Multi-Dimensional Energy Consumption Scheduling for Event Based Demand Response

Rohit Singh Rana

A thesis submitted in partial fulfillment of the requirements for the
Master's in Applied Science degree in Electrical and Computer Engineering

School of Electrical and Computer Engineering
Faculty of Engineering
University of Ottawa

Abstract

The global energy demand in residential sector is increasing steadily every year due to advancement in technologies. The present electricity grid is designed to support peak demand rather than Peak to Average (PAR) demand. Utilities are investigating the residential Demand Response (DR) to lower the (PAR) ratio and eliminate the need of building new power infrastructure. This requires Home Energy Management System (HEMS) at grid edge to manage and control the energy demand. In this thesis, we presented an MDPSO based DR enabled HEMS model for optimal allocation of energy resources in a smart dwelling. The algorithm is designed to lower peak energy demand as well as encourage the active participation of customers by offering a reward to comply with DR request. We categorized appliances as elastic non-deferrable loads and inelastic deferrable loads based on their DR potential and operating characteristics. The scheduling of elastic and inelastic class of appliances is performed separately using canonical and binary version of PSO given how we expressed out load categories. We performed use case simulation to validate the performance of MDPSO for combination of different tariffs: Time of Use (TOU), TOU and Critical peak rebate signal (CPR), TOU and upper demand limit. Simulation results show that algorithm can reduce the electricity cost in range of 28% to 7% under increasing comfort conditions in response to TOU prices and Peak demand reduction of about 24% under TOU pricing and medium comfort conditions for single household. Under CPR DR requests, with respect to TOU pricing, there is effectively no change in the peak under the minimum comfort scenario. Furthermore, algorithm is able to suppress the peak upto 25% under combination of TOU and hard constraint on maximum power withdrawn from grid with no change in the electricity cost. Scheduling of multiple houses under TOU pricing results in peak reduction of 7 % as compared to baseline state. Under combination of TOU and CPR the aggregate peak energy demand of multiple households during DR activation time intervals is reduced by 32 %. The algorithm can suppress the peak demand by 27% under TOU and hard constraint on maximum power withdrawn from grid by multiple houses.

Acknowledgements

Firstly, I would like to express sincere gratitude to my advisor Prof. Henry Schriemer for the continuous support and motivation in completing my thesis. His guidance helped me throughout the time of my research and writing of thesis. My sincere thanks to Dr. Javad Fattahi for his insightful comments, encouragement and hard questions which incented me to widen my research from various perspectives. Also, I would like to extend thanks to my fellow SAGE lab mates, especially my best friend Robert Pollak for stimulating discussions, sleepless nights before deadlines and all the fun we have had together.

Last but not the least, I would like to thank my family: my parents, my brother (Shamsher Singh) and fiancé (Anisha Nagpal) for supporting me throughout my thesis and my life in general.

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

1.1. Motivation

The changing nature of electricity use and generation is creating problems like blackouts, voltage instability, frequency, environment pollution, deployment of new power infrastructure etc. Smart grid envisions to solve these problems by engaging demand side of the equation and encouraging the consumers to play an active role in managing the complex power system. This led to the development of residential demand response (DR) programs to manage behind the meter resources by utilizing their inherent flexibility. Home Energy Management System (HEMS) is required at grid edge to implement the DR programs. HEMS manages the loads in smart houses by utilizing optimization algorithms for scheduling the resources to minimize the objectives such as electricity cost, consumer discomfort or emission of green house gases. The primary motive of our work is to advance a technique to better investigate the capability of DR enabled HEMS. This requires formulating an algorithm to optimize the energy consumption of dwelling under day ahead time varying electricity prices or demand response (DR) event while taking into considerations the utility requirements and consumer comfort. A significant research effort had been made to investigate optimization algorithms including mathematical optimization techniques and heuristic algorithms for solving energy consumption scheduling problem. Most of the approaches consider hierarchal optimization approach to schedule appliances which may result in sub-optimal solution. In our work, we proposed a multi-dimensional Particle swarm optimization (PSO) based approach for simultaneous scheduling of appliances to minimize a commonly shared fitness function. Particle swarm optimization is selected for HEMS modelling due to its proven ability to perform efficiently over high dimensional search spaces, easy implementation, inbuilt flexibility to lower peak demand and low computational complexity.

1.2. Thesis Contribution

The main contributions of our work include:

1. We investigated the demand response potential of appliances in a smart home and categorized them based on their elasticity in energy and time. HEMS model is formulated as a multi-objective optimization problem with an aim to find optimal tradeoff between cost and comfort. The first objective is to minimize electricity cost based on TOU or combination of TOU and CPR signal and objective is to minimize the user discomfort. The different categories of appliances are modelled as constraints for the optimization problem.

2.We proposed a multi-dimensional PSO based HEMS model for optimally scheduling the energy consumption of appliances. We considered scheduling multiple appliances simultaneously rather than taking a hierarchal optimization approach of scheduling each appliance individually.

3.We evaluated the performance and efficiency of proposed algorithm under different pricing signal: TOU, combination of TOU and CPR and combination of TOU and hard limit on maximum energy consumption throughout the scheduling time range.

4.Our proposed solution minimizes the participating user's electricity cost as well as reduces PAR for utility under peak periods.

1.3. Thesis Organization

This thesis is organized as follows:

Chapter 2 reviews the literature pertaining to the evolution of the traditional electricity grid in terms of power and communication interfaces. Furthermore, we reviewed the pros and cons of various Residential DR programs implemented by utilities. In context of this thesis, a detailed description is presented on the modelling complexities and considerations for developing an efficient and reliable HEMS. In addition, the various optimization techniques in the HEMS literature used to solve the energy consumption scheduling problems are presented.

Chapter 3 reviews the classical version of global best PSO and discrete PSO. We discussed the techniques used in PSO literature to enhance the performance of PSO by proper selection and tuning of various parameters. Furthermore, the constrained PSO for handling various equality and inequality constraints is discussed. The MDPSO for solving multi-dimensional knapsack combinatorial problems is presented.

Chapter 4 presents the higher-level overview of the system model in this study. A detailed discussion on modelling different categories of loads and problem formulation is done. Load reshaping strategies under mandatory and non-mandatory DR event are discussed. The energy consumption scheduling using proposed MDPSO is presented.

Chapter 5 presents the use case studies to validate the problem formulation and proposed algorithm. The sensitivity analysis of algorithm is done.

Chapter 6 presents the simulations under different comfort conditions and pricing signals for single and multiple dwellings. We discussed the cost and PAR reductions using proposed algorithm.

Chapter 7 concludes the results and presents future research direction.

2. LITERATURE REVIEW

2.1. Evolution of the Electricity grid

The electricity grid has continuously evolved since the second industrial revolution [1]. Initially, the power grid was isolated and decentralized having self-sufficient energy production and consumption. It was soon recognized that the interconnection of power systems could significantly improve the energy costs and system reliability. However, the centralized large-scale power production also brought serious problems like dependency on fossil fuels, underused systems and increased carbon emission levels [2]. In order to solve the increasing environmental concerns, a greater priority was given to build renewable energy resources; e.g. solar power and wind power. This is leading to the transformation of the centralized power system into a distributed power system, resulting in emergence of new challenges such as scaling up the renewable generation and transmission technologies. The rapid development in information and communication technologies is further elevating the electricity grid to the future grid termed as Smart grid, which is more smart, secure, economical and provides sustainable supply of energy.

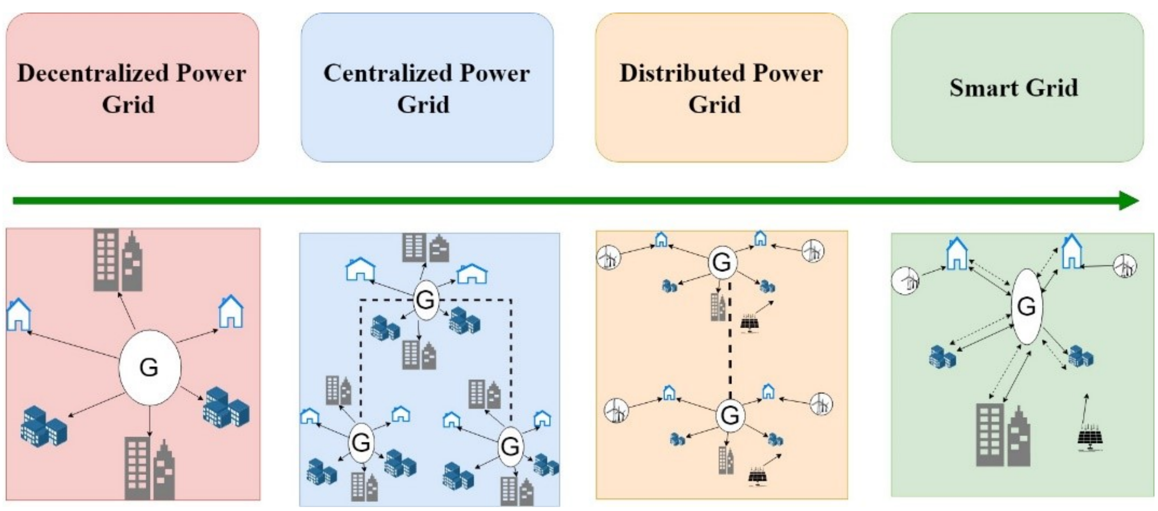

Figure 2.1 Evolution of Electricity Grid

2.1.1. Smart Grid

National Institute of standards (NIST) Framework and Roadmap for Smart Grid Interoperability Standard defines smart grid as “a modernized grid that enables bidirectional flows of energy and uses two-way communication and control capabilities that will lead to an array of new functionalities and applications" [3]. Smart grid offers a wide range of benefits like self healing capability, user-centric, accommodation of diverse distributed energy resources, optimized asset operations and interoperability. In recent years, there has been increasing interest in development of reference smart grid models to build new generation power grids. To provide a better understanding of smart grid stake holders and feasible electrical and communication interfaces, a smart grid conceptual model is developed by National Institute of Standards NIST[3] as shown in Figure 2.2.

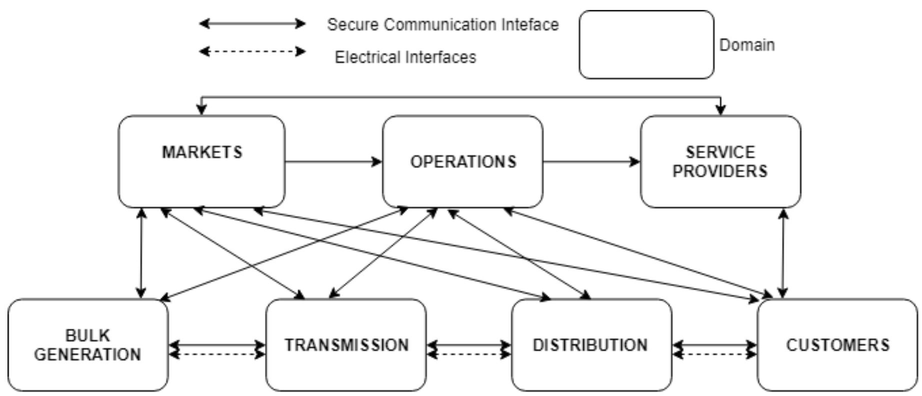

Figure 2.2 Smart Grid Conceptual Model[3]

The standard presents a smart grid reference model consisting of seven functional blocks: customers (commercial or residential), Markets, service providers, operations, bulk generation, transmission, and distribution. The bulk generation domain consists of base load energy generation units that supports the variability of renewable resources. The transmission and distribution domains supply the produced energy to long distances, monitor and control systems, and are controlled by the regional load dispatch centres or operation domain. The energy balance is maintained over the long term by the market domain, which acts as an intermediator between bulk generation units, retailers and transmission and distribution operators. The role of service providers is to provide service to electrical consumers and utilities such billing and financial management. The customer domain includes the consumers or prosumers which are mainly end users of electricity and categorized as residential, commercial and industrial.

