位で表す。すなわち、各ユーザ  $u_i$  は、 $R$  の要素のいずれかを  $u_i.pos$  に保持する。また、ユーザのIDや行動などの他の属性は  $u_i.attr$  として保持されたとする。ユーザ  $u_i$  がスマートスペース内に居ない場合は、 $u_i.pos$  と  $u_i.attr$  は、 $null$  である。以上のユーザに関する情報は、スマートスペース内に設置されたセンサデバイスやセンシングされた情報を基にユーザの位置や行動を認識するソフトウェアによって得られるものとする。ユーザ  $u_i$  の現在の状態も、 $v_c^{u_i} = \langle u_i.pos, u_i.attr \rangle$  のように属性値の組として表し、 $l$  人のユーザに関する状態を、 $\langle v_c^{u_1}, \dots, v_c^{u_l} \rangle$  と表す。

さらに、スマートスペースは、各位置  $r_1, \dots, r_k$  ごとに環境に関する属性を持つものとする。これらの属性として、温度、湿度、照度、ほこりレベル、騒音レベルなどが考えられるが、本論文では、簡単のため、温度と湿度のみを扱い、位置

$r_i$  の温度を  $r_i.temp$ , 湿度を  $r_i.humidity$  としてそれぞれ表す. これらの属性の値は, デバイスの動作を通して変化し, センサデバイスなどから取得された値が設定される. 位置  $r_i$  の現在の状態を,  $v_c^{r_i} = \langle r_i.temp, r_i.humidity \rangle$  のように表し,  $k$  個の位置に分割されたスマートスペース内の現在の環境を,  $\langle v_c^{r_1}, \dots, v_c^{r_k} \rangle$  と表す.

以上のことを踏まえて,  $m$  個のデバイス,  $l$  人のユーザ,  $k$  ヶ所の位置の状態を保持するスマートスペースの状態 (コンテキスト)  $c$  を,  $c = \langle v_c^{d_1}, \dots, v_c^{d_m}, v_c^{u_1}, \dots, v_c^{u_l}, v_c^{r_1}, \dots, v_c^{r_k} \rangle$  のように表す.

### 3.2.2 コンテキスト遷移

スマートスペースのコンテキストの遷移は, ノードの集合  $V$  とエッジ (有向辺) の集合  $E$  からなる有向グラフ  $G = (V, E)$  で表すことができる. ここで, ノードはスマートスペースのコンテキスト, 有向辺は2つのコンテキスト間の遷移に対応している. また, 各有向辺には, 2つのコンテキスト間の遷移を可能にするイベントとコストが関連付けられる. しかしながら, デバイスの属性, ユーザの位置, 環境の属性を連続値として扱うとコンテキストとして扱う範囲 (以下, コンテキスト空間と表す) が無限になる. また, 使用するデバイスの仕様や性能 (センシング可能な値の範囲および分解能) に合わせて上限と下限を持った離散値として扱うことが可能であるが, 設定可能な, または取得可能な値を全てコンテキストとして扱うと, 同様にコンテキスト空間は膨大になり, コンテキスト空間を探索するためのシステムの負担が大きくなる. この問題に対し, 本論文では, コンテキストの各属性を, 有限個の連続的な範囲 (レンジ) に適切に分割することで対処する. 例えば, 温度のレンジは,  $(-\infty, 0], (0, 10], (10, 20], (20, 30], (30, \infty)$  のような有限個のレンジに区切ることができる. 同様にユーザの位置に関してもスマートスペースの空間全体を適切に分割することで, 有限個の部分空間の集合  $R$  として扱う.

本研究では, **非環境イベント**, **環境デバイスイベント**, **環境遷移イベント** という3種類のイベントを考える. 非環境イベントは, テレビのようなスマートスペース内の環境に影響を与えない (与えたとしても無視できる) デバイスの属性に値を設定するイベントである. テレビ, 音楽プレーヤー, パソコンなどのデバイス

は直接的にスマートスペース内の環境に影響を与えないため、これらのデバイスを非環境デバイスと呼ぶ。例えば、テレビの電源やチャンネルを設定する際には、テレビの電源を”ON”にするイベントやチャンネルを10chに設定するイベントなどが非環境イベントとして生成される。また、ユーザの位置を設定するイベントに関しても、非環境イベントとして扱う。

環境デバイスイベントは、エアコンや加湿器といったスマートスペース内の環境に影響を与えるデバイスの属性を設定するイベントである。これらの環境に影響を与えるデバイスを環境デバイスと呼ぶ。例えば、エアコンACの電源を”ON”にするイベントや”OFF”にするイベントが環境デバイスイベントとして生成される。一方、環境遷移イベントは、温度や湿度といった環境に関する属性が、あるレンジから異なるレンジに遷移することによるコンテキストの遷移を表すイベントである。

デバイスの属性を直接設定できる非環境/環境デバイスイベントを、値  $VALUE$  とデバイス  $d_i$  の属性  $j$  を用いて、 $E\{A^{d_i}(j) \leftarrow VALUE\}$  と表す。また、システムが値を直接設定できない環境遷移イベントを、遷移前のコンテキスト  $c$  と遷移先のコンテキスト  $c'$  を用いて、 $E\{c \rightarrow c'\}$  と表す。

### 3.2.3 問題定義

ここでは、あるコンテキストから任意の別のコンテキストに最小コストで遷移できるイベントシーケンスを導出する問題を定義する。

イベント  $e$  によるコンテキスト  $c$  から  $c'$  への遷移を  $c \xrightarrow{e} c'$  と表す。  $path(c_s, c_t) = \langle e_1, \dots, e_n \rangle$  は、ソースコンテキスト  $c_s$  から対象コンテキスト  $c_t$  にスマートスペースを遷移させるイベントシーケンスを表す。  $c_1, \dots, c_{n+1}$  は、  $c_s$  と  $c_t$  の間で経由する中間コンテキストを表す。ここで、  $c_1 = c_s$ 、  $c_{n+1} = c_t$  として扱う。  $c_s$  から  $c_t$  への遷移可能なイベントシーケンスの条件を以下に記す。

$$\begin{aligned}
& \exists c_2, \dots, \exists c_n, \exists e_1, \dots, \exists e_n \\
& c_s \xrightarrow{e_1} c_2 \wedge c_2 \xrightarrow{e_2} c_3 \wedge \dots \wedge c_n \xrightarrow{e_n} c_t \\
& \wedge e_1 \in E(c_s) \wedge \dots \wedge e_n \in E(c_n)
\end{aligned} \tag{9}$$



ここで、 $E(c)$  は、コンテキスト  $c$  で実行可能なイベントの集合を表す。

本研究では、コンテキスト遷移に必要な消費電力量の最小化を目的にする。そのため、 $e.energy$  をイベント  $e$  の消費電力量とすると、目的関数は以下のように表される。

$$\text{minimize } \sum_{i=1}^n e_i.cost \text{ subject to (9)} \quad (10)$$

各イベントの消費電力量  $e.cost$  は、デバイス毎にあらかじめ与えられた出力対消費電力モデルを利用して計算するものとする。この出力対消費電力モデルの構築方法として、実際にモニタリングした値を用いて構築する方法やカタログスペックを基にして構築する方法などが考えられる。

### 3.3 PathSim : 準最適デバイス操作シーケンス導出ツール

本節では、準最適コストのデバイス操作シーケンス（イベントシーケンス）を導出するツール PathSim について詳細を述べる。

PathSim は A\* アルゴリズムをベースに、ソースコンテキスト  $c_s$  から対象コンテキスト  $c_t$  に遷移させる準最適コストのデバイス操作シーケンスを導出するツールである。A\* アルゴリズムは、ソースコンテキストから隣接するコンテキストを探索することで、経路を導出する手法であるが、対象とする問題では、遷移に必要なイベントおよびコストがあらかじめ与えられないため、隣接するコンテキストに遷移可能かどうかわからない。そのため、A\* アルゴリズムを直接適用できないという問題がある。そこで PathSim では、探索を行う各コンテキストにおいて、シミュレータを用いてイベントを模擬実行することで、遷移する隣接コンテキストと遷移コストを導出し、A\* アルゴリズムをベースに準最適コストのデバイス操作シーケンスを導出する。

3.2 節で説明したように、PathSim では、非環境イベント、環境デバイスイベント、そして、環境遷移イベントという 3 つのイベントを扱う。TV の操作のような非環境イベントは直ぐに反映できるコンテキスト属性を対象とするため、シー

ケンスの順番を考慮する必要は無い。一方、温度や湿度を変化させる環境デバイスイベントにかかる消費電力量は、使用する機器、タイミング、稼働時間に依存する。そのため、イベント選択や実行順序を適切に行うことで、対象コンテキストに到達するために必要なコストを最小化する。

PathSim は、まず非環境イベントに関するシーケンス  $non\_env\_path(c_s, c_r)$  (以下、非環境イベントシーケンスと表す) を導出する。

ここでコンテキスト  $c_r$  は、全ての非環境イベントを反映したあとのコンテキストを表す。その後、環境イベントのみによって構成される  $c_r$  から対象コンテキスト  $c_t$  へのイベントシーケンス  $env\_path(c_r, c_t)$  (以下、環境イベントシーケンスと表す) を導出する。PathSim は、それら2つのシーケンスを統合することで、 $c_s$  から  $c_t$  に準最適コストで遷移可能なパス  $path_{(c_s, c_t)}$  を導出する。以下の節で、非環境イベントシーケンスと環境イベントシーケンスの導出手法の詳細を述べる。

### 3.3.1 非環境イベントシーケンスの導出法

PathSim は、 $c_s$  と  $c_t$  の非環境デバイスの各属性を比較し、その差に基づいて、非環境イベントシーケンス  $non\_env\_path(c_s, c_r)$  を導出する。PathSim は、最初に各非環境デバイス  $d_i \in D$  について、属性値ベクトル  $\mathbf{v}_{c_s}^{d_i}$  と  $\mathbf{v}_{c_t}^{d_i}$  を比較する。 $j$  番目の属性に差がある場合、すなわち、 $\mathbf{v}_{c_s}^{d_i}(j) \neq \mathbf{v}_{c_t}^{d_i}(j)$  である場合には、イベント  $E\{A^{d_i}(j) \leftarrow \mathbf{v}_{c_t}^{d_i}(j)\}$  をイベントシーケンスに追加する。例えば、 $c_s$  のテレビの状態が  $\mathbf{v}_{c_s}^{TV}(Status) = ON$  であり、 $c_t$  のテレビの状態が  $\mathbf{v}_{c_t}^{TV}(Status) = OFF$  である場合には、非環境イベント  $E\{A^{TV}(Status) \leftarrow OFF\}$  がシーケンスに追加される。また、シーケンスの導出においては、テレビのチャンネルを設定する前に、まずテレビの電源を”ON”に設定する、というようなデバイスの操作順序を考慮する。

