

パラメータ最適化のためのスマートコミュニティシミュレーション基盤の設計と実装

牧野 義樹^{1,a)} ジャヴィッド サハル^{1,b)} リム 勇仁^{1,c)} 丹 康雄^{1,d)}

概要: 筆者らはコミュニティエネルギーマネジメントシステム (CEMS) 等の大規模実験を行うためにスマートコミュニティシミュレータの設計と開発を行っている。本システムを利用することで様々なパラメータを変更したシミュレーションを実行することが可能となっている。一方、シミュレータを利用してパラメータ最適化を実現する手法が知られているが、本シミュレータを利用してパラメータの最適化を行うことは簡単に実現することができない。そこで、MATLAB 等のツールからスマートコミュニティシミュレータを実行可能とし、シナリオに応じた最適なパラメータを生成することが可能なシミュレーション基盤の設計と実装を新たに行った。本稿では、その設計と実装について述べる。

キーワード: スマートコミュニティ, CEMS, シミュレーション, パラメータ最適化

1. はじめに

現在、日本国内でも様々な地域でスマートコミュニティの実証実験が行われている。しかしながら、スマートコミュニティは巨大なシステムであり、動作試験のために実際にシステムを構築することは簡単なことではない。また、実際にスマートコミュニティを構築し利用できる環境があったとしても、病院や工場の動作を停止する危険がないようにするなど、住人の生活等への影響がないことが保証される範囲でしか実験することができない。さらに、コミュニティの設計を変更した場合の実験も実システムを用いた場合には実現することが容易ではない。

そこで、コンピュータを利用したシミュレーションを行う利用することが考えられている。筆者らは CEMS 等の大規模実験を行うためにスマートコミュニティシミュレータの設計と開発 [1][2] を行っている。本シミュレータは各施設の動作をエミュレータやシミュレータによる計算により行うエミュレータベースのシミュレータとなっている。

本シミュレータを用いることでコミュニティをシミュレーションを行うことは可能であるが、その利用方法については利用者に委ねられているのが現状である。そこで、

本シミュレータの実行をより容易にし、また変更可能な様々なパラメータを最適化することを可能とするシミュレーション基盤の構築を行っている。本システムを利用することで、利用者はスマートコミュニティの最適なパラメータを探し出すシステムを構築することが容易となる。

2. スマートコミュニティシミュレータ

筆者らが構築中であるスマートコミュニティシミュレータは様々な要素をモジュールとして実装されたソフトウェアからなる。モジュールはエミュレータで実現されることもソフトウェアで実装されたシミュレータで実現することも可能となっている。コミュニティシミュレータ内の全モジュールの構成は図 1 のようになる。これらのモジュールは一台以上のシミュレータノードで動作する。シミュレータノードを複数利用することも可能であり、その場合にはシミュレーションを分散して実行することが可能である。また、本シミュレータの動作設定はコミュニティプロファイルと呼ばれる XML ファイル群によって行われる。本章ではそれぞれのモジュールとコミュニティプロファイルについて説明する。

2.1 シミュレータ管理モジュール

シミュレータ全体の管理を行うモジュールである。管理ノード上で動作を行い、後述するコミュニティプロファイルの読み込み、シミュレーションを行うコミュニティ全体の設定を各シミュレータノードに割り当てる。また、シ

¹ 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
a) m-yoshi@jaist.ac.jp
b) saher@jaist.ac.jp
c) ylim@jaist.ac.jp
d) ytan@jaist.ac.jp

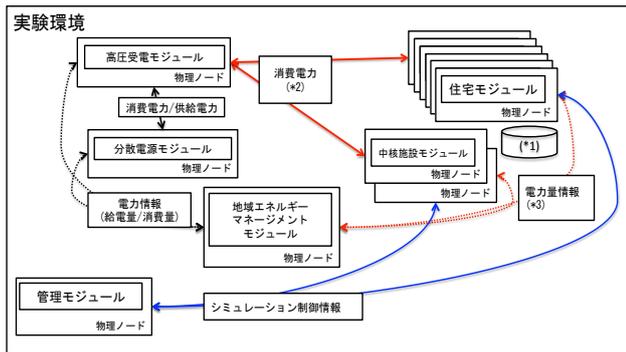


図 1 スマートコミュニティシミュレータのモジュール

シミュレーションの開始のリクエストメッセージを送信し、シミュレータ間で同期した動作を実現するための同期メッセージの各シミュレータノードとの間で送受信する。

2.2 地域エネルギー管理モジュール

地域全体のエネルギー管理を行う Community Energy Management System (CEMS) を実装するモジュールである。他のモジュールから電力エネルギー、熱エネルギーの利用状況の情報を収集する。またその結果に基づきデマンドレスポンス命令を各モジュールに転送することで、シミュレートされたコミュニティ全体の消費電力の制御を行う。本モジュールの動作を変更することでコミュニティ全体の制御方針を変更することが可能である。