2.1.2. **Communication Infrastructure**

The development of modern information and communication technologies (ICT) is expected to play an important role in improving the reliability and efficiency of power grid. Smart grid functionality is achieved by overlaying communication infrastructure with an electrical system infrastructure. The key challenge is to build an open system smart grid architecture having commonly shared communication standards and protocols to operate several systems and sub-systems. It can be hard to build a single and composite smart grid architecture; hence system architectures can be developed independently or in association with other systems. Figure 2.3 presents the hierarchal overview of smart grid communication infrastructure consisting of different members and technologies as per NIST domains interacting with each other. Our research is focused on designing a demand side management technique at Home Area Network (HAN) which is a sub-system in the hierarchical chain of smart grid. HAN provides demand response capabilities at grid edge level by active communication between consumers and load serving entities. HAN represents the smart devices with sensors and actuators, smart meters and Home energy management system (HEMS). HEMS helps to manage and control the energy consumption of dwelling. Smart devices in HAN can communicate with each other using both wireline technologies such as power-line communication (PLC), BACnet protocol, and wireless technologies (e.g. Zigbee and Wi-Fi).

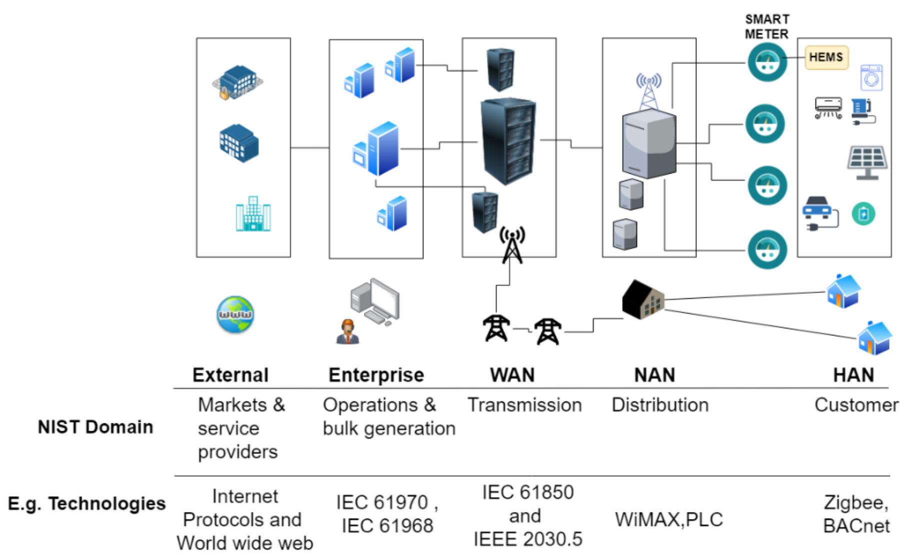

Figure 2.3 Technologies and Different members enabling Smart Grid[3]

2.2. Demand Side Management

The power grid is usually designed for peak demand rather then average demand which results in underutilization of power system infrastructure. This problem can be solved by reshaping the demand to lower the peak to average ratio and to more effectively use any intermittent energy sources. However, increasing expectations of prosumers in terms of power quality and quantity, the emergence of new types of demand such as PHEV (Plug-in hybrid electric vehicles), which can double the average household energy consumption [4] and customer privacy concerns gives rise to need of developing advanced DSM mechanisms.

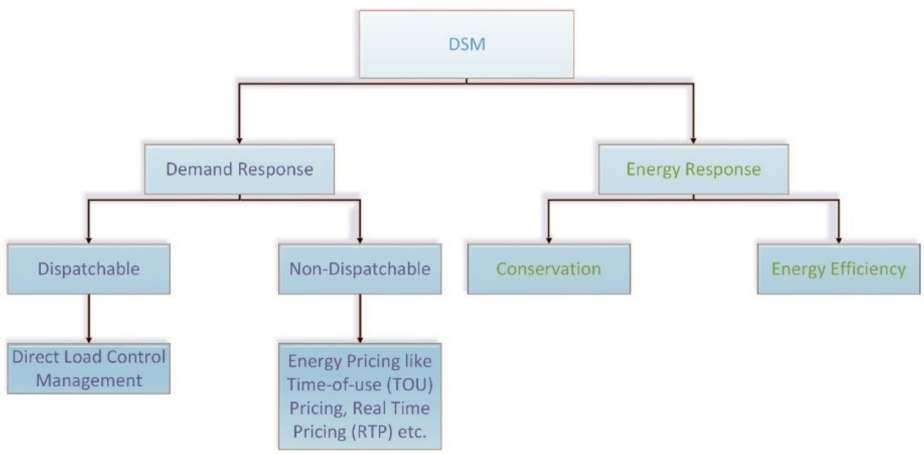

Figure 2.4 DSM Categories[5]

Various DSM techniques based on peak clipping, strategic conservation, valley filling, load shifting, strategic load growth and flexible load shaping [6] have been implemented by utilities to encourage the participation of consumers at grid edge. The two fundamental DSM techniques employed are: Energy Efficiency and Demand Response, which are further sub-categorized as shown in Figure 2.4 The former refers to the techniques employed for conservation of energy by using efficient energy solutions like lighting retrofits, HVAC improvements resulting in overall decrease of energy consumption. Demand response is a term that refers to the programs designed by utilities to encourage end-users to make short time reduction in energy demand in response to different pricing signals or DR request from utility usually resulting in short time reduction/curtailment of power ranging from 1 to 4 hours. As shown in Figure 2.5, under baseline scenario, a peak power is observed between 5:00 to 8:00 hours and DR event raised during peak hours suppresses the peak demand. However, a peak rebound effect is observed during off peak hours due to scheduling of appliances to low price periods which may cause overloading at distribution level. The detailed description of different types of residential demand response programs is discussed briefly in following section.

2.3. Residential Demand Response Programs

Residential sector alone contributes to significant amount of the total energy consumption worldwide and is expected to play a major role in smart grid framework. Hence, utilities are investigating the advanced residential DR programs at the grid edge to engage the end-users to lower the peak to average ratio. Demand response (DR) is becoming increasingly common within the electricity sector and is defined by U.S. Federal Energy Regulatory Commission as "Changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized" [7].

Different DR programs can be broadly classified as Direct load control and Indirect load control based on interaction between end-users and utilities.

2.3.1. **Direct Load Control (DLC)**

Incentive based DLC programs are based on agreements between the consumer and utility. Utilities can remotely control the smart appliances at the grid edge to minimize the peak load demand. For example, utilities can control the operating cycle of HVAC by adjusting the room temperature set point through a smart thermostat. The customers can volunteer or participate in DR events based on a pre-agreed incentive. The authors in [8]–[11] proposed various DLC mechanisms designed to reshape the aggregate household load profiles. DLC programs are further sub-classified into interruptible load programs and capacity-based programs. In interruptible load programs, participating consumers are subjected to temporary load interruption whereas in capacity-based programs users commit to reduce by pre-specified energy levels during demand response events. DLC programs are considered highly effective in reducing PAR [7], but may lead to serious concerns such user privacy, discomfort due to mandatory interruptions, which may discouraging them to participate in demand response events.

2.3.2. **Indirect Load Control / Price Based Programs**

Price based demand response programs are designed to encourage the consumer to shift/curtail the energy consumption in response to various smart pricing signals such as Time of Use Pricing (TOU), Critical Peak Rebate (CPR) and Real Time Pricing (RTP). Indirect load control mechanisms are widely deployed due to their potential to reduce peak

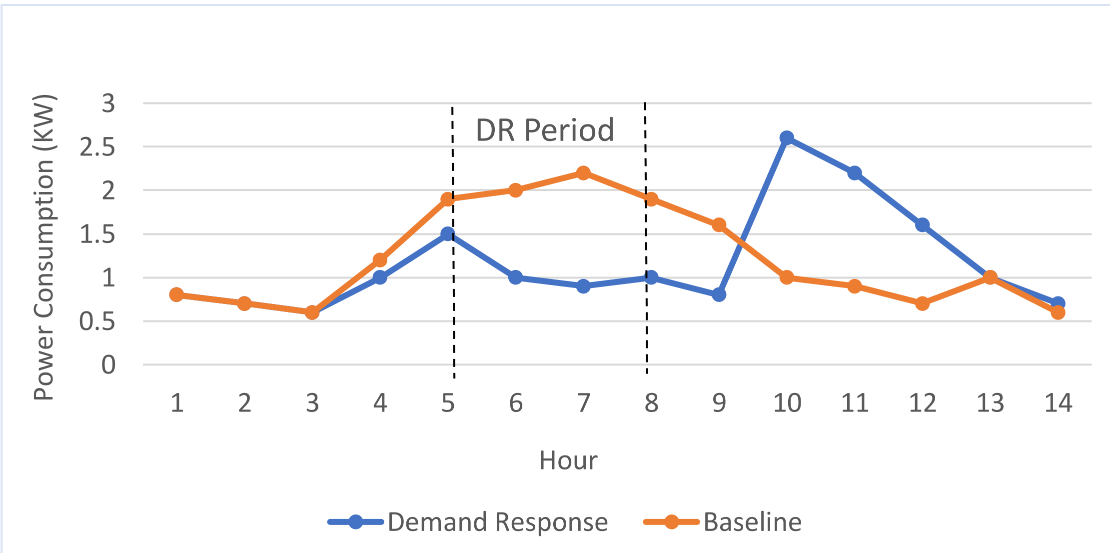

Figure 2.5 Typical Demand Response (DR) effect on household power consumption, with subsequent rebound effect

demand, decrease electricity cost for consumers while preserving user privacy provides customer and users ability to control its comfort. Farooqui *et.al.* [12] reviewed 15 residential time varying DR programs and noticed a 3% to 6% reduction in peak demand. Furthermore, CPR with advanced automated technologies enabled DR programs achieved a peak load reduction of 51% in California's Advanced DR Program [12]. However, these categories of programs require minimum consumer interaction i.e. eliminate the need of customer manually responding to time varying prices, hence advanced automated technologies are key to the success of these programs. Furthermore, with TOU based DR programs users can decide their optimal energy consumption pattern, so it is quite possible that undesirable peak rebounding effect may occur [13].

2.4. Home Energy Management System (HEMS)

HEMS are residential demand response tools to shift or curtail the energy consumption of dwelling in response to time varying prices or load shedding request from the utility to the consumer. A recent enabling trend impacting smart HEMS solutions in the residential sector is the increased customer adoption of smart technologies such as smart thermostats. Lobaccaro *et al.* [14] reviewed the different smart home technologies with their potential benefits, advantages and disadvantages and future trends. The capabilities of currently available energy management solutions are quite limited such as controlling a limited number of smart appliances in a dwelling. However, new advanced intelligence and sophisticated energy management solutions are penetrating the electricity market and hence paving a way to implement advanced DR programs. These advanced HEMS includes intelligent algorithm and optimization engines to schedule the home energy consumption maintaining acceptable user comfort levels and convenience. This thesis is focused on designing an event driven algorithm to optimally scheduled the home energy consumption. In the context of our work, we will review the challenges of modelling HEMS, optimization techniques used to manage the distributed energy resources at grid edge, and challenges in selecting the algorithm for optimizing household energy consumption.