### 3.3.2 環境イベントシーケンスの導出法

$c_r$  から  $c_t$  への環境イベントシーケンス  $env\_path(c_r, c_t)$  を導出する。

スマートスペースのコンテキストの遷移（コンテキスト，イベント，そしてイベントに対応したコスト）が全て既知である場合には，コンテキストをノード，有向辺をイベントとコストとした有向グラフで表すことができる．グラフの2つのノード間の最小コストパスを見つける問題は最短経路問題と呼ばれ，その効率良い解法としてA\* アルゴリズム [45] が知られている．

PathSim では，環境イベントシーケンス  $env\_path(c_r, c_t)$  を導出するために，A\* アルゴリズムで使用する有向グラフにおける各辺の存在が予め分かっていないということを考慮してA\*アルゴリズムを改変する．また，コスト関数として本研究に適したものを採用する．以上のことを踏まえ，PathSim は，スマートスペース内に設置されているデバイスに関する情報と，与えられた要求仕様を用いて，スマートスペースのコンテキスト遷移（コンテキスト，イベント，そしてイベントに対応したコスト）をオンザフライに導出する．さらに，PathSim は，スマートスペースが対象コンテキストに到達するまで，環境デバイスイベントと環境遷移イベントを適切に選択する．

提案するパス導出アルゴリズムは，A\*アルゴリズムの探索，コスト計算，パスの導出に関連する (a) PathFinder, (b) イベントジェネレータ, (c) コンテキスト遷移シミュレータ, (d) ActCost 関数, (e) EstCost 関数，という5つの関数で構成される．PathFinder は，A\*アルゴリズムの基本的な処理を行うメイン関数であり，残りの4つの関数を用いて，パス，すなわち，環境イベントシーケンス  $env\_path(c_r, c_t)$  を導出する．イベントジェネレータは，任意のノード  $v_c$  から隣接ノードを探索するために，実行可能な環境イベントの集合  $E(c)$  を生成する．コンテキスト遷移シミュレータは，イベント  $e \in E(c)$  を，コンテキスト  $c$  において実行した時のコンテキストの遷移をシミュレートすることで隣接ノードを導出する．ActCost 関数と EstCost 関数は，任意のノード  $v_c$  のコストを計算するために用いられる．ActCost 関数は， $c_r$  から  $c$  までコンテキストを遷移させるための，実コストを計算する関数を表し，EstCost 関数は， $c$  から  $c_t$  までの推定コストを計算する関数を表す．任意のノード  $v_c$  のコストは，実コストと推定コストの合計コストである．

以降では，3.3.3 項～3.3.5 項において，イベントジェネレータ，コンテキスト

遷移シミュレータ, 各コスト関数について詳細を述べ, これらの関数を利用し PathFinder が環境イベントシーケンスを導出するプロセスを 3.3.6 項において確認する.

### 3.3.3 イベントジェネレータ

この関数は, 任意のコンテキスト  $c$  において実行可能なイベントの集合を生成する関数である. スマートスペース内には, 複数のデバイスが設置されていることから, 単に  $c$  において実行可能なイベントを全て生成する方法では, 解空間と計算時間が肥大化する. そのため, 対象コンテキスト  $c_t$  に到達させるために,  $c$  において**有効な** (または, 利用可能な) イベントのみを生成するようにする. 本論文では, 加湿器 (HM), 除湿機 (DH), ヒータ (HT) の電源を ON にするイベントを, それぞれ, 加湿, 除湿, 温度の上昇に**有効なイベント**と定義する. また, エアコン (AC) に関しては, 冷房, 暖房, ドライ (除湿) といった複数のモードを持つと考えられるため, 状態の属性  $A^{AC}(Status)$  の値域を  $Dom(A^{AC}(Status)) = \{OFF, Cold, Hot, Dehumid\}$  と定義し, モードを切り替えるイベントを, それぞれ, 温度の低下, 温度の上昇, 湿度の低下に**有効なイベント**と定義する. 例えば, コンテキスト  $c = ([22^\circ\text{C}, 23^\circ\text{C}], [40\%, 45\%])$  から,  $c_t = ([25^\circ\text{C}, 26^\circ\text{C}], [50\%, 55\%])$  に向かってコンテキストを遷移させる場合には, 温度や湿度を  $c_t$  に向かって変化させることが可能なヒータの電源を入れるイベント  $E\{v^{HT}(Status) \leftarrow "ON"\}$ , エアコンを暖房にするイベント  $E\{v^{AC}(Status) \leftarrow "Hot"\}$ , そして, 加湿器の電源を入れるイベント  $E\{v^{HM}(Status) \leftarrow "ON"\}$  といったイベントが生成される. 一方, 除湿機の電源を ON にするイベントやエアコンを冷房モードにするイベントなどの,  $c_t$  から遠ざかるようにコンテキストが遷移する可能性があるイベントは**有効なイベント**として定義されていないため, 生成されることは無く, **有効でないイベント**が実行済となっている場合には, 電源を OFF にするイベントが生成される.

3.2.2 項で述べたように, 温度/湿度の変化によるコンテキストの遷移は, 環境デバイスイベント (デバイスの操作) では行うことができないため, 遷移前と遷移後のコンテキストを用いて, 環境遷移イベント  $E\{c \rightarrow c'\}$  で表す. しかしなが

ら、シミュレータによってイベントが模擬実行されるまでは、遷移後のコンテキストは分からない。そこで、イベントジェネレータは、直接的にイベントおよび遷移先のコンテキストを表すことのできない暗示的なイベント (Implicit event) を意味する  $E\{\mathcal{S}\}$  というイベントを生成する。

生成された各イベントは、次項で述べるシミュレータによって、コンテキスト  $c$  において、それぞれ、実行した場合に遷移するコンテキスト  $c'$  を導出するために用いられる。

### 3.3.4 コンテキスト遷移シミュレータ

コンテキスト遷移シミュレータは、 $EventSimulator(c, e)$  という関数で実装される。この関数は、現在のコンテキスト  $c$  とイベント  $e$  を入力し、 $c$  において  $e$  を実行することで到達する隣接コンテキスト  $c'$  を出力する。イベント  $e$  が環境デバイスイベントである場合には、コンテキスト  $c'$  は  $e$  に関連した属性以外は  $c$  と同じ属性値を持つ。一方、 $e$  が環境遷移イベントである場合には、関数はスマートスペース内の温度と湿度の変化をシミュレーションし、 $c$  の温度か湿度が隣接するレンジに遷移した後のコンテキスト  $c'$  を導出する。

シミュレーションでは、スマートスペースの容積 (サイズ)、窓や天井を含めた壁の熱伝導率、外気温、デバイスが発する熱や水蒸気、などが考慮される。

温度変化をシミュレーションするために、シミュレータは空間内の温度  $tmp$  が隣接する温度レンジの中央値  $tmp_{nei}$  に到達するまでの時間  $T_{tmp \rightarrow tmp_{nei}}$  を、以下の式を用いて計算する。

$$T_{tmp \rightarrow tmp_{nei}} = \frac{Q_{tmp \rightarrow tmp_{nei}}}{Q_{dev} + Q_{ext}} \quad (11)$$

where

$$Q_{tmp \rightarrow tmp_{nei}} = C \cdot (tmp_{nei} - tmp)$$

$$Q_{ext} = H \cdot Area \cdot (tmp_{ext} - tmp)$$

ここで、 $Q_{tmp \rightarrow tmp_{nei}}$  は、空間内の温度を  $tmp$  から  $tmp_{nei}$  へ変化させるために必要な熱量を表し、温度差と空間の熱容量  $C$  を乗算することで計算される。 $Q_{dev}$

は、温度に関連した全ての動作中のデバイスから時間あたりに得られる熱量を表している。  $Q_{ext}$  は外気温  $tmp_{ext}$  から壁を通して時間あたりに得られる熱量を表しており、熱伝導率、全ての壁の面積  $Area$ 、そしてスマートスペース内の温度と外気温との差  $tmp_{ext} - tmp$  によって計算される。

湿度の変化をシミュレーションするために、空間内の現在の湿度  $hmd$  が隣接する湿度レンジの中央値  $hmd_{nei}$  に到達するまでの時間  $T_{hmd \rightarrow hmd_{nei}}$  を計算する。この計算では、空間の体積  $Vol$ 、湿度に関する全ての動作中のデバイスが時間あたりに発する（または、除去する）水蒸気量  $M_{dev}$  および1秒ごとに流入出する水蒸気量  $M_{ext}$  を考慮し、以下の式を使用して時間を計算する。

$$T_{hmd \rightarrow hmd_{nei}} = \frac{Vol \cdot (hmd_{nei} - hmd)}{M_{dev} + M_{ext}} \quad (12)$$

where

$$M_{ext} = Vol_{ven} \cdot (hmd_{ext} - hmd)$$

ここで、  $M_{ext}$  は、1秒当たりの換気量  $Vol_{ven}$  および外気の湿度  $hmd_{ext}$  と空間内の湿度  $hmd$  の差によって計算される。この際、相対湿度は空間内の水蒸気量が同じであっても、温度が異なることで変化するため、上記の式では、Tetensの式 [46] を利用して、相対湿度を絶対湿度に変換して計算を行う。

$T_{tmp \rightarrow tmp_{nei}}$  と  $T_{hmd \rightarrow hmd_{nei}}$  を計算した後、シミュレータは両方の値を比較する。この時、  $T_{tmp \rightarrow tmp_{nei}} < T_{hmd \rightarrow hmd_{nei}}$  である場合には、温度が湿度より先に隣接するレンジに到達するため、シミュレータは以下の式を用いて現在の湿度  $hmd$  の値を更新する。

$$hmd \leftarrow hmd + \frac{((M_{dev} + M_{ext}) \times T_{tmp \rightarrow tmp_{nei}})}{Vol}$$