2.3 住宅モジュール

住宅をエミュレートするモジュールである。住宅で利用及び余剰電力エネルギーと熱エネルギーを計算し地域エネルギー管理モジュール及び、高圧受電モジュールと熱受信モジュールに送信する。また Home Energy Management System (HEMS) を実装することで、地域エネルギー管理モジュールから送信されるデマンドレスポンス命令等により、住宅内の家電を制御しエネルギーの利用量の制御を行う。ここで利用されるメッセージ通信は OpenADR(1) をベースにした独自形式の物を用いているが、他のモジュールとの整合性が取れている場合には実システムで利用されるプロトコルを用いることも可能である。この場合には、シミュレータ全体で実時間に同期した処理を行うことで、実 HEMS システムをモジュールの一部に含むシミュレーションの実行を行うことも可能となる。

2.4 集合住宅モジュール

集合住宅をエミュレートするモジュールであり、住宅モジュールの一種となる。集合住宅で利用及び余剰電力エネルギーと熱エネルギーを計算し地域エネルギー管理

メントモジュール及び、高圧受電モジュールと熱受信モジュールに送信する。また Building Energy Management System (BEMS) や HEMS を実装することで、地域エネルギー管理モジュールから送信されるデマンドレスポンス命令等を受信することで、住宅内の家電を制御しエネルギーの利用量の制御を行う。

2.5 分散電源モジュール

地域に存在する分散電源を模擬するモジュールである。太陽光発電装置、コジェネレーションシステム、蓄電池などからなり、その動作を模擬するエミュレータからなる。電力の供給や充電の状況は本モジュールのシステムにより監視及び管理され、その状況は地域エネルギー管理モジュールに送信される。地域エネルギー管理モジュールは、必要に応じて分散電源モジュールに動作の変更の要求を送信することもある。

2.6 スマートコミュニティプロフィール

シミュレータを利用するために、各シミュレータノードに管理モジュールや様々なファシリティのモジュールとして実行されるシミュレータソフトウェアをインストールし、システム全体のセットアップを先に行う必要がある。また、利用者はシミュレートを行いたいコミュニティに合わせてスマートコミュニティプロフィールと呼ばれるコミュニティ定義ファイル群も記述する必要がある。利用者はシステムのセットアップ後、シミュレータ管理モジュールにシミュレータの開始を要求する。スマートコミュニティプロフィールは XML で記述するようになっており、以下の 5 種類からなる。

- コミュニティ全体定義
- シミュレーション情報定義
- 各施設モジュールの定義
- コミュニティ内分散電源の定義
- 各施設の設備定義

ここでは各プロフィールについて説明する。

2.6.1 コミュニティ全体定義

本プロフィールでコミュニティ全体の定義を行う。シミュレートを行うコミュニティ内に存在する中核施設、商業施設、個別住宅、集合住宅、コミュニティ内分散電源の数や種類の定義を行うことで、コミュニティ全体を簡潔に記述できる形式となっている。各施設の詳細については施設毎のプロファイルで別途記述することとなる。