2.4.1. Modelling considerations and complexity

The primary goal of HEMS is to schedule the energy consumption profile with minimum user intervention at the HAN level and coordinate with the utility to lower the peak to average ratio at the distribution level. The energy consumption scheduling problem is usually formulated as an optimization problem with objectives such as minimizing cost, discomfort, carbon emissions and PAR, subject to constraints enforced by the operating characteristics of smart appliances, customer behaviour and demand response event from utilities. However, it is not an easy task as factors such as forecasting uncertainty, customer unique behaviour and multitude of appliances increases the modelling complexity. Most of the present approaches consider user preferences, such as appliance operating periods, and indoor temperature bounds, to reduce modelling complexity. In order to further simplify the modelling complexity, each appliance is typically scheduled in a hierarchal way based on day ahead or real time pricing signal from utility. However, such modelling simplification may lead to a sub-optimal solution as there is a trade off between optimality and complexity. We will discuss the different mechanisms noted in the literature to handle modelling complexity and design an effective, practical and user aware energy management solution.

2.4.1.1. Heterogeneity of Smart Devices

Modelling a multitude of smart appliances is a challenge in the deployment of HEMS, since each smart device in a dwelling has separate operating characteristics and demand response potential. Most of the present approaches in the literature consider modelling each device independently for the HEMS, developing physical models for each smart device, hence increasing the complexity. Many works have tried reducing modelling complexity by grouping appliances with common operating characteristics and demand response potential into unique response classes. Literature has not yet converged on how to best categorize the appliances in a dwelling. For example, a few models consider the refrigerator as an uncontrollable load [15], [16],while others consider it as a time-deferrable load[17], [18], or a regulating energy shiftable load [18][19]. In addition, appliances such as space heaters, electric ovens, cloths dryers, water heaters are considered in three different classes. This complexity in accurately defining each response class is mainly because of differences in the perceived appliances operating characteristics over varying time intervals and resolution. Other possible explanation for disagreement may be due to perception in terms of consumer inconvenience, when considering their DR potential.

2.4.1.2. Multi-Objectivity

The appliance energy consumption scheduling problem is usually formulated as a multi-objective optimization problem as it considers trade off between multiple objectives. Multi-objective energy consumption scheduling requires finding a non-dominated pareto-front solution that satisfies the multiple objectives. Various techniques used frequently in HEMS literature for solving the multi-objective optimization problem include bounded objective [21] ,pareto-fronts [22] and weighted sum approach [23].Bounded objective approach sum approach transforms one objective into constraint with in a acceptable range. This requires prior knowledge about the feasible value of objective function. Pareto-fronts technique allows the decision maker to select the optimum solution from a set of possible solutions for solving multi-objective optimizations. Weighted sum approach is a widely used method for HEMS application due to its straightforward implementation. This approach considers the sum of all objectives after multiplying all objectives with a scalar weight. The value of scalar weight can be defined prior by the decision maker for HEMS application and hence acts as knob to play between different objectives. We took weighted sum approach.

2.4.1.3. Forecast Uncertainty

One of the major complexities in modelling HEMS is forecast uncertainty. Forecast uncertainty in HEMS literature pertains to starting time or finishing time of appliances, weather, insolation etc. The issue of forecast uncertainty can be addressed by avoidance, making decisions based only on past and present data in real time. Another method is to use point forecasts for decision making. For example, the game theoretic model in [15] removes uncertainty in prices by iteratively deciding prices and load quantity based on game between utility and customer. Although, these approaches are considered efficient for solving optimization problem at lower computational costs but incorporating uncertainty into model improves the scheduling process efficiency further. Optimization under uncertainty can be handled using optimization approaches such as deterministic Optimization and Stochastic optimization. Deterministic Optimization approaches considers uncertain data as a set of values representing user preferences such as appliance operating duration and thermostat set point temperature. Therefore, we took deterministic approach because it is practical for HEMS implementation as the complexity is reduced by limiting the uncertain parameters under certain cases. In contrast to deterministic approaches, stochastic optimization considers uncertainty as a probabilistic model. The deterministic optimization techniques such as linear integer programming, heuristic algorithms are considered practical for implementing in HEMS since stochastic optimization causes increase in complexity due to uncertainty and exponential increase in number of variables to be optimized.

2.4.1.4. Problem Dimensionality Reduction

HEMS schedule energy consumption of a dwelling over a certain period in a day, called the scheduling horizon T. The scheduling horizon is discretized into t number of equal time intervals. Hence, each time interval's (t_s) resolution and length of scheduling horizon determines the size of scheduling problem. In addition, increase in the number of devices A to optimize also increases the problem size. The size of problem grows linearly with increase in the temporal parameters T, t_s & with the number of devices A. This is because of increase in the number of variables associated with each time interval to be optimized. Figure 2.6 illustrates the growth of problem size with respect to number of time intervals and devices A.

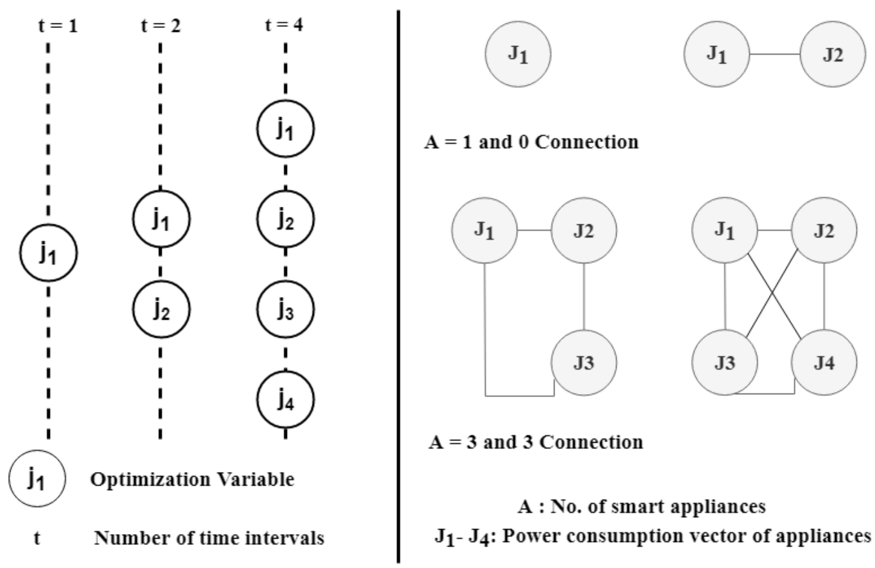

Figure 2.6 Growth in problem size with t and A

It is important to size and model the problem appropriately to balance complexity and optimality of HEMS. We will now discuss the impact of reducing the problem size and modelling complexity on the HEMS solution. Frequently, HEMS modelling techniques use a 24-hour scheduling horizon divided into equal 1-hour time intervals. However, the use of a 1-hour time interval does not capture the intermittency and variability in demand and hence results in sub-optimal or infeasible solutions. As an example, in [24], the peak power consumption using a 10-minute time interval is 6 times higher than 1-hour time interval. Furthermore, the authors of [25] suggested that the reduction in scheduling horizon up to 13-hour horizon may result in myopic optimization. Myopic optimization refers to improving the objective function by assigning a new value instead of given partial solution. In addition to this, most of the present approaches minimize the problem size by optimizing the appliances by sequential scheduling, which may force a peak in consumption at the end of the scheduling period.

In our proposed HEMS model, we used the length of scheduling horizon as 24 hours, time interval windows as 15 minutes and co-ordination among appliances, hence covering the demand variability and preventing myopic optimization. It should be noted that more complex HEMS models may result in increased computation costs as the number of variables to optimize at specific time intervals increases.

2.4.2. **Optimization Techniques**

There exists a wide HEMS literature using different optimization algorithms to manage and control the energy consumption of a dwelling. The various approaches proposed for scheduling household energy consumption can be mainly categorized as mathematical optimization, meta-heuristic algorithms (*e.g.,* evolutionary algorithms), and negotiation-based (*e.g.,* game theoretic) approaches. The selection of optimization algorithm mainly depends on the nature of energy consumption scheduling problem and overall system architecture. In this section, we will review the various approaches used for efficient management of household energy consumption.

2.4.2.1. Mathematical Optimization

Appliance scheduling can use either deterministic or stochastic optimization approaches. Focusing on the former, the input values are selected systematically by the user, or generated using learning algorithms, and an optimal energy consumption schedule is generated. The mathematical optimization techniques widely used in HEMS literature include:

Linear Programming (LP) is the simplest mathematical optimization technique for solving linear objective functions, subject to linear equality and inequality constraints. For example, Lee and Choi [26] formulated the power consumption scheduling at the HAN level using linear programming with an objective to shave hourly peak load via charging and discharging energy storage systems, during off peak hours and on peak hours, respectively. Although this technique is considered the simplest form of mathematical optimization and is solvable in polynomial time, it may lead to sub-optimal or infeasible operations.

Integer linear Programming **(ILP)** problems are similar to linear programming, but the only difference is that the decision variable is an integer value. These types of problems are NP hard in many practical scenarios. Binary Integer linear programming (BILP) is a special class of integer programming where the decision variables are required to be binary, that is, either 0 or 1. If some of the decision variables to be optimized are integer while others are non-integer, then the problem is termed Mixed Integer Linear Programming (MILP). A wide literature is focused on formulating the HEMS problem as ILP, BILP or MILP and solving them using algorithms such as branch and bound algorithm, simplex algorithm and cutting-plane method. For example, Zhu *et al.* [16] formulated a power consumption scheduling mechanism with an integer linear programming objective to minimize peak load for single and multiple dwellings. The authors in [27] presented the appliance scheduling problem as an mixed integer linear programming consisting of two objectives: minimize both cost and peak load. They used a Branch and bound algorithm to find the optimal scheduling interval of appliances. Paterakis *et al.* [28] used a mixed integer linear programming model for HEMS optimization of load and generation assets under different pricing and peak power-limiting-based DR strategies.

Quadratic Programming problems require the objective function and constraints to be quadratic. These problems can be solved in polynomial time only if the objective function is positive definite. Saghezchi *et al.* [29] proposed an DSM approach based on quadratic programming. The quadratic objective function was defined as the sum of squares of hourly energy consumption of both shiftable and non-shiftable appliances of consumer. It was concluded that Quadratic programming is an efficient approach to minimize PAR when compared to MILP approaches.

Dynamic programming problems make the problem scalable by breaking the larger complex problem into smaller sub-problems and solving them in hierarchal order by storing sub-problem optimal solutions in memory. Jeddi *et al.* [30] presented a dynamic programming based HEMS solution for residential customers. The optimization was performed to minimize the consumer's electricity cost by efficient scheduling and management of PV and Energy storage systems. As dynamic programming requires storing the results intermediately for sequential processing, they can be memory intensive.

Non-Linear programming **(NLP)** problems consist of non-linear objective functions or constraints. Most of the real-world problems are non-linear and very difficult to solve, hence may lead to sub-optimal solution. Roh and Lee [31] categorized household appliances into sets, depending upon their energy consumption and operating characteristics, and developed a load scheduling algorithm that incorporates TOU pricing under budget limits. Optimization was formulated as a mixed integer nonlinear programming (MINLP) problem. In order to solve the optimization problem with low computational cost, they used a generalized blenders decomposition (GBD) approach. In addition, the proposed algorithm provided more flexibility to control the trade-off between utility function and electricity cost.

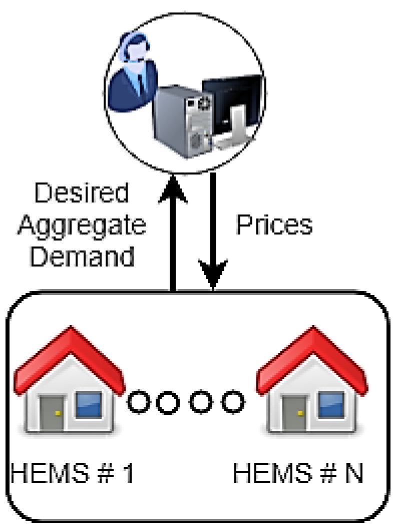

Figure 2.8 Example of Non-negotiation-based

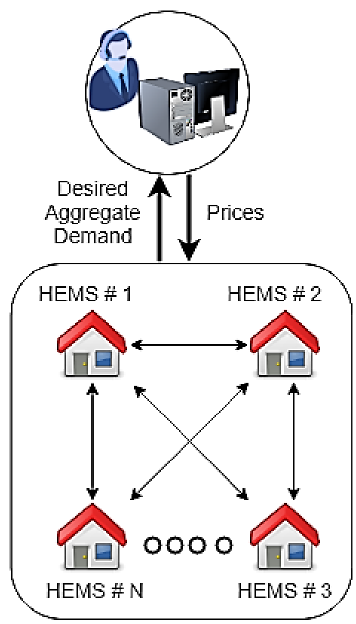

Figure 2.7 Example of negotiation-based approach.