一方、  $T_{tmp \rightarrow tmp_{nei}} > T_{hmd \rightarrow hmd_{nei}}$  である場合には、シミュレータは以下の式を使用して現在の温度  $tmp$  を更新する。

$$tmp \leftarrow tmp + \frac{((Q_{dev} + Q_{ext}) \times T_{hmd \rightarrow hmd_{nei}})}{C}$$

更新された  $tmp$  と  $hmd$  の値は、到達したコンテキスト  $c'$  に、 $c'.tmp$  と  $c'.hmd$  としてそれぞれ保存される。

その後、シミュレータは、以下の式を用いて、遷移時間  $T$  において動作中の全デバイスの消費電力量の総和を求め、コストとして  $e.cost$  に保持する。

$$e.cost = \sum_{d \in D} \int_0^T (power(d, o) \cdot t) dt \quad o \in d.O \quad (13)$$

ここで  $T$  は、 $T_{tmp \rightarrow tmp_{nei}}$  と  $T_{hmd \rightarrow hmd_{nei}}$  のうち小さい方の時間であり、 $power(d, o)$  は、出力  $o \in d$  で動作中のデバイス  $d$  の消費電力を表す。デバイス  $d$  を出力  $o$  で動作させた時の消費電力  $power(d, o)$  は、デバイス毎にあらかじめ与えられた出力対消費電力のモデルを用いて計算する。

### 3.3.5 コスト関数

3.3.6 項で説明する PathFinder では、各シーケンスのコストを評価するために、 $ActCost(c, c')$  と  $EstCost(c, c_t)$  という2つのコスト関数を使用する。 $ActCost(c, c')$  はコンテキスト  $c$  から  $c'$  に遷移する際の、イベントの合計コストを計算する関数である。前節で説明したように、コンテキスト遷移シミュレータを用いて、コンテキストの遷移に掛かるコストを計算できる。そのため、 $c$  から  $c'$  までのパスに含まれる全てのイベントのコストを足し合わせることでパスのコストを計算できる。一方、 $EstCost(c', c_t)$  は、探索が行われていないコンテキスト  $c'$  から対象コンテキスト  $c_t$  へのコストを推定する関数である。 $EstCost()$  は、 $c$  から  $c_t$  へ遷移する間の操作シーケンスを考慮せず、 $c$  と  $c_t$  間の遷移に必要な熱量および水蒸気量に対して、 $c$  から有効なデバイスを動作させた場合のデバイスが発する熱量と水蒸気量を用いて、それぞれの遷移に必要な消費電力量を計算し合計することでコストの推定を行う。この際、温度および湿度にそれぞれ有効なデバイスに対して、使用する場合と使用しない場合の組み合わせの中から、遷移に必要な消費電力量が、最小となる組合せの消費電力量を推定コストとして採用する。

### 3.3.6 PathFinder : 環境イベントシーケンス導出アルゴリズム

PathSim のメイン関数である PathFinder は、コンテキスト  $c_r$  から  $c_t$  への準最適なコストパス（イベントシーケンス）を A\* アルゴリズムに基づいて導出する。

PathFinder によって導出される各ノードは、それぞれ、コンテキスト  $v.c$ 、親ノード  $v.p$ 、 $v.p$  から  $v.c$  に遷移するために実行されたイベント  $v.e$ 、 $v.p$  から  $v.c$  に遷移するために掛かるコスト  $v.cost$ 、そして  $v_r$  から  $v$  までのコストと  $v$  から  $v_t$  への推定コストから計算される合計コスト  $v.cost_{total}$  を保持する。  $v_{cur}$  を現在のノード、 $v_{nei}$  を  $v_{cur}$  からイベント  $v_{nei}.e$  によって到達するノード、そして、 $v_{old}$  を  $v_{nei}$  より先に導出された同じコンテキストを持つノードと定義する。また、 $v_r$  と  $v_t$  を、それぞれソースコンテキスト  $c_r$  を持つソースノードと対象コンテキストを持つ  $c_t$  を持つ対象ノードとして定義する。

PathFinder のアルゴリズムを Algorithm1 に示す。PathFinder のアルゴリズムは、 $c_r$  と  $c_t$  が入力されたときに、コンテキスト  $c_r$  から  $c_t$  に準最適コストで遷移可能なイベントシーケンス  $env\_path(c_r, c_t)$  を導出し、出力するアルゴリズムである。

以下では、アルゴリズムの核となる部分に関して簡潔に説明する。PathFinder は、まず、現在のノードが保持するコンテキスト  $v_{cur}.c$  において実行可能な環境イベントの集合  $E(v_{cur}.c)$  (以下、 $E()$ ) を生成する (line 8)。集合  $E()$  には、 $\{E\{\mathcal{S}\}\}$  で表される環境遷移イベントも含まれている。このイベントは、現在のコンテキスト  $v_{cur}.c$  から、温度と（または）湿度のレンジが異なる隣接コンテキスト  $v_{nei}.c$  にスマートスペースが遷移することを表す。各環境遷移イベントに対して、PathFinder はイベントを実行した後の隣接コンテキスト  $v_{nei}$  を導出する (line 10)。その後、 $v_s.c$  から  $v_{nei}.c$  に到達するためのコストをコスト関数  $ActCost(v_s, c_{nei})$  を用いて計算し、 $v_{nei}.c$  から  $v_t.c$  に到達するための推定コストをコスト関数  $EstCost(v_{nei}, v_t)$  を用いて計算する (line 11)。PathFinder は、計算された 2 つのコストを合計することで、 $v_s.c$  から  $v_t.c$  に  $v_{nei}.c$  を経由して到達するための合計コストを決定する。PathFinder は、最小合計コストを持つノードを  $v_{cur}$  として再帰的に選択し、選択された  $v_{cur}$  ごとに隣接ノードを導出する。この処理を  $v_t$  が  $v_{cur}$  として選択されるまで行う (lines 12–32)。



アルゴリズムの動作を、図12と13の例を用いて説明する。図12では、未探索ノードの集合 *OPEN* から、探索を行うノードとして  $v_s$  が選択され、イベントジェネレータにより5つのイベントを含む  $E()$  が生成される。その後、PathFinderは、 $E()$  の各イベントに対して、 $EventSimulator()$  関数を用いて、隣接コンテキストノード  $v_{nei}$  を決定し (line 10) ,  $v_{nei}$  を経由した場合の  $v_s$  から  $v_t$  へ遷移するために必要な合計コストを計算する (line 11) 。図12では、 $v_s$  から到達可能な  $v_1$  から  $v_5$  までの5つの隣接ノードが合計コストと共に導出される。この例では、 $v_1$  から  $v_5$  までの各ノードに関して、 $v_s$  から各ノードに到達するパスが1つしかないため、line 12 で表されている条件と合致する。そのため、 $v_1$  から  $v_5$  までの5つの隣接ノードは *OPEN* リストに追加される (line 13) 。

2回目のループ (lines 5–33) では、*OPEN* 内で1.01kWhという最も低いコストを持つ  $v_2$  が  $v_{cur}$  として選択される (line 4) 。そして、 $E$  の生成、隣接ノードの導出などの上記の処理が繰り返される。

しかし、隣接ノード  $v_{nei}$  が既に未探索ノードの集合 *OPEN* か探索済ノードの集合 *CLOSE* に含まれている場合 (lines 14, 20) ,  $v_{nei}$  には以前の探索によって異なるパスと合計コストが既に保持されていると言える。簡単に説明するため、以前の探索で導出されたノードを  $v_{old}$  と表す。PathFinderは、 $v_{nei}$  を通る現在のパスと  $v_{old}$  を通る以前のパスのうち、最小合計コストを持つノードのみを保持する。この時、 $v_{nei}$  を保持する場合には、 $v_{old}$  を *OPEN* か *CLOSE* から除外し、 $v_{nei}$  を *OPEN* に追加する (lines 16–19, 22–25) 。

図13は  $v_{nei}$  が *OPEN* に既に含まれている場合を表している。このケースで  $v_3$  は、 $v_2$  からコスト0.2kWhで到達可能であるが、 $v_3$  は既に *OPEN* に含まれている。ここで、 $v_2$  から到達する  $v_3$  を  $v_{nei}$  で表し、 $v_s$  から到達する  $v_3$  を  $v_{old}$  と表す (lines 14, 15) 。PathFinderは、2つのノードの合計コストを比較し、最も小さい合計コストを持つ方を *OPEN* に保持する (lines 16–19) 。図13の場合、 $v_s \rightarrow v_3(v_{old}) \rightarrow \dots \rightarrow v_t$  の合計コストは  $0.015+1=1.015\text{kWh}$  ,  $v_s \rightarrow v_2 \rightarrow v_3(v_{nei}) \rightarrow \dots \rightarrow v_t$  の合計コストは  $0.01 + 0.2 + 1 = 1.21\text{kWh}$  である。そのため、 $v_{old}$  が *OPEN* に保持され続け、 $v_{nei}$  は *OPEN* に追加されず破棄される。Algorithm1 の lines 20–26 で表されてるように、 $v_{nei}.c$  が既に *CLOSE* に含まれている場合にも同様の処理が

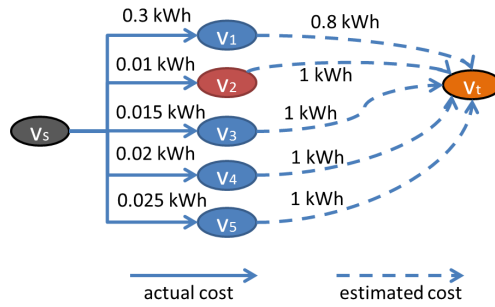


図 12  $v_{nei}$  の導出例と  $v_{cur}$  の選択例

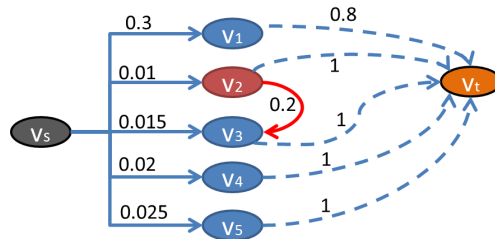


図 13  $v_{nei}$  が既に *OPEN* または *CLOSE* に含まれているケース

適用される。ただし、 $v_{nei}.c$  を保持することが決定された場合には、*CLOSE* に含まれている  $v_{old}$  を破棄し、 $v_{nei}.c$  を *OPEN* に追加する。