2.6.2 シミュレーション情報定義

シミュレーションの内部で利用される開始時刻と終了時刻を定義することで、シミュレータがシミュレートする日時を指定する。

2.6.3 各施設モジュールの定義

施設毎のプロファイルである。中核施設、商業施設、個

別住宅施設、集合住宅施設が存在し、施設毎に記述する内容は少し異なる。例えば個別住宅では以下の項目の設定を行う。

2.6.4 個別住宅種類

本プロファイルのユニークな名前を設定する

2.6.5 電力需用プロファイル

電力需用の計算方法に関するプロファイルを指定する

2.6.6 熱受給プロファイル

熱受給の計算方法に関するプロファイルを指定する

2.6.7 契約電力

住宅の契約電力を記述する

2.6.8 契約熱量

住宅の契約熱量を記述する

2.6.9 分散電源

必要に応じて住宅内にある分散電源のプロファイルを指定する。

2.6.10 xEMS 装置

利用するエネルギー管理システムを指定する。

2.6.11 コミュニティ内分散電源の定義

コミュニティ内に存在する分散電源の構成を定義する。現在は発電量と蓄電量、蓄熱量の時間的な推移の情報を指定する簡易な物となっている。

3. スマートコミュニティシミュレーション基盤の設計

本スマートコミュニティを利用するためには、利用者はスマートコミュニティシミュレータの環境構築からコミュニティプロファイルの作成、その動作の開始まで行う必要がある。またスマートコミュニティシミュレータの環境が先に構築をされていたとしても、同時に実行できるシミュレーションの数は一つに限られてしまう。そこで筆者らはこれらの問題を解決するためにスマートコミュニティシミュレータの実行管理を行うシステムとしてスマートコミュニティシミュレーション基盤の設計を行った。その概要について図2に示す。ここではクライアントとしてMATLAB[3]を利用することとした。本シミュレーション基盤はMATLAB等のアプリケーションからパラメータを与えられ、そのパラメータに基づいてコミュニティシミュレータの初期化と実行を行う。また、その実行が終了するとMATLAB等からシミュレーションの結果を得られるようになっている。本章では本シミュレーション基盤の詳細について述べる

3.1 パラメータ最適化

パラメータ最適化は、ある目的関数を最大化、あるいは最小化するという関数のパラメータを探し出すことで実現される。本システムではスマートコミュニティシミュレータを関数として扱えるようにすることで、スマートコミュ

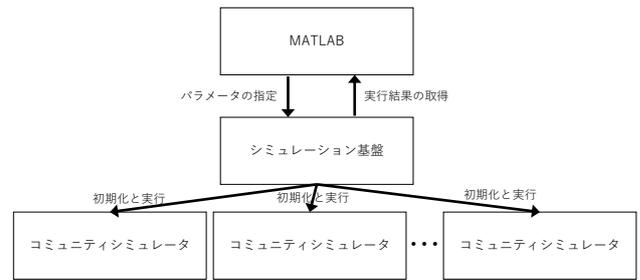


図2 シミュレーション基盤

ニティシミュレータのパラメータの最適化を可能とする。このようなシステムを実現するためには、スマートコミュニティシミュレータを一つの関数として呼び出すことが必要となる。さらに、その関数を並列に呼び出すことが可能であれば、並列計算によるパフォーマンスの向上が期待できる。このようにシミュレータを関数として実行するためのシステムを実現するためには以下の機能を持ったシミュレーション基盤を構築する必要がある。このような機能が存在すれば、コミュニティシミュレータを関数のように扱うためのプログラムを構築することが容易になる。

- パラメータの指定シミュレータのパラメータの指定を容易に行うことが可能なインターフェースを持っている。
- シミュレーション環境の構築の起動パラメータの内容に基づきシミュレータの実行を開始することが可能である。シミュレーションノードが存在する場合にはシミュレーションを実行することが可能である。この時、十分な台数のシミュレーションノードが存在する場合にはコミュニティシミュレータは並列で動作することになる。シミュレーションノードが足りない場合には、必要なノード数を得られるまでシミュレーションの実行を待ってからシミュレータの実行を行う。
- シミュレーション結果の取得シミュレーションの結果を容易に得ることが可能である。これによりシミュレーション結果の評価を容易に行うことができる。

本節ではそれぞれの機能について述べる。

3.1.1 パラメータの指定

本機能により、様々なパラメータでシミュレータを実行することが可能となる。筆者らが開発を行っているスマートコミュニティシミュレータはコミュニティプロファイルと呼ばれるXMLファイルで設定を受け渡す。そのXMLファイルの内容を書き換えることによりコミュニティシミュレータの動作を変更することが可能となる。コミュニティプロファイルを常に全て構築するのは無駄が多くなるため、すでに存在するコミュニティプロファイルを基盤として、一部の設定を変更することで新たなコミュニティプロファイルの生成を行うようにした。ここで与えられる実

行設定ファイルの例を以下に示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<autorun>
  <command>run</command>
  <id>001</id>
  <communityprofile>base.xml</communityprofile>

  <simulation>
    <time>
      <begin>2014-07-31T00:00:00</begin>
      <end>2014-07-31T01:00:00</end>
    </time>
  </simulation>