2.4.2.1. Negotiation Approaches

Most of the DR programs consider interaction between aggregator and HEMS at the grid edge. For example, under TOU-based DR programs, each consumer is expected to shift the energy consumption from peak price periods to low price periods, and under this scenario each HEMS communicates with the aggregator as depicted in Figure 2.7(a). However, better DR can be achieved by enabling coordination among HEMS controllers, as illustrated in Figure 2.7(b). Aggregators can provide incentives to each consumer to coordinate their energy usage. Mohsenian-Rad *et al.* [15] proposed a optimal, autonomous and distributed game theoretic approach for scheduling the energy consumption of dwelling considering such neighbourhood coordination. The optimization problem was formulated to minimize the energy cost for a group of users at the grid edge. The players are the end users and their strategy is to find the minimum electricity cost and PAR. The Nash equilibrium of the energy consumption game among the participating users is the optimal operating point of the system. Saad *et al.* [32] presented a survey of game theoretic approaches for smart grid application, mainly focused on three aspects: micro-grid systems, demand-side management, and communications. They presented future research options for adopting game theory in smart grids.

2.4.2.2. Evolutionary Algorithms

Evolutionary algorithms (EA) are metaheuristics that use large populations that travel within a bounded search space to find the optimal solution without making presumptions about the fitness landscape. Evolutionary algorithms such as Genetic Algorithm (GA), Wind Driven Optimization (WDO), and Particle Swarm Optimization (PSO), are widely used in HEMS literature for solving the residential energy consumption scheduling problem. Genetic algorithms (GA) rely on biomimetic operators such as mutation, crossover and selection to generate solutions to optimization and search problems. Zhao *et al.* [33] used GA to minimize the electricity cost, combining real time pricing (RTP) with the inclining block rate (IBR) model, and including an appliance delay time rate as a customer comfort proxy. They found both PAR and cost reductions under comfort constraints. WDO is a nature-inspired global optimization algorithm based on the atmospheric motion of wind particles. Unlike GA and PSO, its constraints can be implemented in search domains. Rasheed *et al.* [34] used WDO with the objective function of comfort maximization along with minimum electricity cost in an efficient power scheduling scheme. PSO a non-gradient-based computational method that iteratively evolves a candidate solution. It is widely used for consumption scheduling due to its simple implementation, ability to find the optimal solution over high dimensional search spaces, and ability to often outperform mathematical optimization techniques. Laskari *et al.* [35] compared PSO with the widely-used branch and bound algorithm to solve Integer programming benchmark test problems. They concluded that PSO performs better for high dimensional search spaces, exhibiting high success rates even in the cases where BB technique failed. In addition to this, Hassan *et al.* [36] performed a comparative study of GA and PSO in terms quality of solution and efficiency. Algorithms were tested for 12 different benchmark functions and results showed that the efficiency of PSO is better then other EA.

This thesis is focused on developing an enhanced PSO model for HEMS benchmarking, so we will also review the literature applying classical PSO and its variants. Huang *et al.* [37] proposed gradient-based PSO for optimal scheduling of household appliances to minimize the electricity cost based on TOU pricing. Zhou *et al.* [38] proposed a binary PSO based real time optimal usage strategy considering TOU pricing and an energy threshold, where a feedback signal was generated to shift or curtail the power demand. Yang *et al.* [39] considered DR using PSO based on time-of-use (TOU) and critical peak rebate (CPR) signals, finding that load shifting was achieved with minimum user discomfort. In our work, we have taken a multi-dimensional PSO approach to optimally distribute the energy consumption of dwelling based on TOU pricing incentive or CPR signal or mandatory DR request to elevate the system stress. In the next chapter, we detail the PSO fundamentals and build upon them, describing our enhancements.

2.4.3. Approaches to Scheduling with PSO

3. PSO & ITS ENHANCEMENTS

This chapter first provides the reader a conceptual overview of classical global best continuous and binary PSO. The working of PSO for solving unconstrained and constrained optimization is then discussed briefly. Furthermore, a detailed overview of optimized parameters selection for tuning the performance of PSO is provided. Finally, the working of our enhanced PSO for solving multi-dimensional combinatorial knapsack problems is presented.

3.1. Optimization Algorithm

3.1.1. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based stochastic optimization method developed in 1995 by Kennedy and Eberhart [40]. It is a biomimetic gradient free optimization engine exhibiting behavior similar to that of a group of birds searching for food. Like most evolutionary techniques, PSO uses a random initial population and searches for the optimal solution by updating the generations. In analogy to evolutionary algorithms (EA), a swarm is like a population and a particle is analogous to an individual. In brief, PSO converts the decision vector to be optimized to a scalar value by computing the fitness function and then uses the value of the fitness function to find the best solution vector from population.

Since its introduction, many improvements have been made to increase the accuracy, convergence capability and diversity of PSO. This led to the development of various versions of PSO, which are basically derived from two basic types of PSO algorithms: the *global best* and the *local best*. Due to low computational cost and proven ability to converge to global best solution, the gbest algorithm is widely applied to solve both linear and non-linear optimization problems.

3.1.2. Global Best PSO Model

Gbest PSO is the canonical version of PSO and is popular because of its simplicity in implementation. It maintains a randomly initialized group of particles called a swarm, where each individual particle represents a candidate solution. At every iteration, each particle in the swarm flies through a multi-dimensional search space. The position of each particle is adjusted based on its own best experience and social interaction with neighborhood particles. To illustrate the working of Gbest PSO, let us assume a general unconstrained optimization problem defined as

$$minimize\ f(\boldsymbol{x}) = \sum_{i=1}^{N} x_i^2 , \qquad \boldsymbol{x} = (x_1 , \dots \dots , x_N) \tag{3.1}$$

where a decision vector $\boldsymbol{x}$ of length N is optimized to find the minimum value of the fitness function $f(\boldsymbol{x})$,which is a quadratic function having global best solution at origin. In order to optimize the objective function (3.1), let us consider a swarm S of size M. The position vector of each particle $j \in M$ at iteration t is represented as $\boldsymbol{x}_j(t)$. Each position vector represents a possible solution to the optimization problem at hand. Furthermore, each particle has an associated velocity vector $\boldsymbol{v}_j(t)$, which drives the optimization process. Each element of the velocity vector is clamped between v_{min} and v_{max} values to prevent the likelihood of particles leaving the search space. The velocity of particle j is computed based on the following, which will be more fully described below:

a.) *The previous velocity* represents the memory of a particle's previous flight direction. This term act as a momentum, which prevents the particles from drastically changing their flight direction. The particle's previous velocity is weighted by an inertia component $w(t)$, which acts a knob to control the influence of a particle's previous velocity on its new velocity.

b.) *The cognitive component reflects* the experiential knowledge of particle j. It quantifies the performance of particles relative to previous performances. Let $\boldsymbol{p}_j^l(t)$ refer to the *local* best position achieved by particle j of swarm S since the first iteration. The cognitive component is the distance of $\boldsymbol{p}_j^l(t)$ from current position $\boldsymbol{x}_j(t)$. It is stochastically weighted by $\boldsymbol{r}_{1,j}(t)$, a random vector sampled from a uniform distribution and scaled by a positive acceleration coefficient c_1.

c.) *The social component* reflects the socially exchanged information among the particles. It is the social component that attracts each particle towards the best position achieved by neighborhood particles. In other words, it quantifies the performance of particle j relative to the swarm S . Let $\boldsymbol{p}_S^g(t)$ refers to the *global* best position achieved swarm S since the first iteration. The social component is distance between $\boldsymbol{p}_S^g(t)$ and current position $\boldsymbol{x}_j(t)$. It is stochastically weighted by $\boldsymbol{r}_{2,j}(t)$, a random vector sampled from a uniform distribution and scaled by a positive acceleration coefficient c_2.

The following steps describe a Gbest PSO algorithm to solve (3.1).

Step1: Initialization

a.) Randomly initialize $\boldsymbol{x}_j(t), \boldsymbol{v}_j(t),\ \boldsymbol{p}_j^l(t)$ for each j in swarm S and $\boldsymbol{p}_S^g(t)$.

b.) Set the value of constants w, v_{max}, v_{min}, c_1 and c_2 as per section 3.1.3.

c.) Set iteration count $t = 0$.

Step 2: Optimization

a.) Evaluate iteratively (over t) the value of the fitness function for each particle j of swarm S as follows

$$f\left(\boldsymbol{x}_j(t)\right) = \sum_{i=1}^{N} x_i^2(t) \tag{3.2}$$

b.) Update the $pbest$ position of each particle j by comparing the current value of the $pbest$ cost (fitness function) with the best value of the fitness function achieved by particle j since its first iteration, as

$$\boldsymbol{p}_j^l(t+1) = \begin{cases} \boldsymbol{p}_j^l(t) & \text{if } f\left(\boldsymbol{x}_j(t+1)\right) \geq f\left(\boldsymbol{p}_j^l(t)\right) \\ \boldsymbol{x}_j(t+1) & \text{if } f\left(\boldsymbol{x}_j(t+1)\right) < f\left(\boldsymbol{p}_j^l(t)\right) \end{cases} \tag{3.3}$$

Where $f\left(\boldsymbol{p}_j^l(t)\right)$ is a scalar value representing $pbest$ cost of particle j at iteration t.

c.) Update the global best position, $\boldsymbol{p}_S^g(t)$ at each iteration using

$$f\left(\boldsymbol{p}_S^g(t)\right) = \min\left\{f\left(\boldsymbol{p}_1^l(t)\right), f\left(\boldsymbol{p}_2^l(t)\right) \dots, f\left(\boldsymbol{p}_j^l(t)\right)\right\}, \tag{3.4}$$

where $f\left(\boldsymbol{p}_S^g(t)\right)$ is the $gbest$ cost of swarm S at iteration t.

d.) Update the velocity $\boldsymbol{v}_j(t)$ of each particle n at iteration t using the velocity update equation

$$\boldsymbol{v}_j(t+1) = w(t)\,\boldsymbol{v}_j(t) + \overbrace{c_1\boldsymbol{r}_{1,j}(t)\left[\boldsymbol{p}_j^l(t) - \boldsymbol{x}_j(t)\right]}^{\text{cognitive component}} + \overbrace{c_2\boldsymbol{r}_{2,j}(t)\left[\boldsymbol{p}_S^g(t) - \boldsymbol{x}_j(t)\right]}^{\text{social component}} \tag{3.5}$$

where $w(t)$ is the inertia weight; c_1, c_2 are positive acceleration constants used to weight the social and cognitive components, respectively; and $\boldsymbol{r}_{1,j}(t), \boldsymbol{r}_{2,j}(\boldsymbol{t})$ are random vectors sampled from a uniform distribution, in the range (0, 1), that introduce stochastic character to the algorithm.

e.) Check that the updated velocities are within the defined limits v_{max} as

$$v_j^k(t+1) = \begin{cases} v_j^k(t+1) & if\ v_j^k(t+1) < v_{max}^j \\ v_{max}^j & if\ v_j^k(t+1) \geq v_{max}^j \end{cases} \tag{3.6}$$

where $v_j^k(t+1)$ represents the velocity of the k^{th} element of particle j in swarm S and $v_{max}^j = \delta_{vmax} * \left(x_{\max}^j - x_{\min}^j\right)$, δ_{vmax} is the step size, and $k \in N$.

f.) Update the position of each particle as

$$\boldsymbol{x}_j(t+1) = \boldsymbol{x}_j(t) + \boldsymbol{v}_j(t+1). \tag{3.7}$$

g.) Repeat the above steps until a stopping criterion is met.