上記の探索は、対象ノードに到達するまで繰り返される。すなわち、 $v_{cur}$  と  $v_t$  が同じ場合に、アルゴリズムは lines 5-33 のループを終了する。イベントシーケンス  $env\_path(c_r, c_t)$  は、対象ノード  $v_t (=v_{cur})$  から  $v_s$  に到達するまで親ノード  $v_{cur}.p$  を再帰的に辿りながら、各ノードのイベント  $v_{cur}.e$  をパスの先頭に追加していくことで導出される。

探索するノードが *OPEN* リストに存在しない場合には、対象ノードへのパスが存在しないと判定し、探索を終了する。この場合には、イベントシーケンスは出力されず、探索が失敗したことが通知される。

---

**Algorithm 1** PathFinder

---

**Input:**  $c_r, c_t$   
**Output:**  $env\_path(c_r, c_t)$

- 1:  $v_s.c \leftarrow c_r, v_s.p \leftarrow null, v_s.e \leftarrow null, v_s.cost \leftarrow 0, v_s.cost_{total} \leftarrow 0$
- 2:  $v_t.c \leftarrow c_t, v_t.p \leftarrow null, v_t.e \leftarrow null, v_t.cost \leftarrow 0, v_t.cost_{total} \leftarrow 0$
- 3:  $OPEN \leftarrow \{v_s\}, CLOSE \leftarrow \emptyset$
- 4:  $v_{cur} \leftarrow \operatorname{argmin}_{v \in OPEN} v.cost_{total}, v_{nei} \leftarrow null, v_{old} \leftarrow null$
- 5: **while**  $v_{cur}.c \neq v_t.c$  **do**
- 6:    $CLOSE \leftarrow CLOSE \cup \{v_{cur}\}$
- 7:    $OPEN \leftarrow OPEN \setminus \{v_{cur}\}$
- 8:    $E \leftarrow \text{EventGenerator}(v_{cur}.c, v_t.c)$
- 9:   **for all**  $e \in E$  **do**
- 10:      $v_{nei} \leftarrow \text{EventSimulator}(v_{cur}, e)$
- 11:      $v_{nei}.cost_{total} \leftarrow \text{ActCost}(v_s, v_{nei}) + \text{EstCost}(v_{nei}, v_t)$
- 12:     **if**  $\neg \exists v \in OPEN \cup CLOSE (v.c = v_{nei}.c)$  **then**
- 13:        $OPEN \leftarrow OPEN \cup \{v_{nei}\}$
- 14:     **else if**  $\exists v \in OPEN (v.c = v_{nei}.c)$  **then**
- 15:        $v_{old} \leftarrow v$
- 16:       **if**  $v_{nei}.cost_{total} < v_{old}.cost_{total}$  **then**
- 17:          $OPEN \leftarrow OPEN \setminus \{v_{old}\}$
- 18:          $OPEN \leftarrow OPEN \cup \{v_{nei}\}$
- 19:       **end if**
- 20:     **else if**  $\exists v \in CLOSE (v.c = v_{nei}.c)$  **then**
- 21:        $v_{old} \leftarrow v$
- 22:       **if**  $v_{nei}.cost_{total} < v_{old}.cost_{total}$  **then**
- 23:          $CLOSE \leftarrow CLOSE \setminus \{v_{old}\}$
- 24:          $OPEN \leftarrow OPEN \cup \{v_{nei}\}$
- 25:       **end if**
- 26:     **end if**
- 27:   **end for**
- 28:   **if**  $OPEN = \emptyset$  **then**
- 29:     **search failed**
- 30:   **else**
- 31:      $v_{cur} \leftarrow \operatorname{argmin}_{v \in OPEN} v.cost_{total}$
- 32:   **end if**
- 33: **end while**

```

34:  $env\_path(c_r, c_t) \leftarrow \epsilon$ , where  $\epsilon$  is the empty string
35:  $env\_path(c_r, c_t).cost \leftarrow v_{cur}.cost_{total}$ 
36: while  $v_{cur} \neq v_s$  do
37:   {Traversing nodes from  $v_{cur}$  to  $v_s$  to derive the path  $env\_path(c_r, c_t)$ }
38:    $env\_path(c_r, c_t) \leftarrow \langle v_{cur}.e \rangle.env\_path(c_r, c_t)$ 
39:    $v_{cur} \leftarrow v_{cur}.p$ 
40: end while
41: return  $env\_path(c_r, c_t)$ 

```

---

### 3.3.7 非環境/環境イベントシーケンスの結合

$non\_env\_path(c_s, c_r)$  と  $env\_path(c_r, c_t)$  という 2 つのシーケンスを導出した後に、それらを結合することでデバイス操作シーケンス  $path(c_s, c_t)$  を獲得できる。この時、非環境イベントは、ユーザが直ぐに提供されることを望むサービスだと考えられる。例えば、テレビの電源を”ON”にするイベントは、スマートスペース内の環境を変化させる前に実行されるべきイベントである。従って、 $env\_path(c_r, c_t)$  は  $non\_env\_path(c_s, c_r)$  の後に実行されるように結合される。

## 3.4 シミュレーションによる評価

本節では、実際のスマートホームを想定したシミュレーションによって、PathSim で導出したデバイス操作シーケンスの消費電力量を評価する。

### 3.4.1 対象とするスマートホームおよび使用するデバイス

対象とするスマートホームとして、図 14 で表したスマートホームのフロアプランのうち、簡単化のため、図内の赤色の線で囲まれた 1 つの部屋のみを対象とする。対象空間を囲む各壁の熱伝導率として、木製の壁と天井の熱伝導率を  $0.15W/(m \cdot K)$ 、ガラス窓の熱伝導率を  $0.2W/(m \cdot K)$  とした。また、流入出する水蒸気量を計算するために、予備実験として図 14 の対象空間において湿度の変化量を実際に計測し、1 秒当たりの換気量  $Vol_{ven} = 0.0762$  を導出した。外気の温

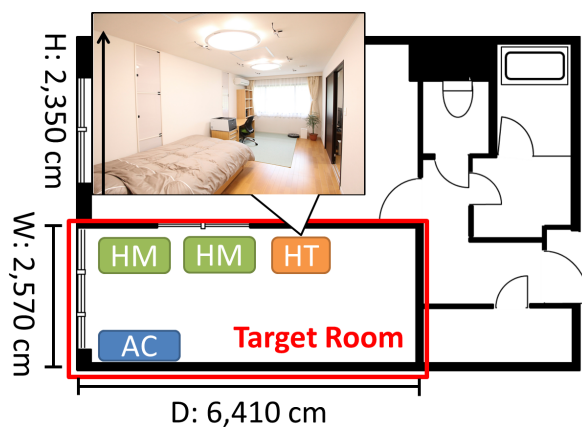


図 14 対象とするスマートホーム

度と湿度は、3.4.2項で述べるソースコンテキストの温度レンジと湿度レンジの中央値とする。

使用するデバイスとして、1台のエアコン、1台のヒーター、1台の除湿器、2台の加湿器という5つのデバイスを使用する。エアコンは東芝製 RAS-281EDR、ヒーターは De'Longhi 製 TDD0915W、除湿機は SHARP 製 DW-S180 を想定し、加湿器は ALCOLLE 製 ASH-600 と Panasonic 製 F-VXF45 という性能の異なる2つの加湿器を想定した。エアコンは、暖房、冷房、ドライ（除湿）という3つのモードを備えている。各デバイスの出力対消費電力モデルとして、表9に示したカタログスペックに基づいて、出力を最大出力のみ、消費電力を最大出力時の消費電力とするモデルを構築した。また各デバイスの能力についてもカタログスペックの最大出力時の能力を用いる。さらに、コンテキストウェアシステム自体の動作コストとして、本ケーススタディでは、ノートパソコンでコンテキストウェアシステムが動作するものと仮定し、毎秒 15W の消費電力が消費されるとともに同値の熱量が放出されるものとする。

実際の環境においては、例えば、加湿器の場合、加熱式では湿度と共に温度も上昇し、気化式では湿度の上昇とは反対に温度は低下するように、副次的な効果を持つデバイスも存在する。これらの副次的な効果に関しては、カタログスペック

表 9 実験で使用するデバイスの設定

	消費電力	W:冷/暖房能力, L:加/除湿量
エアコン RAS-281EDR	COLD:990W, HOT:1225W, DH:155W	COLD:3.4kW, HOT:4.7kW, DH:1.5L/h
ヒーター TDD0915W	1500W	1500W
除湿器 DW-S180	420W	750mL/h
加湿器 ASH-600	35W	500 mL/h
加湿器 F-VXF45	35W	450 mL/h
Laptop	15W	15W

クには記述されておらず、実測値に基づいて求める必要がある。簡単のため、本論文においては、副次的な効果を考慮しないものとする。

### 3.4.2 要求仕様およびソース/対象コンテキスト

本評価では、夏と冬においてスマートホームの環境を調整するコンテキストウェアシステムを想定し評価を行う。

対象コンテキストは、室内環境を専門とする温度計・湿度計を製造するメーカーが公開している「快適な温度・湿度の目安」[47]を参考に生成される。夏の快適な温度[25℃, 28℃)を1℃ごとに区切ることで3つのレンジ([25℃, 26℃), [26℃, 27℃), [27℃, 28℃))に分割し、夏の快適な湿度[55%, 65%)に関しても2.5%ごとに区切ることで4つのレンジ([55%, 57.25%), [57.25%, 60%), [60%, 62.25%), [62.25%, 65%))に分割する。また、冬の快適な温度[18℃, 22℃)と快適な湿度[47.5%, 57.5%)も同様に3つの温度レンジと4つの湿度レンジに分割する。そして、温度と湿度の