  <params>
    <param>
      <name>house[house1].num</name>
      <value>100</value>
    </param>
  </params>
</autorun>
```

この実行設定ファイルにより指定される要素は以下の通りである。

command

スマートコミュニティシミュレーション基盤への要求内容を記述する。ここでは run を指定することで、新たなシミュレーションの実行を要求している。

id

シミュレーションの ID を指定する。この ID が同一のシミュレーションが、内容が同一のコミュニティプロファイルで実行されている場合には、このシミュレーションはすでに実行されたものとして扱われる。また、ID が同一であっても内容が異なるコミュニティプロファイルの実行が要求された場合には、以前の実行結果は消去されて新たなシミュレーションが実行される。

communityprofile

基盤となるコミュニティプロファイルを指定する。

simulation

time 要素によりシミュレーションを行う時間を指定する。

params

param 要素により基盤となるコミュニティプロパティへの変更箇所を指定する。

この設定では、communityprofile 要素の内容が base.xml

となっている。これは、コミュニティプロファイルのベースとして base.xml を利用するように指定している。また、params 要素により、house1 という住宅が 100 戸存在するシミュレーションを行うように指定している。またシミュレーションを行う仮想的な時間は 2014 年 7 月 31 日の 0 時から 2014 年 7 月 31 日の 1 時までであるという指定をしている。これ以外の要素については base.xml の内容が利用されるようになる。

3.1.2 シミュレーション環境の構築と起動

スマートコミュニティシミュレーション基盤は、実行設定ファイルを受け取り、その情報を内部で管理しているジョブ管理データベースに格納する。ジョブ管理データベースに格納された実行設定ファイルの内容はシミュレータジョブ管理システムにより読み出され、必要な数のシミュレーションノードを利用可能であれば、必要な数のシミュレーションノードを新たなシミュレーション用に割り当て、それらのノードを利用してスマートコミュニティシミュレータを実行するために必要な初期化を行う。初期化に成功すると、スマートコミュニティシミュレータの実行を行う。

これにより、多数のシミュレーションノードが存在する場合には、スマートコミュニティシミュレータを並列に動作させることが可能である。シミュレーションノード一台で実行可能な住宅等のシミュレーション数はシステムの設定ファイルで指定を行う。また、それぞれのコミュニティシミュレータは同じ IP アドレス体系を利用することが考えられるため、それぞれのスマートコミュニティシミュレータはネットワーク的に分離している必要がある。本システムでは VLAN の技術を用いてネットワークの分割を実現することとした。

3.1.3 シミュレーション結果の取得

本システムではシミュレーション結果を格納するデータベースが動作し、スマートコミュニティシミュレータの実行結果は全てそのデータベースに格納される。データの登録や取得には HTTP プロトコルを利用した REST インタフェースを用いた設計となっている。本システムを利用するアプリケーションは REST インタフェースを利用することで、シミュレーションの結果を得ることが可能であり、そのシミュレーション結果の評価を行うことが可能である。

これらの機能を利用して、パラメータの指定やシミュレーション結果の評価を行う関数を作成することで、既存のパラメータ最適化ソフトウェアから関数として利用することが可能となる。本設計ではパラメータ最適化を行うためのアプリケーションとして MATLAB を利用することとした。

3.2 スマートコミュニティシミュレーション基盤

ここまでで説明した機能を実現するために、スマートコ

コミュニティシミュレーション基盤は以下の機能を持つ。

- アプリケーションインタフェースサブシステム
- シミュレーションノード管理サブシステム
- シミュレーションジョブ管理サブシステム

3.2.1 アプリケーションインタフェースサブシステム

クライアントからの要求を受信するサブシステムである。ネットワークを介して他のシステム上のアプリケーションからの要求を受信することが可能である。アプリケーションからは主にシミュレーションの実行の要求が行われるが、それ以外にもシミュレーションの実行状況の要求を受け付けることが可能となっている。シミュレーションの実行要求を受信した場合にはシミュレーションジョブ管理サブシステムにスマートコミュニティシミュレータの実行要求を行う。

3.2.2 シミュレーションノード管理サブシステム

スマートコミュニティシミュレータを動作することが可能なシミュレーションノードの管理を行うサブシステムである。シミュレーションノードが利用中であるという情報も内部で保持する。シミュレーションジョブ管理からシミュレーションノードの利用要求があった場合には、利用可能なノードを提供する。さらに、シミュレーション結果を格納するデータベースのノードや VLAN の設定を実行するノードの情報をシミュレーションジョブ管理サブシステムに提供する。

3.2.3 シミュレーションジョブ管理サブシステム