3.1.3. **Performance Tuning of PSO**

The efficiency and accuracy of PSO is determined by its ability to operate over large search spaces, known as exploration, and its ability to concentrate on promising areas to refine a candidate solution, known as exploitation. The proper selection of control parameters can enhance the performance efficiency and convergence capability of PSO. This section discusses the influence and tuning techniques of a number of control parameters, namely the size M of the swarm, the acceleration coefficients (c_1, c_2), the inertia weight w, number of iterations, maximum velocity of particles v_{max} and the random values r_1, r_2 that weight the cognitive and social terms.

1.) *Population size:* The number of particles in the swarm determines the search space region. A large swarm allows more search space exploration per iteration but increases the overall computational cost. A smaller swarm on the other hand may need a greater number of iterations to reach an optimal solution as compared to large swarm. Clerc [41] has determined that a swarm size of 10 + $2\sqrt{2 * N}$ where N is the number of dimensions in the problem, is sufficient to ascertain the global optimum.

2.) *Efficient Initialization*: The basic PSO operates by updating the randomly initialized population, based on random sampling from uniform distribution. Clerc [41] reviewed the different initialization techniques for standard PSO and the majority of the techniques have chosen initial population as $x_j^k = U\left(x_j^{k,max}, x_j^{k,\min}\right)$ where x_j^k is the k^{th} element of decision vector $\boldsymbol{x}_j$ randomly sampled from a uniform distribution, and $x_j^{k,max}, x_j^{k,\min}$ are the upper and lower limits on decision variable x_j^k, where $k \in N$.

3.) *Acceleration Coefficients ($c_1$and c_2):* The selection of acceleration coefficients is usually problem dependent. Most illustrations use equal values for $c_1$and c_2 so that the particles are attracted towards the average of $\boldsymbol{p_j^l(t)}$ and $\boldsymbol{p_S^g(t)}$, but the values are typically tuned for the specific problem. For example, in the case of unimodal problems, a larger value for the social component ($c_2 > c_1$) can be more effective to reach global best solution. However, for rough multimodal problems, the larger cognitive component ($c_1 > c_2$) may allow particles to explore larger search space and eliminate pre-mature convergence to local minima. Various techniques are proposed in literature to initialize values for the acceleration coefficients. Ratnaweera *et al*. [42] proposed linear scaling with a cognitive bias during the early stages of optimization and a social bias at the end, as given by

$$c_1(t) = \left(c_{1_{min}} - c_{1_{max}}\right) * \frac{t}{t_{max}} + c_{1_{max}} \tag{3.8}$$

$$c_2(t) = \left(c_{2_{max}} - c_{2_{min}}\right) * \frac{t}{t_{max}} + c_{2_{min}} \tag{3.9}$$

where $c_{1_{min}} = c_{2_{min}} = 0.5$, $c_{1_{max}} = c_{2_{max}} = 2.5$ and t_{max} defines the maximum number of iterations. We took this approach.

4.) *Inertia Weight (w):* Shi and Eberhart [43] introduced this quantity as a mechanism to constrain the movement of particles by weighing the previous velocity to prevent sudden change in direction of particles. It helps tune swarm exploration and exploitation processes, with large values facilitating a global search and small a local search. For $w \geq 1$, velocity increases with each iteration and for $w < 1$, velocity decreases with each iteration. It should be noted that the choice of w is made in conjunction with c_1, c_2. Naka *et al.* [44]proposed a linearly decreasing inertia weight to facilitate exploration at early stages, as particles preserve the memory of previous flight directions, and to promote local exploitation during the last stages of the optimization process. They give

$$w(t) = \frac{\big(w(0) - w(t_{max})\big)(t_{max} - t)}{t_{max}} + w(t_{max}), \tag{3.10}$$

where t_{max} is the maximum number of iterations, $w(0)$ is the initial inertia weight (usually 0.9) and $w(t_{max})$ is the final inertia weight (usually 0.4). This is the approach we take.

5.) *Velocity Clamping (v_{max}):* The trade off between exploration and exploitation is balanced by clamping the velocity of particle j in the velocity update equation (3.5). Velocity clamping acts a parameter to control the step size of the movement of particles in the multi-dimensional search space. However, this leads to the problem of selecting the appropriate values for the maximum velocity. Shi and Eberhart [45] tested the effects of different values of maximum velocity v_{max} for some common benchmark problems. They proposed that, given lack of knowledge, v_{max} can be set equal to x_{max}, the maximum value of particle j.

6.) *Termination Criteria:* The termination criteria of PSO is also usually problem dependent. The termination of PSO can be defined in terms of the number of iterations, the best gbest solution thus far achieved, or no further improvement noted in the global best solution.

3.1.4. **Single Particle Position Update Illustration**

The fundamental concept behind PSO is the competition between the three distinct parameters of the velocity update equation – the particle's previous velocity, its cognitive component, and its social component. To illustrate the effect of these parameters on PSO, let us consider a single two-dimensional particle $\boldsymbol{x} = [x_1, x_2]$ in swarm S at iteration $t = 0$, as shown in Figure 3.1(a). The particle subscripts are not mentioned to avoid notational inconvenience. The particle at $x(t)$ updates its position at iteration $t = t + 1$ as per velocity update equation (3.5). As seen from Figure 3.1(b), at iteration $(t + 1)$, assuming the personal best position (p^l) remains unchanged, the particle moves towards the global best position (p^g) thus far achieved by the swarm.

3.1.5. **Swarm Position Update Illustration**

To visualize the effect of the velocity update equation on the swarm behavior, we consider a swarm with 10 1D-particles. The fitness function (3.1) is to be minimized with its optimum at origin. Figure 3.2(a) depicts the initial state of swarm at iteration t = 0, with

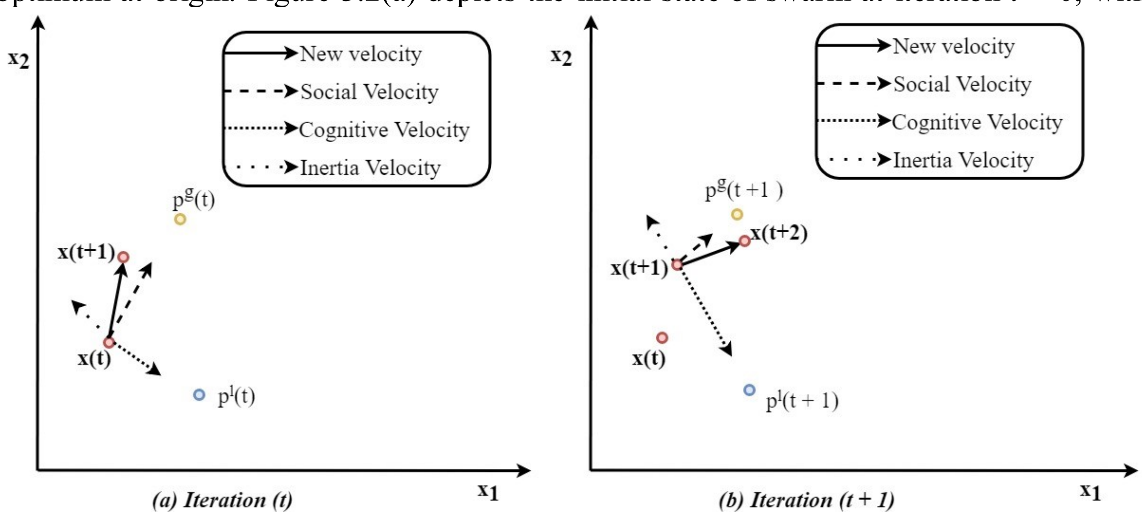

Figure 3.1 Illustration of velocity and position update for 1D particle

particles randomly oriented. At iteration $t = t + 1$, for a zero initial velocity, considering the particles are influenced by the global best position only as cognitive component is equal to initialized state, the particles fly towards global best position as shown in Figure 3.2(b).

With velocity and position update until maximum iterations t_{max}, the swarm finds the optimal solution to objective function (3.1), which is the origin as shown in Figure 3.2(c).

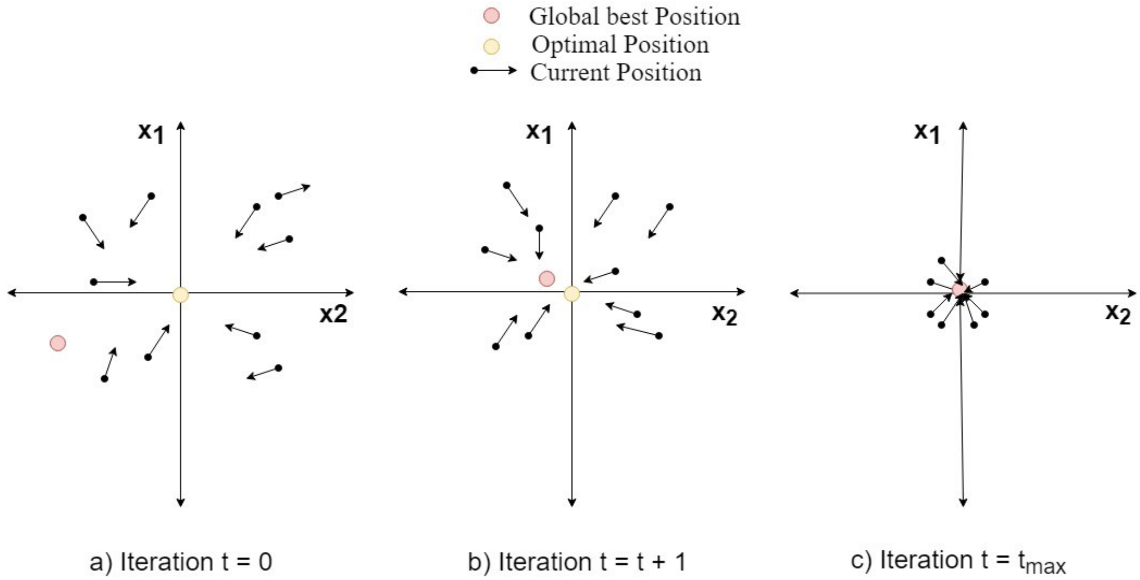

Figure 3.2 A 10-particle position and velocity update Illustration

3.2. Constrained PSO

In the previous section we discussed the working of PSO for solving unconstrained optimization problems. However, most real-world optimization problems have equality or inequality constraints on the decision variables or vectors. PSO is considered efficient for solving constrained optimization problems and has been so applied [46]–[48]. The application of PSO to constrained optimization problems requires efficient mechanisms to ensure that optimal solution is within the constrained search space. The most common techniques used to handle both equality and inequality constraints in PSO is the penalty function method, rejection of infeasible solution, and conversion to unconstrained problem. Figure 3.3 describes the process of solving constrained optimization problems using PSO. The algorithm described in Section 3.1.2 is now presented as a flow chart, where now we include a boundary constraint validate step that is more fully described immediately below.