ペアを用いて季節毎に 12 個の対象コンテキストを生成する。

一方、ソースコンテキストは、冬に関しては、図 14 の対象スペースで実際にモニタリングされた室温を参考に  $[15^{\circ}\text{C}, 18^{\circ}\text{C})$  を、夏に関しては、奈良県の気象情報を参考に  $[28^{\circ}\text{C}, 31^{\circ}\text{C})$  をそれぞれ 3 つの温度レンジに分割し、湿度に関しては、モニタリングした値と気象情報を参考に、夏に関しては  $[60\%, 70\%)$  を、冬に関しては  $[40\%, 50\%)$  をそれぞれ 4 つの湿度レンジに分割し、対象コンテキストと同様に温度と湿度のペアを用いて季節毎に 12 個のソースコンテキストを生成する。

また、生成されたソースコンテキストと対象コンテキストにおいて、全てのデバイスの電源状態は OFF に設定されているものとする。

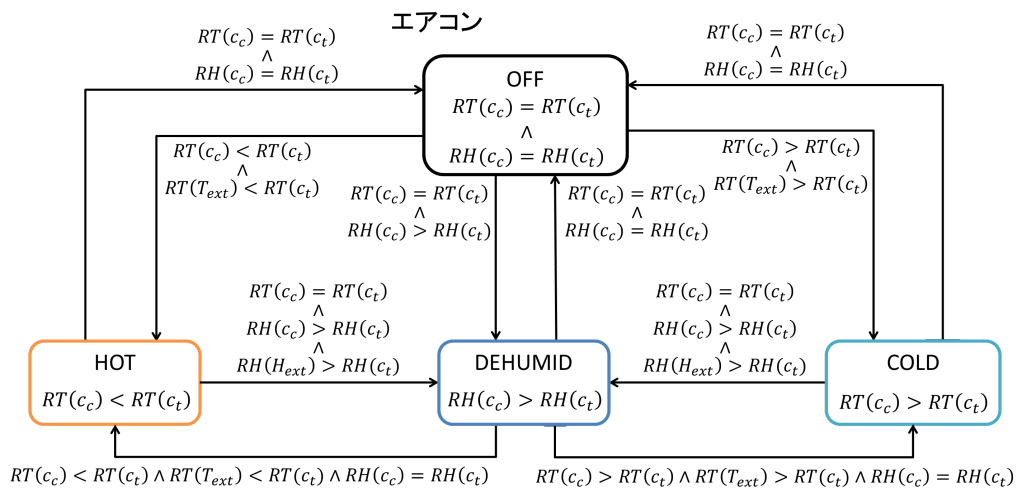
### 3.4.3 実験方法

本実験では、季節毎に生成した 12 個のソースコンテキストと 12 個の対象コンテキストに対して、 $12 \times 12$  の 144 通りのデバイス操作シーケンスを導出し、各シーケンスの消費電力量や遷移時間を評価する。

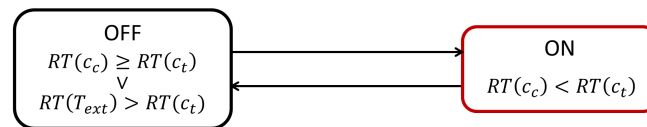
本実験における比較手法として、図 15 に表した単純なハイブリッドダイナミカルシステム (Hybrid Dynamical System, HDS) のオートマトンに基づき各デバイスが独立して動作する場合の操作シーケンスを導出し PathSim と比較を行う。ハイブリッドダイナミカルシステム [48] は、温度や湿度などの連続値とデバイスの状態などの離散値を扱う動的システムであり、PathSim で導出した操作シーケンスに基づいて動作するコンテキストウェアシステムも HDS の 1 つである。図 15 で表したハイブリッドオートマトンは、それぞれ、デバイスの状態をモード (コンテキスト)、各エッジの添え字、および各モードの下部に記述された条件式は遷移条件 (遷移を許可する条件) と不変条件 (モードを維持するために満たすべき条件) を表す。また、ハイブリッドオートマトンは、各モードにおいて、変数として温度、湿度、消費電力などの連続値、およびこれらの変化量を保持するが、図 15 では簡略化と省略を行い遷移に関する条件のみを表記している。各モードの不変条件は、モードを維持するために必要な条件が表されており、条件を満たさなくなった時点で必ず遷移が行われることを表している。例えば、ヒータの

OFF モードは，現在の温度レンジ  $RT(c_c)$  が対象コンテキスト  $c_t$  の温度レンジが同じ場合か表す温度が高い場合，または外気温が対象コンテキストの温度レンジが表す温度が高い場合にのみ維持される．これは，夏場に温度を上げる場合に，一般的にヒータを使わず外気からの影響のみで上昇させるように，外気によって対象コンテキストに遷移可能な場合には OFF モードを維持し，現在の温度と外気温が低い場合にヒータが ON モードに遷移することを表している．エアコンに関しても，外気のみで遷移できる場合には OFF モードを維持するが，簡単のため図 15 では省略し遷移条件のみを記述した．

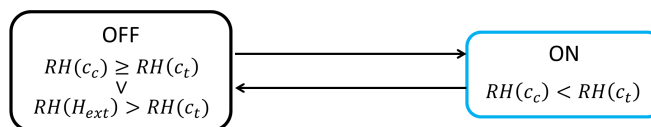




**ヒーター**



**加湿器**



**除湿器**

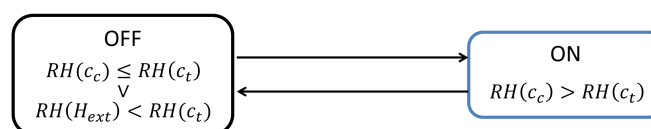


図 15 比較手法における各デバイスのオートマトン

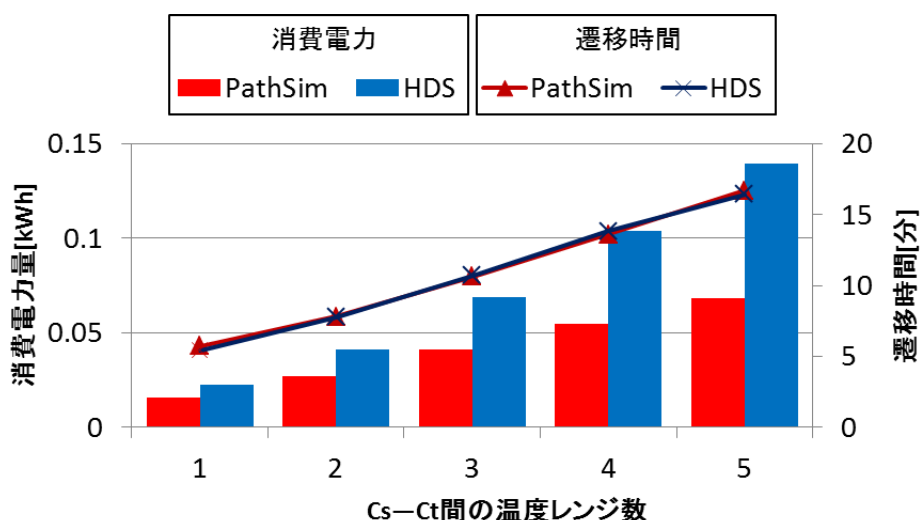


図 16 冬想定時のデバイス操作シーケンスの消費電力量および遷移時間

### 3.4.4 実験結果

実験結果を図 16 および図 17 に示す。図 16 および図 17 は、それぞれ、季節毎に導出したデバイス操作シーケンスの消費電力量と遷移時間を表している。消費電力量を表す棒グラフと遷移時間を表す折れ線グラフは、144 通りのデバイス操作シーケンスを、ソースコンテキストと対象コンテキスト間の温度レンジの数（以下、遷移レンジ数）で分割し、平均したものを表している。

図 16 と図 17 の棒グラフから、PathSim で導出したデバイス操作シーケンスは単純な HDS のオートマトンに従って動作した場合と比べて少ない消費電力量でコンテキストを遷移できることが確認できる。遷移時間に関しては、冬を想定した実験結果において、HDS と同様の遷移時間で、且つ少ない消費電力量で遷移できるデバイス操作シーケンスを導出していることが分かる。一方、夏を想定した実験結果では、冬を想定した場合と比べ、消費電力量の差も少なく、遷移時間も HDS より遅くなっていることが分かる。これは、冬と比べ、消費電力の削減につながる選択肢が少ないためだと考えられる。本実験では実際に使用できるデバイスを想定してシミュレーションを行っているが、例えば、エアコンを 2 台使える

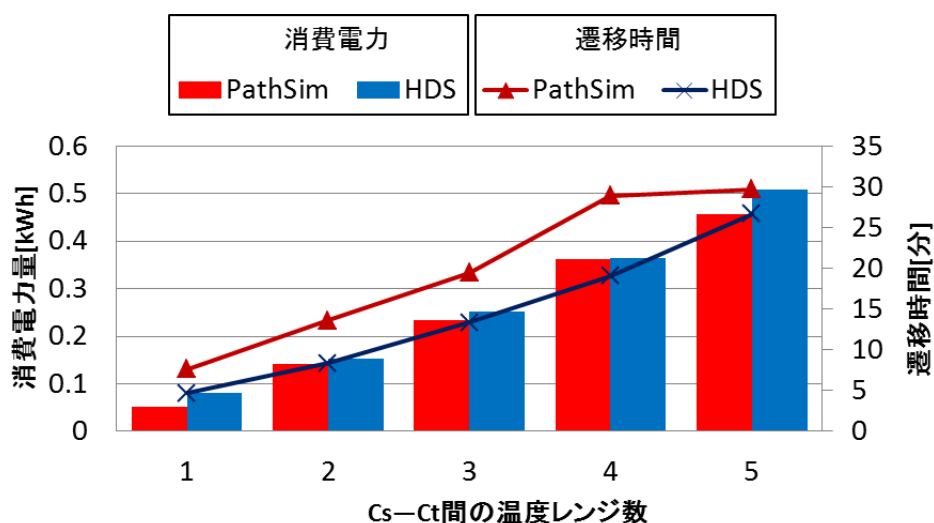


図 17 夏想定時のデバイス操作シーケンスの消費電力量および遷移時間

環境を考えるなど，PathSim が取れる選択肢を増やすことにより，より少ない消費電力量のデバイス操作シーケンスを導出できる可能性がある。

次に比較手法に対する遷移レンジ数ごとの消費電力量の削減率を表 10 に示す。表 10 の冬想定時を確認することで，提案手法は，特に冬環境で多く消費電力量を削減できていることが分かる。また，遷移レンジ数が大きくなると，消費電力量の削減率が上昇している。これは，PathSim がソースコンテキストと対象コンテキストの遷移の間に適切なデバイス制御を行っているからだと考えられる。また，夏と冬の削減率を平均すると PathSim は単純な HDS と比べて平均で約 26% の消費電力量を削減できることが分かる。夏想定時に関しては，最も遷移レンジ数が小さい場合を除いて冬想定時より削減率が低いが，前述した通り，温度や湿度を下げるデバイスの選択肢を追加することで，より削減率を高めることが可能だと考えられる。