本サブシステムの主な機能は、シミュレーション基盤インタフェースからスマートコミュニティシミュレータの実行要求を受け付けることである。本サブシステム内ではジョブ管理データベースが動作しており、スマートコミュニティシミュレータの実行要求は未実行であるという状態が保存される。シミュレーションジョブ管理サブシステムはシミュレーションノード管理サブシステムに利用可能なノード数の確認を行い、その数がスマートコミュニティシミュレータを実行するのに十分である場合には、そのノードを利用中であるという情報を更新し、VLAN を設定することで隔離されたネットワークの環境を構築しスマートコミュニティシミュレータの実行を開始する。その際、ジョブ管理データベースにはシミュレーションが実行状態であるというマークをする。

4. スマートコミュニティシミュレーション基盤の実装

以上のシステムの実装を行いスマートコミュニティシミュレーション基盤を用いたシステム全体を実現した。実装したシステムの全体構成を図3に示す。スマートコミュニティシミュレーション基盤とデータベースは異なるノードで動作させることも可能であるが、現在は同じマシンで動作するようにした。本実装ではシミュレーションノード

を2台利用しており、図3では2台のシミュレーションノードを利用するスマートコミュニティシミュレータが2つ実行中である状態を示している。

また、今回の実装では管理用ネットワークと実験用ネットワークに異なるネットワークスイッチを利用して実装を行った。管理用ネットワークは固定 IP アドレスを利用することで各ノード間の通信を自由に行うことができる構成になっている。実験用ネットワークスイッチはシミュレーション基盤からの要求により各コミュニティシミュレータごとに異なる VLAN に属するように設定される。そのため、コミュニティシミュレータ内の通信は仮想的に異なるネットワークを利用することになるため、同一の IP アドレス体系であっても自由に設定できるようになっている。

また、本実装では大規模なテストベッドである StarBED[4][4] で利用されている管理システムである SpringOS の一部の機能を利用することで VLAN の設定を行う実装になっている。SpringOS の機能を利用することで、異なるベンダのネットワークスイッチでも統一した方式で VLAN の設定を行うことが可能となる。現在の実装では、SpringOS もスマートコミュニティシミュレーション基盤と同一のノード上で動作させている。

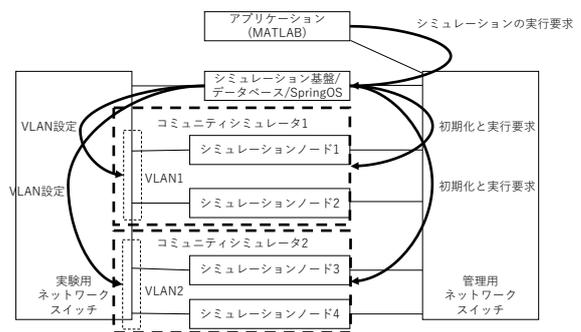


図3 実装したシステムの全体構成

さらに、本システムを利用した Python のプログラムを作成した。MATLAB から Python のプログラムを実行する仕組みを利用することで、そのプログラムを呼び出すことが可能であり、MATLAB の関数としてコミュニティシミュレータの実行を行う関数を実現している。

本システムを利用した MATLAB のアプリケーションの例として粒子群最適化 (PSO)[6] が動作する実装を行った。PSO は各パラメータの設定を粒子としてみだてて変化させていくことで、最適な解を見つけるという非線形計画問題のためのアルゴリズムの一つである。PSO では適切な解を得られるまで相応の繰り返し実行が必要である。また粒子の数だけ並列で動作させることでシステムを高速化することが可能である。

5. おわりに

本稿では、スマートコミュニティシミュレータを利用したパラメータ最適化のためのシミュレーション基盤の設計と実装について説明を行った。本基盤を利用することでMATLABなどを利用することでスマートコミュニティのプロパティ最適化を行うことが可能となる。今後は本システムを利用したパラメータの最適化を実際に行い、その評価を行っていく予定である。

謝辞 本研究の一部は「先進的 ICT 国際標準化推進事業 (スマートコミュニティにおけるエネルギーマネジメント通信技術)」の委託研究の成果である。

参考文献

- [1] 牧野義樹, 藤田浩, リム勇仁, 丹康雄, "マルチエージェントシミュレーションを指向するエミュレータ混合スマートコミュニティシミュレータ", 電気学会研究会 スマートファシリティ研究会, SMF-15-010, Jun. 2015.
- [2] 牧野義樹, 丹康雄, 金島正治, "都市・地域対応スマートコミュニティシミュレータの研究開発", 2015 年度日本建築学会大会 (関東) 大会梗概集, D-1 分冊, p.805, Sep. 2015.
- [3] MATLAB version 9.2.0. Natick, Massachusetts: The MathWorks Inc., 2017.
- [4] Miyachi, Toshiyuki, Ken-ichi Chinen, and Yoichi Shinoda. "StarBED and SpringOS: large-scale general purpose network testbed and supporting software." Proceedings of the 1st international conference on Performance evaluation methodologies and tools. ACM, 2006.
- [5] Miyachi, Toshiyuki, et al. "StarBED and SpringOS architectures and their performance." International Conference on Testbeds and Research Infrastructures. Springer, Berlin, Heidelberg, 2011.
- [6] Kennedy, James. "Particle swarm optimization." Encyclopedia of machine learning. Springer US, 2011. 760-766.