3.2.1. **Penalty Function Method**

The constrained optimization problem consists of an objective function to be optimized subject to equality and inequality constraints. The penalty function method adds a function to the fitness function to penalize infeasible solutions, discouraging particles from moving towards the infeasible regions of the search space. To demonstrate the working of the penalty function method, since it is employed in this thesis, let us consider the constrained linear optimization problem (3.1) along with set of constraints formulated as

$$minimize\ f(\boldsymbol{x})$$

$$subject\ to:\ l_m(\boldsymbol{x}) \leq c_m, m = 1,2,\dots,N_1 \tag{3.11}$$

$$g_k(\boldsymbol{x}) \geq c_k, k = 1,2,\dots,N_2 \tag{3.12}$$

$$e_i(\boldsymbol{x}) = c_i, i = 1,2,\dots,N_3 \tag{3.13}$$

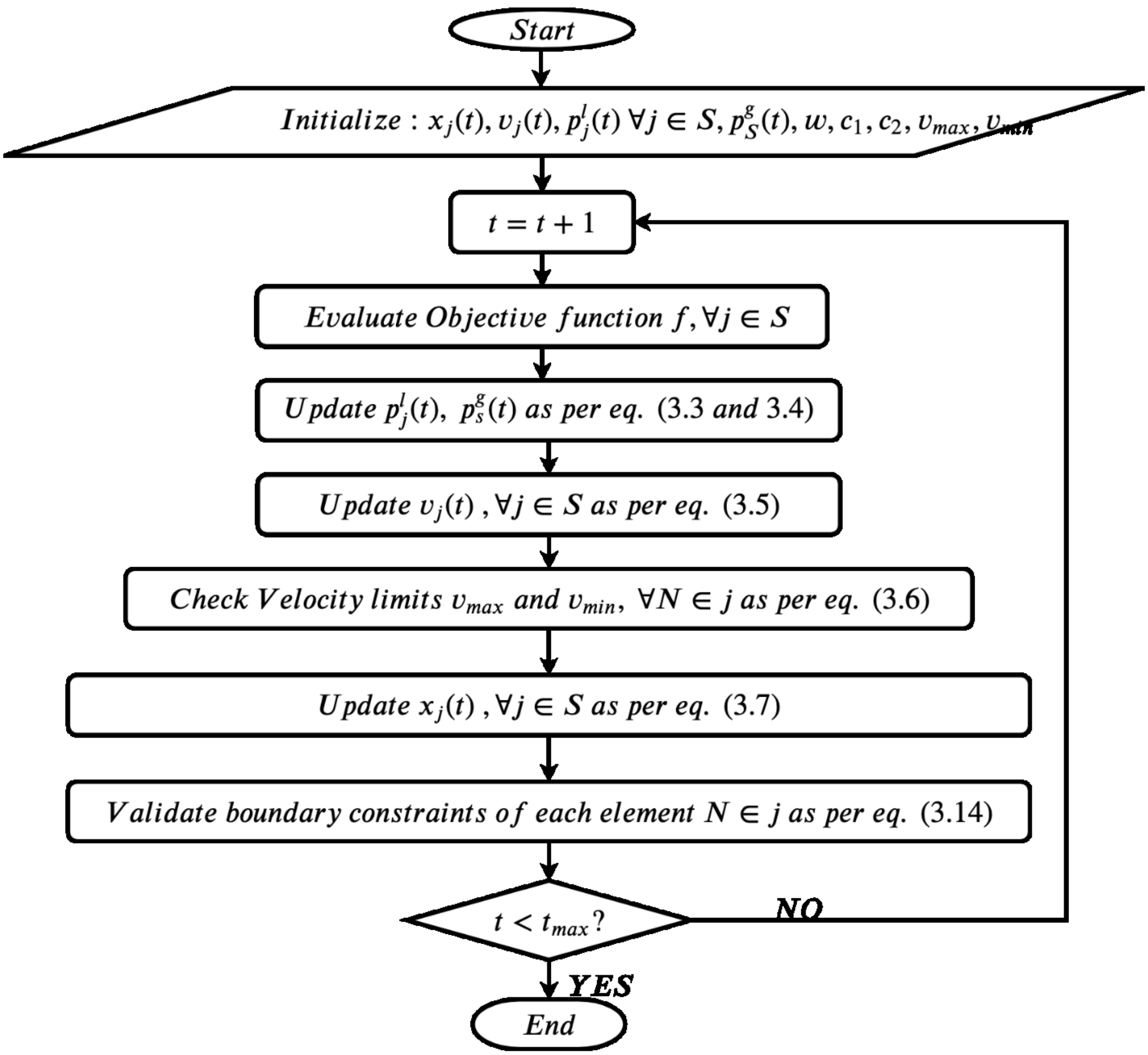

Figure 3.3 Flowchart of Penalty function approach for PSO

Where $\boldsymbol{x}$ is the vector to be optimized, m and k refer to inequality constraints and i refers to equality constraints; c_m, c_k, c_l are constant values defined in the problem statement. Each element of the decision vector may also be subject to boundary conditions, such as

$$x_i \in (x_{max}, x_{min}). \tag{3.14}$$

To solve this constrained optimization problem using PSO, the penalty function method adds another term to the objective function and hence the combined objective function is :

$$F(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}), & x \in \ feasible\ region \\ f(\boldsymbol{x}) + \lambda\, P(\boldsymbol{x}), & x\ \in infeasible\ region \end{cases} \tag{3.15}$$

where λ is the positive penalty function coefficient of sufficiently high magnitude to search within the constrained feasible space; and *P(**x**)* is the imposed penalty function computed as

$$P(\boldsymbol{x}) = \sum_{m=1}^{N_1} \max(0, (l_m\ (\boldsymbol{x}) - c_m)) + \sum_{k=1}^{N_2} \max(0, (c_k - g_k(\boldsymbol{x}))$$
$$+ \sum_{i=1}^{N_3} (\ E_i(\boldsymbol{x}) - c_i)^2 \tag{3.16}$$

Figure 3.4 illustrates the penalty function method. Here, $\boldsymbol{x}^*$ indicates the optimal solution, $\boldsymbol{p}^{g}(t)$ indicates the global best position, and $\boldsymbol{p}^{l}(t)$ indicates the personal best position of particle j. The entire search space is represented implicitly, with the feasible regions

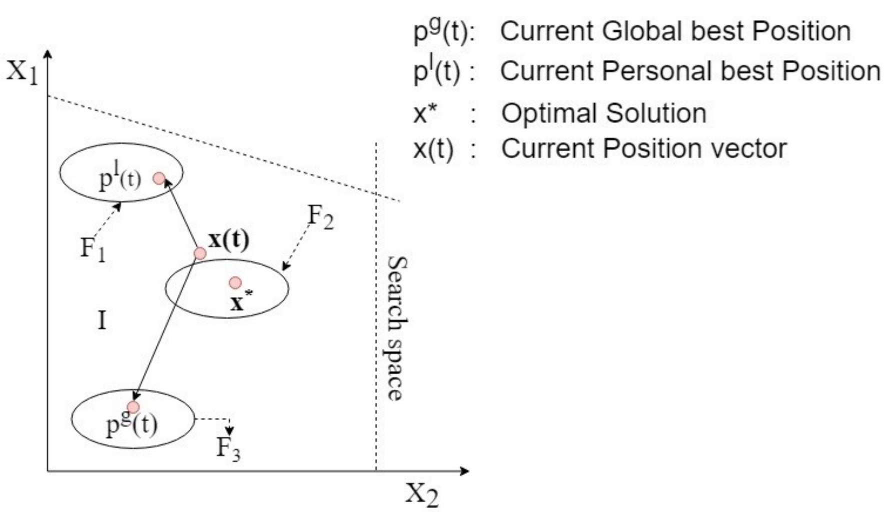

Figure 3.4 Illustration of Penalty Function Method

indicated as F_s and the rest as the infeasible search space I_s. Because the fitness function is adapted by adding a penalty function to fitness function value if $\boldsymbol{x}_n(t) \in I_S$,hence discouraging particles to move away from optimum solution. Similarly, if $\boldsymbol{x}_n(t) \in F_S$ the penalty function term is zero and particles are encouraged to explore feasible search space. This leaves the problem of finding the appropriate value of penalty function coefficient as using static penalty coefficient, particles $\boldsymbol{p}^l(t)$ may not change even if it moves towards optimal solution. This can be avoided by either emphasizing more on social component or using an iteration-dependent penalty function coefficient.

3.2.2. **Reject Infeasible Solution**

The most basic method to handle constraints is to reject the infeasible particles $\boldsymbol{x}_n\ (t) \in I_s$. Infeasible particles can be treated by not allowing them to be selected as personal best or global best solutions so they will not influence other particles in the swarm. Another mechanism is to reinitialize the infeasible solutions with new randomly generated positions from within the feasible search space. The later mechanism is usually used in gbest PSO to handle the boundaries constraint (3.14) by replacing them with user defined values i.e.

$$x_n = \begin{cases} x_{max}, & x_n > x_{max} \\ x_{min}, & x_n < x_{min} \end{cases} \tag{3.17}$$

3.3. Binary PSO

Kennedy and Eberhart [49] developed the Binary version of PSO to address binary decision vectors, which occur in our work. In BPSO, particles represent the binary search space, that is, the decision variable $\boldsymbol{x} \in \{0,1\}$. The basic process of Binary PSO for solving the optimization problem is similar to the continuous gbest PSO except for the position update equation and the selection of initial parameters, specifically the maximum velocity and inertia weight. They tested different values of maximum velocity and inertia weight for a set of benchmark problems[49]. Figure 3.5 shows the working process of BPSO to solve

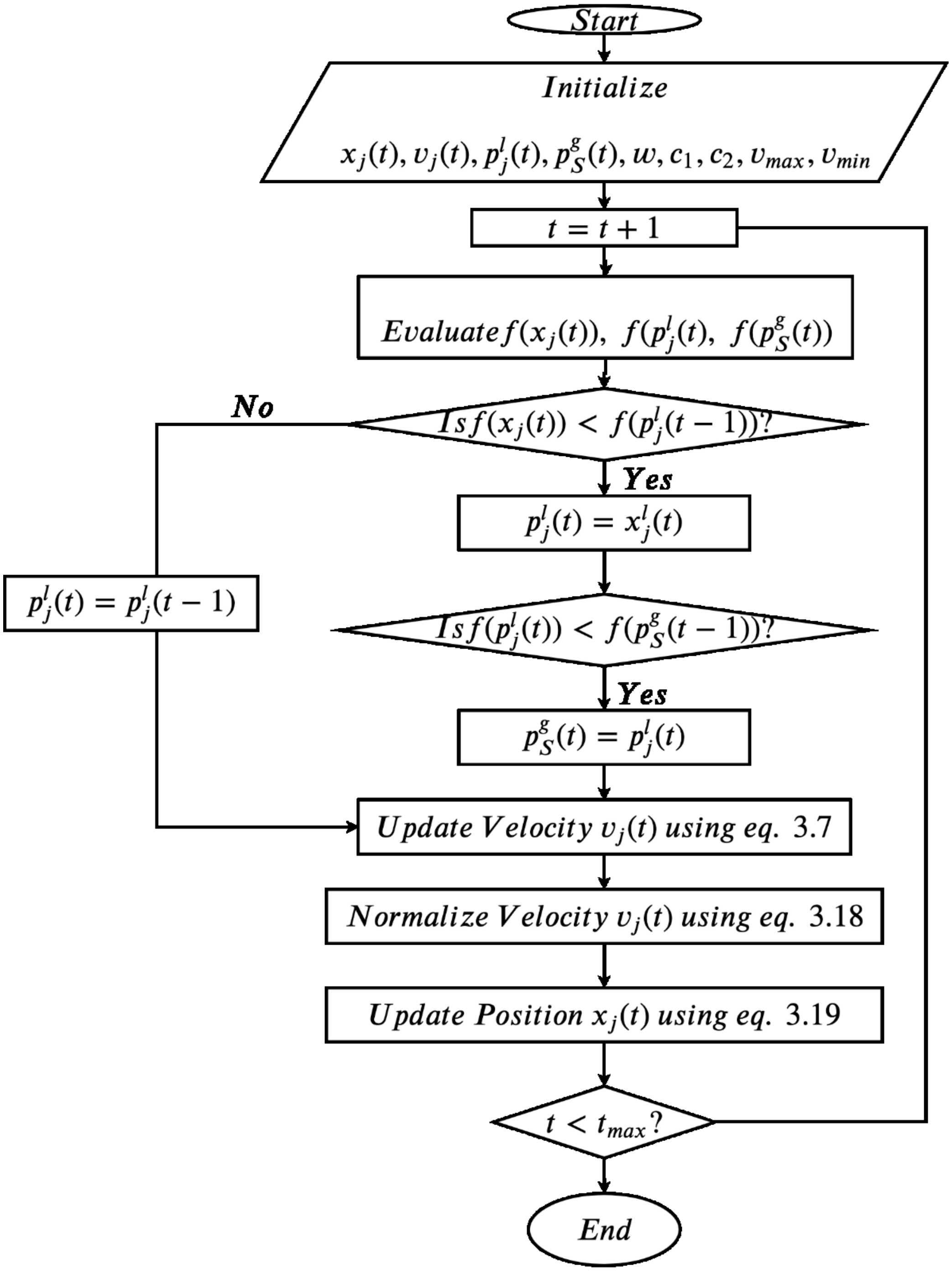

Figure 3.5 Working of Binary PSO

constrained optimization problems. Velocity updates are defined in terms of the probability of a bit to change from one state to other. Different approaches have been developed to normalize the velocities to avoid pre-mature convergence due to limited exploration abilities. The most common method to normalize the velocities is with the sigmoid function, as

$$v_j^N(t) \leftarrow sigmoid\left(v_j^N(t)\right) = \frac{1}{1 + \exp^{\left(-v_j^N(t)\right)}}. \tag{3.18}$$