上記のように，提案手法は省エネルギーにコンテキストを遷移できる手法であるが，図 17 のように，時間が多く掛かる可能性も存在する。本研究では，消費電力量の最小化を目的関数としているが，遷移するまでの時間が長くなることに対して，不便を感じるユーザもいると考えられる。この問題は，提案手法に時間

表 10 HDS に対する消費電力量の削減率

	$c_s - c_t$ 間の温度レンジの数					
	1	2	3	4	5	平均
冬想定時	-28%	-34%	-40%	-47%	-51%	-40%
夏想定時	-36%	-6%	-7%	-0.47%	-10%	-12.1%

制約を追加し、時間制約を満たすデバイス操作シーケンスを導出することにより、解決できると考えている。しかしながら、長い時間がかかる場合でも、例えば、ユーザが就寝するために寝室に入る時刻や起床する時刻に合わせて環境を調整するような場合には、遷移時間の長さはあまり問題にならないため、消費電力量のみの最小化でも有効だと考えられる。

### 3.5 実環境実験による評価

PathSim で導出したデバイス操作シーケンスが実環境においても有効かどうか確認するために、実際のスマートスペースで動作するコンテキストウェアシステムを開発した。本章では、開発したコンテキストウェアシステムの概要と、デバイス操作シーケンスの有効性について述べる。

#### 3.5.1 開発したコンテキストウェアシステム

開発したコンテキストウェアシステムの要件を以下に示す。

- 対象スペースのコンテキストを監視する
- 現在のコンテキストから与えられたコンテキストへのデバイス操作シーケンスを PathSim で導出する
- デバイス操作シーケンスに基づきデバイスを操作する

- デバイス操作シーケンスとは異なるコンテキストに遷移した場合にシーケンスを再導出する

本実験では、3.4節で用いたスマートスペースを対象とし、エアコン、ヒーター、加湿器に関しても同様のデバイスを使用する。各デバイスにはスマートメータが装着されており、時間ごとの消費電力量が把握可能である。また、部屋の内部と外部にはそれぞれ、温度・湿度センサが設置されており、実時間で温度と湿度を把握可能である。

本システムは各種センサを用いて現在のコンテキストを実時間で監視し、与えられたコンテキストへデバイスを操作して遷移させる。そのために、システムは目標となるコンテキストが与えられた際に、現在のコンテキストから目標となるコンテキストまでのデバイス操作シーケンスを PathSim で導出する。そして、導出されたデバイス操作シーケンスに基づいてデバイスを操作する。

予備実験として開発したシステムを動作させたところ、コンテキストが遷移した直後に再び直前のコンテキストに遷移するなどデバイス操作シーケンス通りにコンテキストが遷移しない場合が確認できた。これは、センサの精度やリアルタイムに変化する外気の影響、そして、PathSim でシミュレートした物理量の変化量と実環境における物理量の変化量との差が原因であると考えられる。そこで、遷移した直後に直前のコンテキストに遷移する問題には、システムが保持している現在のコンテキストと新たに観測されたコンテキストが異なる場合に、新たに観測されたコンテキストがそのコンテキストを一定時間維持し続けた場合にのみ、コンテキストの遷移が起こったと判断するように対処した。また、予期しないコンテキストに遷移した場合には、遷移したコンテキストから目標のコンテキストまで、再度 PathSim によりデバイス操作シーケンスを導出することで対処する。

### 3.5.2 有効性の評価

評価指標として、PathSim で導出した消費電力量と実際に掛かった消費電力量の差およびデバイス操作シーケンスの再導出回数を計測した。

本実験では、システムを起動させた時間のコンテキストをソースコンテキストとして、温度と湿度をそれぞれ1ずつ高いレンジに遷移させるように対象コンテ

表 11 実環境実験で生成されたコンテキスト（デバイスの状態は全て OFF に設定したため省略）

	ソースコンテキスト $C_s$	対象コンテキスト $C_t$
遷移 1	room.temp = [18 °C, 20 °C), room.humid = [45%,50%)	room.temp = [20 °C, 22 °C), room.humid = [50%, 55%)
遷移 2	room.temp = [20 °C, 22 °C), room.humid = [45%,50%)	room.temp = [22 °C, 24 °C), room.humid = [50%, 55%)

表 12 実環境実験の結果

	遷移 1	遷移 2
シーケンスの予測消費電力量	14.978 Wh	16.422 Wh
実際の消費電力量	11.28 Wh	27.472 Wh
消費電力量の差	3.68 Wh	-11.05 Wh
シーケンス再導出回数	1 回	0 回

キストを生成した。表 11 は、本実験で生成されたソースコンテキストと対象コンテキストの組を表している。本実験では簡単化のため、ユーザの位置情報に関しては考慮せず、デバイスの状態も全てのソースコンテキストと対象コンテキストで OFF に設定した。上記で述べたコンテキストウェアシステムにより各遷移のためにデバイス操作シーケンスを導出し、シーケンスに従って対象コンテキストまで遷移させた。

表 12 は、各遷移で導出されたデバイス操作シーケンスの消費電力量と実際に掛かった消費電力量の差、そして各遷移で再導出が行われた回数を表している。実験結果として、PathSim を導入したコンテキストウェアシステムが表 11 に示した全てのソースコンテキストから対象コンテキストまで遷移できることを確認できた。遷移 1 および遷移 2 はそれぞれ、約 0.004 kWh および 約-0.011 kWh というわずかな消費電力量の差でコンテキストを遷移できたことが分かる。また、遷移 2

ではデバイス操作シーケンスに完全に一致してコンテキストを遷移することができた。しかしながら、消費電力の差を比率としてみると、遷移1では実際より約32%多く、遷移2では約40%少なく、消費電力量を見積もっている。これらの誤差に関しては、今回カタログスペックに基づいた簡易的な消費電力予測を行っているために生じた誤差だと考えられる。そのため、デバイスごとの正確な消費電力モデルと出力モデルを構築し、より正確な遷移時間と消費電力量をシミュレートすることで、誤差の軽減ができると考えられる。

### 3.6 関連研究

提案した PathSim に関連する (1) スマートホーム/コンテキストウェアシステム (2) コンテキストの遷移に関するモデル化とシミュレーション, (3) スマートホームの効率的なデバイス制御, に関する既存研究について以下で概要を述べる。

スマートホーム [18] とコンテキストウェアシステムに関する研究 [13, 49, 50] が多く存在する。MavHome[18] では、家の状態 (コンテキスト) を感知し、自動的に快適なサービスをユーザに提供する知的エージェントの実現を目的に、住民の行動予測に基づいたコンテキストウェアサービスを提案している。しかしながら、サービスを提供する際に、エネルギー効率の良いデバイス制御を考慮していない。スマートスペースにおけるコンテキストウェアサービスはしばしばルール (トリガと動作のペア) の集合で指定される。CADEL[49] は、UPnP プロトコルを利用したコンテキストウェアなデバイス操作を実現するルールベースのミドルウェアである。また、Guら [50] は、オントロジーに基づいたコンテキストモデルを定義し、異なる粒度でコンテキストを扱うことができるサービス指向型コンテキストウェアミドルウェア (SOCAM) を提案している。これらの既存ミドルウェアは、ルールの集合でスマートホームに関する要求仕様を指定しているが、デバイスの制御処理を明示的に指定する必要がある。PathSim を用いることで、スマートホームの管理者が詳細なデバイスの制御処理を指定することなく、コンテキストを遷移させることが可能である。

コンテキストのモデル化と推論 [2], そしてコンテキスト遷移シミュレーション [51] を扱ういくつかの研究が存在する。文献 [2] では、基本的なモデルとして

オブジェクトロールベースモデル，空間モデル，オントロジーベースモデル，という3つのモデルが紹介されている．これらのモデルは，形式的に任意のコンテキストを表すことができる．しかしながら，これらのモデルでは，コンテキスト間の遷移に関しては形式的に定義していない．UbiREAL[51]は，仮想空間上にスマートスペースとUPnPプロトコルを実装したデバイスを再現するシミュレータである．このシミュレータは，空調デバイスによる温度変化のようなコンテキストの遷移を，スマートスペースのサイズやデバイスの出力などを考慮してシミュレートする．しかしながら，UbiREALは特定のコンテキストに到達するための最適コストのイベントシーケンスを導出する機能を具えていない．

近年，スマートスペースのデバイス进行操作することで，消費エネルギーの最適化を目指すような研究[19, 52]がいくつか行われている．Jahnら[19]は，デバイスごとの消費電力と累積コスト（エネルギー消費量）を，テレビや携帯電話の画面上にグラフィカルユーザインタフェースを介して，リアルタイムにユーザに提示するエネルギーアウェアなスマートホームを提案している．この研究では，電気エネルギーの動的価格設定をサポートしており，電気料金がしきい値を下回った場合にのみ，指定されたデバイスが起動する機能を提供している．しかし，この研究では，最小エネルギーを伴った複数サービスの協調制御をサポートしていない．柏本ら[52]は，省エネのために期間ごとに利用可能なエネルギー量が制限されているという仮定のもと，各家電のエネルギー消費量，ユーザの状況，ユーザの嗜好などを考慮して，居住者の快適さのレベルが最大化されるように，スマートホーム内の家電に利用可能なエネルギー量を準最適配分する手法を提案している．しかしながら，この研究では，各環境変数の値を変えることができるデバイスは1つだけと仮定しており，複数のデバイスを協調させながら，最小消費電力で特定のコンテキストに遷移することは考えていない．また，3.4節において述べたように，温度や湿度のような連続値とデバイスの設定値のような離散値を扱う動的システムは，ハイブリッドダイナミカルシステム[48]と呼ばれており，本章で対象としたスマートホーム上で動作するコンテキストアウェアシステムもハイブリッドダイナミカルシステムとみなすことが可能である．このようなハイブリッドダイナミカルシステムを最適制御する手法[53, 54, 55]もいくつか提案さ



れている。しかしながら、これらの研究の多くが、数理モデル化のみを行うもので、実システムへの実装や実システムで問題になる現象などは、ほとんど考慮されていない。また、提案手法のようにスマートホームのコンテキスト遷移を最短経路問題にモデル化し、解決する手法を提案している文献も存在せず、提案手法の新規性は高いと考えられる。

提案手法は、最適な消費電力量でスマートスペースを特定のコンテキストに遷移できる複数デバイス操作シーケンスを導出する最初の手法である。

### 3.7 まとめ

本章では、コンテキストウェアなスマートスペースシステムにおいて、2つのコンテキスト間の遷移に対する準最適コストのイベントシーケンスを導出するツール PathSim を提案した。

PathSim は、スマートスペースの有限個のコンテキスト遷移を適切に導出することで、準最適コストのシーケンスを導出できるツールである。PathSim は、スマートスペースの変数の値域を有限個に適切に分割し、(a) コンテキストと関連するイベントを最適に選択/決定し、(b) 選択したイベントを実行した時のコンテキストの遷移と関連する電力消費量をシミュレートし、そして (c) A\*アルゴリズムに基づいて準最適コストのデバイス操作シーケンスを導出する。

PathSim と単純なハイブリッドシステムを用いた評価実験により、PathSim は、単純なハイブリッドシステムよりおよそ 26%少ない消費電力量を持つシーケンスを導出できることが分かった。また、PathSim を導入したコンテキストウェアシステムを開発し、実環境実験を行った結果、PathSim で導出したデバイス操作シーケンスにより実環境のコンテキストを任意のコンテキストに遷移できることが確認できた。

今後は、PathSim がより精度の高いデバイス操作シーケンスを導出できるように、実測値を用いて様々なデバイスの出力対消費電力モデルを構築することを検討している。

## 4. 結論

本論文では、効率と実用性を考慮したコンテキストウェアシステムの実現を目指し、筆者がおこなった2つの研究について記述した。研究内容は、(1) 災害時における傷病者の救命率の最大化を目指した搬送計画手法およびシステムの研究、(2) スマートハウスにおいて任意の2つのコンテキスト間を最適コストで遷移させるデバイス操作シーケンス導出手法およびシステムの研究である。以下、得られた成果および今後の課題をまとめる。

電子トリアージタグと予測生存率を利用した搬送計画手法およびシステムでは、シミュレーションによる実験により、提案方式が、より多くの傷病者を救命可能であること、そして、実時間で計画が可能であるため傷病者の急変などにも実時間で対応可能なことを確認した。本論文では予測生存率モデルとして、救急医療の現場で用いられるカーラーの救命曲線に近似する4つの曲線作成し使用した。しかしながら、実際の傷病者の予測生存率は傷病や心拍数などのパラメータによって異なるため、多くのモデルが必要となる。これらのモデルの作成には、実際に電子トリアージタグを使用した多くの医療データが必要である。これは、現在、世界中で電子トリアージタグが開発されていること、また、一部の病院で導入が始まっていることから、今後データが集まり解析が進むことで、解決されることが考えられる。

最適コストのデバイス操作シーケンス導出手法およびシステムでは、提案したPathSimを用いて導出したデバイス操作シーケンスと、単純なハイブリッドシステムの動作を想定して導出したデバイス操作シーケンスで消費する消費電力量をシミュレーションで算出し比較した結果、PathSimで導出した方が省エネルギーでコンテキストを遷移可能であることを確認した。また、PathSimを採用したコンテキストウェアシステムを実装し、実環境実験を行った結果、導出したデバイス操作シーケンスで目的のコンテキストに正しく遷移できることを確認した。実環境実験を通して、カタログスペックのデバイスの消費電力および出力と実際の消費電力および出力の間には差が存在したことから、今後は、より詳細なシミュレーションを行うために、正確なデバイスの電力モデルと出力モデルを構築し再実験を行うことを検討したい。また、提案したPathSimはできるだけ少ないコ

ストのデバイス操作シーケンスを導出できることから、既に構築済みのスマートスペースで動作するコンテキストウェアシステムの省エネ評価にも利用可能だと考えられる。

本論文では、近年特に注目が高まっている災害とスマートハウスに対して効果的なコンテキストウェアシステムを提案した。しかしながら、コンテキストウェアシステムの研究分野は比較的新しく本論文で対象とした分野以外にも様々な分野に最適化アルゴリズムを適用できる可能性を秘めている。そのため、今後は、コンテキストウェアシステムの適用がより有効な分野および適用による効果の検討を行っていききたい。

## 謝辞

指導教員である安本慶一 教授には，素晴らしい研究環境を与えて頂くとともに，時に優しく時には厳しく適切な御指導および御助言を頂きました．ここに深く感謝致します．

伊藤実 教授には，本研究の副指導教員として，本論文の作成および研究にわたり有益な情報や御助言を賜り，御指導して頂きました．ここに厚く感謝致します．

また，副指導教員として関浩之 教授には，本論文の作成および研究にわたり，適切かつ鋭い御助言を賜りました．心より感謝致します．

AUS の Khaled El Fakih 准教授には，本研究を進めるにあたり，多くの御助言を頂きました．ここに深く感謝致します．

また，研究に関してのみならず，研究室の生活から国際会議への同行など多くの場面で手助けして頂いた荒川准教授，玉井助教，および大阪大学 孫為華 助教に深く感謝致します．

金岡恵 事務補佐員 には，研究費管理や物品購入などの事務手続き等をお世話して頂きました．ここに厚く感謝致します．

卒業された諸先輩方，同期，後輩，そして現在の研究室の皆様には，日頃から公私に渡りお世話になりました．ここに深く感謝致します．

最後に，これまで長い間支えてくれた家族，友人に感謝します．

## 参考文献

- [1] Krumm, J. (Ed.). *Ubiquitous computing fundamentals*, pp.321–348, CRC Press, 2009.
- [2] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D.: “A survey of context modelling and reasoning techniques,” *J. Pervasive and Mobile Computing*, Vol. 6, No. 2, pp. 161–180, 2010.
- [3] Henricksen, K., Indulska, J. and Rakotonirainy, A.: “Modeling context information in pervasive computing systems,” *J. Pervasive Computing*, Vol. 2414, pp. 167–180, 2002.
- [4] Henricksen, K. and Indulska, J.: “Developing context-aware pervasive computing applications: Models and approach,” *J. Pervasive and mobile computing*, Vol. 2, No. 1, pp. 37–64 2006.
- [5] Dey, A.: “Understanding and using context,” *J. Personal and Ubiquitous Computing*, Vol.5, No. 1, pp. 4–7, 2001.
- [6] Leonhardi, A., Kubach, U., Rothermel, K. and Fritz, A.: “Virtual information towers – A metaphor for intuitive, location-aware information access in a mobile environment,” *Proc. ISWC 1999*, pp. 15–20, 1999.
- [7] Chen, H., Perich, F., Finin, T.W. and Joshi, A.: “SOUPA: Standard ontology for ubiquitous and pervasive applications,” *Proc. MobiQuitous 2004*, pp. 258-267, 2004.
- [8] Wang, X.H., Zhang, D.Q., Gu, T. and Pung, H.K.: “Ontology based context modeling and reasoning using OWL,” *Proc. CoMoRea 2004*, pp. 18–22, 2004.
- [9] Bao, L. and Intille, S.S.: “Activity recognition from user-annotated acceleration data” *J. Pervasive Computing*, Vol. 3001, pp. 1–17, 2004.

- [10] Voigtmann, C., Schutte, C., Wacker, A. and David, K.: “A new approach for distributed and collaborative context prediction,” *Proc. CoMoRea 2013*, pp. 20–24, 2013.
- [11] Roggen, D., Lukowicz, P., Ferscha, L., del R, M., Troster, G. and Chavarriaga, R.: “Opportunistic human activity and context recognition,” *IEEE J. Computer*, Vol. 46, No.2, pp. 36–45, 2013.
- [12] Henricksena, K. and Indulskab, J.: “Developing context-aware pervasive computing applications: Models and approach,” *Pervasive and Mobile Computing*, Vol. 2, No. 1, pp. 37-64, 2006.
- [13] Baldauf, M., Dustdar, S. and Rosenberg, F.: “A survey on context-aware systems,” *J. Ad Hoc and Ubiquitous Computing*, Vol. 2, No. 4, pp. 263-277, 2007.
- [14] Panzieri, S., Pascucci, F. and Ulivi, G.: “An outdoor navigation system using GPS and inertial platform,” *IEEE/ASME Trans. Mechatronics*, Vol. 7, No. 2, pp. 134–142, 2002.
- [15] 武兵, 村田佳洋, 柴田直樹, 安本慶一, 伊藤 実: “天気変化を考慮した観光スケジュール群の探索アルゴリズム” 情報処理学会論文誌 数理モデル化と応用, Vol.3, No.1, pp.87-97, 2010.
- [16] Chan, M., Campo, E., Esteve, D. and Fourniols, J.Y.: “Smart homes—current features and future perspectives,” *J. Maturitas*, Vol. 64, No. 2, pp. 90–97, 2009.
- [17] Anliker, U., Ward, J.A., Lukowicz, P., Troster, G., Dolveck, F., Baer, M., Keita, F., Schenker, E.B., Catarsi, F., Coluccini, L., Belardinelli, A., Shklarski, D., Alon, M., Hirt, E., Schmid, R. and Vuskovic, M.: “AMON: a wearable multiparameter medical monitoring and alert system,” *Trans. Information Technology in Biomedicine*, Vol. 8, No. 4, pp. 415-427, 2004.