To illustrate the binary assignment process, consider a velocity of $v_j^N(t) = 0.4$. This signifies a 40% chance of x to be bit 1 and a 60% chance to be bit 0. Here, $r(t) \sim U(0,1)$, is the random number between 0 and 1 sampled from a uniform distribution. Therefore, the position update changes to

$$x_j^N(t+1) = \begin{cases} 1 & if\ r(t) < sigmoid\left(v_j^N(t+1)\right) \\ 0 & if\ r(t) \geq sigmoid\left(v_j^N(t+1)\right) \end{cases}. \tag{3.19}$$

3.4. Pros and Cons of PSO

3.4.1. Pros

The advantages of PSO approaches are the following:

I. Particle swarm optimization is a gradient free algorithm, and thus can function well where the latter tends to fail, such as noisy and discontinuous functions.

II. Since few parameters are involved, PSO is typically easy to implement.

III. PSO performs better compared than gradient based algorithms in high dimensional search spaces.

IV. The algorithm can be easily parallelized for concurrent processing.

V. The Global Best has ability to converge to a near optimal solution in few iterations.

3.4.2. Cons

The disadvantages of PSO approaches are the following:

I. Improper selection of initial parameters can lead to a sub-optimal solution.

II. Computation cost is directly proportional to the dimensions of search space.

3.5. Enhanced PSO

Most the real-world combinatorial optimization problems are solved by reducing the length of the operating dimensions, *i.e.,* solving the problem by parts, which may lead to an overall sub-optimal solution as there is always a trade off between optimality and complexity. The basic PSO is designed to operate over fixed dimensions, however it can be extended to solve multi-dimensional combinatorial problems. The fundamental concept behind MDPSO is that multiple swarms share personal and social information among themselves to minimize a commonly shared fitness function.

3.5.1. **Multi-Dimensional PSO (MDPSO)**

Instead of optimizing over a single swarm or search space, we consider a swarm $S = \{S_1, S_2, \dots, S_D\}$ divided into D sub-swarms where the length of S is defined by the optimization problem to be solved. Swarm S consists of N_s particles distributed throughout the *D* subswarms, each of which have dimensionality *N*. Let each particle (*i.e.,* position vector) $n \in N_S$ be denoted by $\overrightarrow{x_n}$, and be represented as the array of decision vectors

$$\overrightarrow{\boldsymbol{x}_n} = \left[\boldsymbol{x}_n^{S_1}, \boldsymbol{x}_n^{S_2}, \dots, \boldsymbol{x}_n^{S_D}\right], \tag{3.20}$$

where $\overrightarrow{\boldsymbol{x}_n}$ represents n^{th} position vector in swarm S; $\boldsymbol{x}_n^{S_i} = \left[x_{n,1}^{S_i}, x_{n,2}^{S_i}, \dots, x_{n,N}^{S_i}\right]$ is the decision vector of particle n in swarm S_i of dimension N; and $x_{n,j}^{S_i}$ is the j^{th} element of the decision vector of particle n in swarm S_i of dimension N. The swarm structure is illustrated in Figure 3.6.A swarm *S,* is composed of multiple sub-swarms $(S_1, S_2, \dots, S_D)$. Each particle in dimension *D* of swarm *S* updates its *local* best position and *global* best position based on the minimum value of fitness function achieved by swarm *S*.

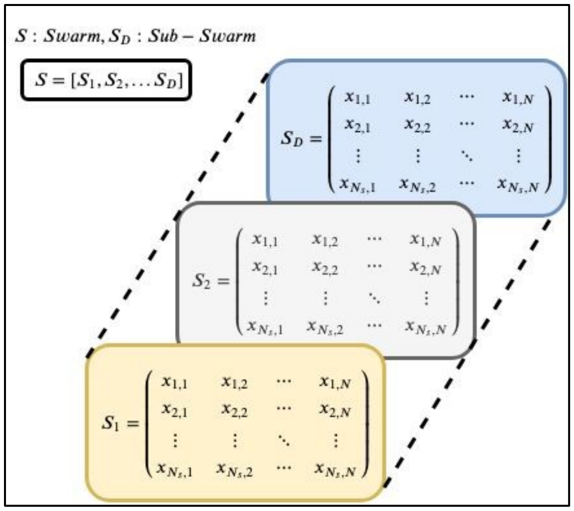

Figure 3.6 Swarm Structure

The fitness function for particle n may be defined as

$$f(\overrightarrow{\boldsymbol{x}_n}) = \sum_{i=1}^{N}\sum_{j=1}^{D} \boldsymbol{x}_{n,i}^{S_j}. \tag{3.21}$$

Each particle $\overrightarrow{\boldsymbol{x}_n}$ has associated with it a personal best position vector $\overrightarrow{\boldsymbol{P}_n^l}$, a global best position vector $\overrightarrow{\boldsymbol{P}^g}$, and a velocity vector $\overrightarrow{\boldsymbol{v}_n}$, as

$$\overrightarrow{\boldsymbol{P}_n^l} = \left[\boldsymbol{P}_n^{l,S_1}, \boldsymbol{P}_{n,}^{l,S_2}, \dots, \boldsymbol{P}_n^{l,S_D}\right], \tag{3.22}$$

$$\overrightarrow{\boldsymbol{P}^g} = [\boldsymbol{P}^{g,S_1}, \boldsymbol{P}^{g,S_2}, \dots, \boldsymbol{P}^{g,S_D}], \tag{3.23}$$

$$\overrightarrow{\boldsymbol{v}_n} = \left[\boldsymbol{v}_n^{S_1}, \boldsymbol{v}_n^{S_2}, \dots, \boldsymbol{v}_n^{S_D}\right], \tag{3.24}$$

where $\boldsymbol{P}_n^{l,S_i} = \left[P_{n,1}^{l,S_D}, P_{n,2}^{l,S_D}, \dots, P_{n,N}^{l,S_D}\right]$ is the n^{th} local (l) best position (*i.e.,* decision) vector of sub-swarm S_i of dimension N; $\boldsymbol{P}^{g,S_j} = \left[P_1^{g,S_D}, P_2^{g,S_D}, \dots, P_N^{g,S_D}\right]$ is the global (g) best position vector of sub-swarm S_j of dimension N; and $\boldsymbol{v}_n^{S_k} = \left[v_{n,1}^{S_D}, v_{n,2}^{S_D}, \dots, v_{n,N}^{S_D}\right]$ is the velocity vector of sub-swarm S_k of dimension N.

Each element of the position vector $x_n^{S_i}$ and velocity vector $v_n^{S_j}$ is constrained by upper and lower bounds $x_{n,N}^{S_i} \in (x_{max}, x_{min})$ & $v_{n,N}^{S_D} = \delta * (x_{max} - x_{min})$, where delta is empirically determined. The position vector bounds define the search space for entire swarm, and velocity vector defines the step size of position vector movement towards the next possible optimal solution.

The personal best cost of swarm S is computed from the fitness function as

$$f\left(\overrightarrow{\boldsymbol{P}_n^l}\right) = \sum_{i=1}^{N}\sum_{j=1}^{D} \boldsymbol{P}_{n,i}^{S_j}, \tag{3.24}$$

and the global best cost of swarm S is found by

$$P^{gcost} = \min\left\{ f\left(\overrightarrow{\boldsymbol{P}_1^l}\right), f\left(\overrightarrow{\boldsymbol{P}_2^l}\right), \ldots, f\left(\overrightarrow{\boldsymbol{P}_{Ns}^l}\right)\right\}. \tag{3.25}$$

The particles check all the possible combinations during each iteration t and updates the position based on local best experience $\boldsymbol{f}\left(\overrightarrow{\boldsymbol{P}_{\boldsymbol{n}}^{\boldsymbol{l}}}\right)$ and global best experience P^{gcost} of the entire swarm S. The velocity and position update equations can then be written as

$$\overrightarrow{\boldsymbol{v}_{\boldsymbol{n}}}(t+1) = \overbrace{w(t).\overrightarrow{\boldsymbol{v}_{\boldsymbol{n}}}(t)}^{inertia\ weight} + \overbrace{c_1.\boldsymbol{r}_{\boldsymbol{1},\boldsymbol{j}}(t)\left[\overrightarrow{\boldsymbol{P}_{\boldsymbol{n}}^{\boldsymbol{l}}} - \overrightarrow{\boldsymbol{x}_{\boldsymbol{n}}}(t)\right]}^{\text{cognitive component}} + \overbrace{c_2.\boldsymbol{r}_{\boldsymbol{2},\boldsymbol{j}}(\mathrm{t})\left[\overrightarrow{\boldsymbol{P}^{\boldsymbol{g}}} - \overrightarrow{\boldsymbol{x}_{\boldsymbol{n}}}(t)\right]}^{\text{social component}} \tag{3.26}$$

$$\overrightarrow{\boldsymbol{x}_{\boldsymbol{n}}}(t+1) = \overrightarrow{\boldsymbol{x}_{\boldsymbol{n}}}(t) + \overrightarrow{\boldsymbol{v}_{\boldsymbol{n}}}(t+1) \tag{3.27}$$

Figure 3.7 demonstrates how each sub-swarm co-operates with each other by sharing the cognitive and social knowledge. The cognitive acceleration coefficient c_1 linearly decreases as per equation (3.8), social acceleration coefficient c_2 is linearly increased as per equation (3.9) and inertia weight is linearly decreased as per equation (3.10) to explore maximum search space at early stages and refine the solution at last stages of optimization. Algorithm 3.1 describes the process flow for solving multi-dimensional problems using enhanced PSO.