- [18] Cook, D. J., Youngblood, M., Heierman III, E. O., Gopalratnam, K., Rao, S., Litvin, A. and Khawaja, F.: “MavHome: An agent-based smart home,” *Proc. IEEE PerCom 2003*, pp. 521-524, 2003.
- [19] Jahn, M., Jentsch, M., Prause, C. R., Pramudianto, F., Al-Akkad, A. and Reiners, R.: “The energy aware smart home,” *Proc. of 5th IEEE Int. Conf. on Future Information Technology (FutureTech2010)*, pp. 1–8, 2010.
- [20] 大友康裕：プレホスピタルMOOK シリーズ4 多数傷病者対応，永井書店，2007.
- [21] 東野輝夫：災害時救命救急支援を目指した人間情報センシングシステム，入手先〈<http://etriage.jp>〉（参照 2011-10-22）.
- [22] Champion, H.R., Copes, W.S. and Sacco, W.J.: “The major trauma outcome study: Establishing national norms for trauma care,” *J. Trauma*, Vol.30, pp.1356–1365, 1990.
- [23] 藤木直子，阪本雄一郎，本村陽一ほか：ベイジアンネットワークを用いた生存率予測モデルの統計学的学習と評価，人工知能学会全国大会，2009.
- [24] 日本外傷学会，日本救急医学会：改訂第3版 外傷初期医療ガイドライン JATEC<sup>TM</sup>，へるす出版，2009.
- [25] 福田正輝，高山純一，中山晶一郎：三次救急搬送活動を対象とした医療情報デジタル伝送システム運用のためのアンテナ基地局配置方策の検討，土木学会土木計画学研究発表会講演集，Vol.39, pp.88, 2009.
- [26] 福田正輝，高山純一，中山晶一郎：三次救急搬送活動を対象とした医療情報デジタル伝送システム運用のためのアンテナ基地局配置方策に関する研究，土木学会中部支部研究会発表会公演概要集，pp.397 – 398, 2010.
- [27] 守谷 俊，丹正 勝久：救命救急活動の現状—どのようなプレホスピタルケアが必要なのか—，国際交通安全学会誌，Vo.34, No.3, pp.6–15, 2009.

- [28] Benson, M., Koenig, K.L. and Shultz, C.H.: “Disaster triage: START, then SAVE-A new method of dynamic triage for victims of a catastrophic earthquake,” *Prehospital Disaster Med*, Vol.11, No.2, pp.117–124, 1996.
- [29] Jotshi, A., Gong, Q. and Batta, R.: “Dispatching and routing of emergency vehicles in disaster mitigation using data fusion,” *Socio-Economic Planning Sciences*, Vol.43, No.1, pp.1–24, 2009.
- [30] Gao, T., Massey, T., Selavo, L., Crawford, D., Chen, B., Lorincz, K., Shnyder, V., Hauenstein, L., Dabiri, F., Jeng, J., Chanmugam, A., White, D., Sarrafzadeh, M. and Welsh, M.: “The advanced health and disaster aid network: A light-weight wireless medical system for triage,” *IEEE Trans. BCAS*, Vol.1, No.3, pp.203–216, 2007.
- [31] Gao, T., Pesto, C., Selavo, L., Chen, Y., Ko, J. G., Lim, J.H., Terzis, A., Watt, A., Jeng, J., Chen, B., Lorincz, K. and Welsh, M.: “Wireless medical sensor networks in emergency response: Implementation and pilot results,” *Proc. HST’08*, pp.187–192, 2008.
- [32] 楠田 純子, 木山 昇, 内山 彰ほか: 無線センサーネットワークを利用した電子トリアージシステムの実現, 電子情報通信学会技術研究報告 (MoMuC, モバイルマルチメディア通信), vol.109(204), pp.33–38, 2009.
- [33] 木山 昇, 楠田 純子, 内山 彰ほか: 災害時救急救命支援に向けた電子トリアージシステム的设计開発, 情報処理学会マルチメディア, 分散, 協調とモバイル DICOMO2009 シンポジウム論文集, pp.1837–1848, 2009.
- [34] 安倍 史江, 山本 匠, 西垣 正勝: 人体通信による電子トリアージタグへの情報伝達: システムの実装, 情報処理学会マルチメディア, 分散, 協調とモバイル DICOMO2009 シンポジウム論文集, pp.1849–1854, 2009.
- [35] Gunawan, L.T., Voshell, M., Oomes, A.H.J. and Woods, D.: “Envisioning collaboration at a distance for the evacuation of walking wounded,” *Proc. ISCRAM’07*, pp.431–437, 2007.



- [36] 栖関 邦明, 杉山 阿葵, 長橋 健太郎, 岡田 謙一: 治療優先度を付加した自動トリアージシステムの提案, 情報処理学会論文誌, Vol.51, No.1, pp.2–13, 2010.
- [37] Champion, H.R., Sacco, W.J., Copes W.S., Gann, D.S., Gennarelli, T.A. and Flanagan, M.E.: “A revision of the trauma score,” *J. Trauma*, Vol.29, pp.623–629, 1989.
- [38] Baker, S.P., O’Neil, B., Haddon, W.Jr. and Long, W.B.: “The injury severity score: a method for describing patients with multiple injuries and evaluating emergency care,” *J. Trauma*, Vol.14, pp.187–196, 1974.
- [39] 日本外傷診療研究機構: 日本外傷データベース, 入手先 (<http://www.jtcr-jatec.org/traumabank/index.htm>) (参照 2014-1-6) .
- [40] West, T.A., Rivara, F.P., Cummings, P., Jurkovich, G.J. and Maier, R.V.: “Harborview assessment for risk of mortality: an improved measure of injury severity on the basis of ICD-9-CM,” *J. Trauma*, Vol.49, pp.530–541, 2000.
- [41] Osler, T., Glance, L., Buzas, J.S, Mukamel, D., Wagner, J. and Dick, A.: “A trauma mortality prediction model based on the anatomic injury scale,” *Ann. Surg*, Vol.247, pp.1041–1048, 2008.
- [42] Weiser, M.: “The computer for the 21st century,” *Scientific American*, Vol. 265, No. 3, pp. 94–104, 1991.
- [43] Weiser, M.: “Some computer science issues in ubiquitous computing,” *Communication of the ACM*, Vol. 36, No. 7, pp. 75–84, 1993.
- [44] Edwards, W.K. and Grinter, R.E.: “At home with ubiquitous computing: Seven challenges,” *Proc. UbiComp 2001*, pp. 256-272, 2001.
- [45] Hart, P.E., Nilsson, N.J. and Raphael, B.: “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Trans. SSC*, Vol. 4, No. 2, pp. 100–107, 1968.

- [46] Tetens, O.: “Uber einige meteorologische Begriffe,” *Zeitschrift fur Geophysik*, Vol. 6, pp. 297-309, 1930.
- [47] 株式会社クレセル : “Q&A 快適な温度・湿度湿度は?”  
入手先 <http://www.creecer.jp/Q-A/HTML/A-11.html> (参照 2014-2-5) .
- [48] Henzinger, T.: “The theory of hybrid automata,” *Proc. 11th Ann. Symp. Logic in Computer Science*, pp. 278–292, 1996.
- [49] Nishigaki, K., Yasumoto, K., Shibata, N., Ito, M. and Higashino, T.: “Framework and rule-based language for facilitating context-aware computing using information appliances,” *Proc. 1st Int. Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet (SIUMI’05) (ICDCS’05 Workshop)*, pp. 345–351, 2005.
- [50] Gu, T., Pung, H. K., and Zhang, D. Q.: “A service-oriented middleware for building context-aware services,” *J. Network and computer applications*, Vol. 28, No. 1, pp. 1–18, 2005.
- [51] Nishikawa, H., Yamamoto, S., Tamai, M., Nishigaki, K., Kitani, T., Shibata, N., Yasumoto, K. and Ito M.: “UbiREAL: Realistic smartspace simulator for systematic testing,” *Proc. UbiComp 2006*, LNCS4206, pp. 459-476, 2006.
- [52] Kashimoto, Y., Ogura, K., Yamamoto, S., Yasumoto, K. and Ito, M.: “Saving energy in smart homes with minimal comfort level reduction,” *Workshop Proc. IEEE PerCom 2013*, pp. 372-376, 2013.
- [53] Bemporad, A. and Morari, M.: “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, Vol. 35, No. 3, pp. 407-427, 1999.
- [54] 浅野一哉, 津田和呂, Bemporad A., Manfred M. : “ハイブリッドシステムの予測制御とそのプロセス制御への適用”, システム制御情報学会誌, Vol. 46, No. 3, pp.110–119, 2002.

- [55] 河村謙, 佐藤洋平, 久村富持: “複数バッファ切り替えシステムのMLDによるモデル化と制御”, 電子情報通信学会技術報告, Vol. 102, No. 96, pp.1-4, 2002.

## 業績リスト

### 論文誌

1. 水本旭洋, 孫為華, 安本慶一, 伊藤実: “多数傷病者事故時での救命率向上のための電子トリアージの利用を前提とした搬送計画システム”, 情報処理学会論文誌, Vol. 53, No. 7, pp. 1745–1756 (July 2012)  
(2章に対応)

### 国際会議 (査読有り)

1. Teruhiro Mizumoto, Weihua Sun, Keiichi Yasumoto, Minoru Ito: “Transportation Scheduling Method for Patients in MCI using Electronic Triage Tag,” *Proc. of the 3rd Int’l. Conf. on eHealth, Telemedicine, and Social Medicine (eTELEMED 2011)*, pp. 156–163 (February 2011).  
(2章に対応)
2. Teruhiro Mizumoto, Khaled El-Fakih, Keiichi Yasumoto: “PathSim: A Tool for Finding Minimal Energy Device Operation Sequence for Reaching a Target Context in Smart-Home,” *the 10th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC-2013)* (December 2013),  
(3章に対応)

### 国内学会

1. 水本旭洋, 孫為華, 安本慶一, 伊藤実: “電子トリアージタグに基づいた傷病者搬送計画システムの提案”, 情報処理学会研究報告, 2010-DPS-142(39), pp.1-8, (March 2010).  
(2章に対応)
2. 水本旭洋, 孫為華, 安本慶一, 伊藤実: “多数傷病者事故での救命率向上を目指した電子トリアージタグに基づく傷病者搬送計画手法”, 第18回マルチ

メディア通信と分散処理ワークショップ (DPSWS2010), pp.43-48 (October 2010).

(2章に対応)

3. 水本旭洋, 柴田直樹, 安本慶一, El-Fakih, Khaled : “コンテキストウェアシステムの実装をテストするフレームワーク SmartTEST の提案”, DI-COMO2012 シンポジウム論文集, pp. 1050-1059 (July 2012) .

(3章に対応)