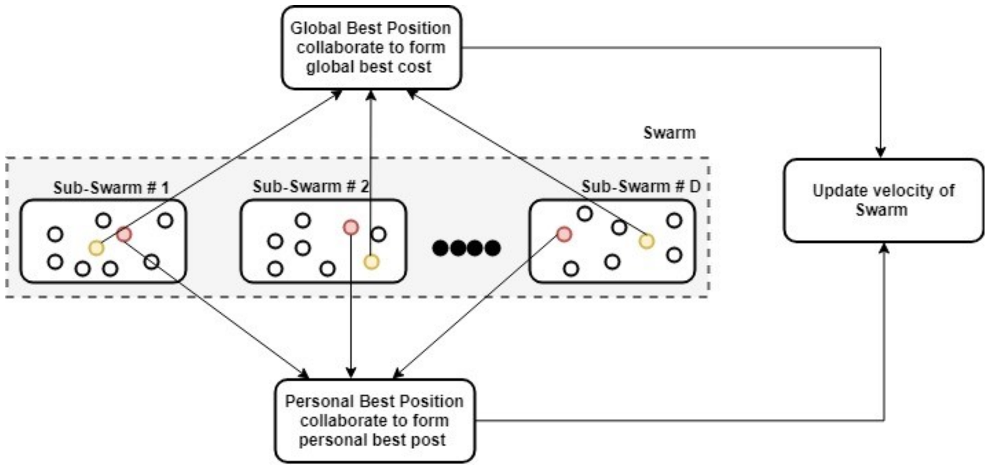

Figure 3.7 Sub-Swarms Coordination Mechanism

Algorithm 3.1 Pseudo Code for MDPSO

Create and initialize a random population.
Set the initial parameters $w, c_{1max}, c_{2max}, c_{1min}, c_{2min}, t_{max}, \delta, x_{max}, x_{min}$
Initialize $\overrightarrow{\boldsymbol{v}_n}, \overrightarrow{\boldsymbol{P}_n^l}, \overrightarrow{\boldsymbol{P}^g}$
while $t \leq t_{max}$
    for each particle $n \in N_S$ $\boldsymbol{do}$
    Evaluate $f\left(\overrightarrow{\boldsymbol{x}_n(\boldsymbol{t})}\right), f\left(\overrightarrow{\boldsymbol{P}_n^l(\boldsymbol{t})}\right), \boldsymbol{P}^{gcost}$
    //update the personal best position
    if $f\left(\overrightarrow{\boldsymbol{x}_n}(t+1)\right) < f\left(\overrightarrow{\boldsymbol{P}_n^l(\boldsymbol{t})}\right)$ **do**
        $\overrightarrow{\boldsymbol{P}_n^l}(\boldsymbol{t}+\boldsymbol{1}) = \overrightarrow{\boldsymbol{x}_n}(t+1)$
    end
    // update the global best position
    if $f\left(\overrightarrow{\boldsymbol{P}_n^l(\boldsymbol{t}+\boldsymbol{1})}\right) < P^{gcost}(t)$
        $\overrightarrow{\boldsymbol{P}^g}(\boldsymbol{t}+\boldsymbol{1}) = \overrightarrow{\boldsymbol{P}_n^l(\boldsymbol{t}+\boldsymbol{1})}$
    end
    for each particle $n \in N_S$ do
        update the velocity as per equation (3.26)
        update the position as per equation (3.27)
    end
$\boldsymbol{t} = \boldsymbol{t} + \boldsymbol{1}$

4. SYSTEM MODEL AND PROPOSED ALGORITHM

This chapter presents the higher-level energy management architecture for a neighbourhood system, followed by system description of smart house. We then categorize appliances into different classes based on their flexibility in energy consumption, operating time, and DR potential. The energy consumption scheduling problem is formulated as a multi-objective optimization problem and each category of appliances is modelled as constraints of the problem. Furthermore, the working of the HEMS under different load shaping strategies is discussed. We also present a detailed description of our load reshaping engine based on MDPSO for scheduling different categories of appliances.

4.1. The System Model

We consider a secondary power distribution network with a single point of common coupling (POCC) and H dwellings, as shown in Figure 4.1. An neighbourhood aggregator, which may be considered as a utility company transactive agent, resides at the POCC (typically a distribution step down transformer) to reshape the aggregate load profile and protect asset health (*i.e.,* the overloading of distribution transformer. Each smart dwelling has an intelligent HEMSC, a smart meter, smart thermostats, and other smart loads. In order to preserve the privacy of customer, each HEMSC reports its aggregate load profile to the neighbourhood aggregator and optimally schedules the operating time and energy consumption of its smart appliances in response to pricing signals or load shifting request from the aggregator. Figure 4.2 shows the communication and power links between various devices inside a smart dwelling. HEMS controller interfaces with the set of appliances inside a dwelling and is connected to aggregator through a smart meter. Communication between HEMSC and aggregators can be wired or wireless and should use appropriate communication protocols and standards, but this is not the focus of this work.

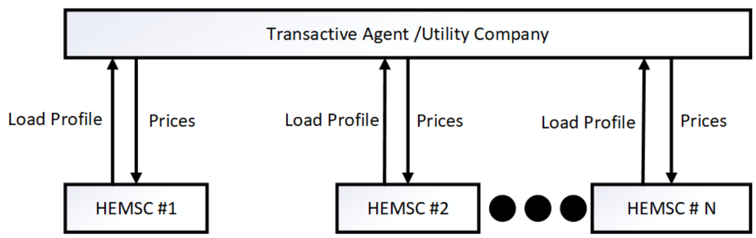

Figure 4.1 Higher level system model

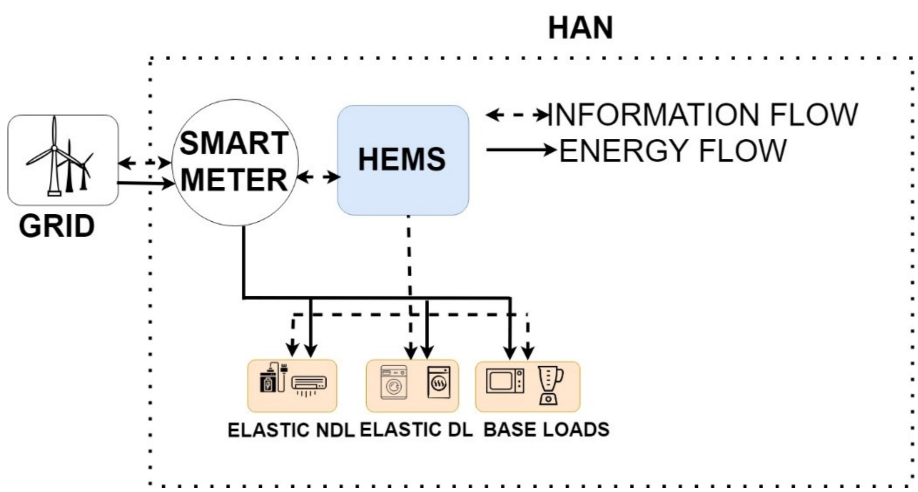

Figure 4.2 Smart Home Model

4.2. General Categorization

A smart home consists of various types of loads with unique operating characteristics. To model each type of load individually can be a significant barrier to HEMS deployment. In order to provide the flexibility to integrate new loads, and to reduce modelling complexity, the appliances with similar characteristics are grouped together by category. We categorize the loads based on their DR potential and operating characteristics. There are three fundamental appliance categories which we define hierarchically in terms of their elasticity, deferability, and interruptibility. Elasticity implies loads with flexible energy consumption during any given time interval; deferrablity implies loads that may be time shifted; and interruptibility implies loads whose operation may be paused and restarted.

Various combinations of these classes yield appliance types. A simple (and fundamental) type is a fixed load – inelastic→non-deferrable→uninterruptible.

We consider the various appliance types, and the operating constraints applied to each, first broadly grouping the classes according to their dominant character – consumption or scheduling. This load categorization is presented in Figure 4.3, and further described below. In addition to the simple Fixed (F) loads, we generally classify appliance types as either Elastic Non-Deferrable Loads (E), Inelastic Deferrable Loads (I), or Fixed. E loads are those whose energy consumption can be modified in each time interval in response to a Pricing or DR signal. I loads are those whose energy consumption profile is fixed but whose operating cycle can be scheduled with or without intermittent interruption in response to pricing or DR signal from utility.

Elastic loads are further sub-categorized as class 1 (Ec_1), where total energy consumption over the operating time is fixed (*i.e.,* constrained elasticity); class 2 (E_{c_2}), where total energy consumption over operating time is not fixed (*i.e.,* unconstrained elasticity); and class 3 (E_{c_3}), where energy consumption is a dependent on external parameter (*i.e.,* comfort-based).

Inelastic loads are further sub-categorized as Interruptible (I_{in}), which can be shifted in time with intermittent interruptions in their operating cycles (*e.g.,* clothes dryer); and Uninterruptible (I_{un}), which can only be shifted in time without any interruption in its duty cycle (*e.g.,* washing machine).

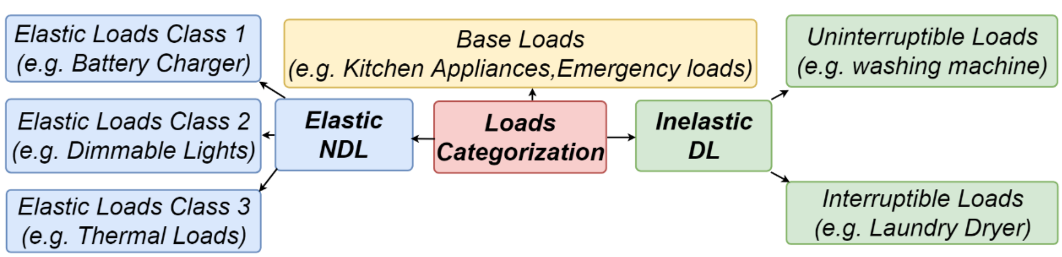

Figure 4.3 Categorization of Loads

4.3. Load Modelling

We consider H smart homes where appliances a are distributed across the load sets $E = \{E_{c_1} \cup E_{c_2} \cup E_{c_3}\}$, $I = \{I_{un} \cup I_{in}\}$ and F. Let us consider the scheduling of loads within a time period T, divided into N equal time intervals t, such that $T = \{t_1, t_2, \dots, t_N\}$. HEMSC optimal schedules the energy consumption of elastic appliances and optimal operation time of inelastic appliances in response to pricing signals and DR requests. We define a binary scheduling decision vector $\boldsymbol{X_a}$ for appliance a as

$$\boldsymbol{X_a} \triangleq \left[x_a^{t_1}, x_a^{t_2}, \dots, x_a^{t_N}\right], \forall\, a \in I, \tag{4.1}$$

where x_a^t is unity if a is on during t or zero if it is off. We also define an energy consumption scheduling decision vector $\boldsymbol{J_a}$ as

$$\boldsymbol{J_a} \triangleq \left[j_a^{t_1}, j_a^{t_2}, \dots, j_a^{t_N}\right], \forall\, a \in E, \tag{4.2}$$

where j_a^t is the appliance a energy consumption during t.

Since optimization proceeds either by consumption or by scheduling, $\boldsymbol{X}$ and $\boldsymbol{J}$ are conjugate decision vectors. The former is thus used solely for I loads, and the latter for E loads. Constraints on appliance operation are used to address the conjugate quantity. As regards the latter, we define a permissible operating time window $\tau_a = \left[t_s^a, t_{s+1}^a, \dots, t_f^a\right]$, where t_s^a and t_f^a are the starting and finishing times, respectively, of appliance a. These are governed by user behavior.

4.3.1. Fixed Loads (*F*)

Fixed loads are non-manageable loads whose energy consumption and operating time cannot be modified (*i.e.,* those loads not controlled by the HEMS) in response to pricing signals. They may define in a manner similar to (4.3), as

$$\boldsymbol{J_f} \triangleq \left[j_a^{t_1}, j_a^{t_2}, \dots, j_a^{t_N}\right] \forall\, a \in F, \tag{4.3}$$

where j_a^t is the energy consumption of appliance a in the interval $t \in T$

4.3.2. **Elastic Non-Deferrable Loads (*E*)**

Elastic Non-Deferrable Loads is the category of appliances with flexibility in the amount of energy consumption over a permissible operating window τ_a. A detailed description of its sub-categories along with their operating constraints is provided below.

4.3.2.1. Elastic Loads Class 1

The set of appliances E_{c_1} (constrained elasticity; *e.g.* battery charger) are constrained within each time interval t across the permissible operating window τ_a first by minimum and maximum energy consumption. Thus

$$j_{c_{1a}}^{min} \leq j_{c_{1a}}(t) \leq j_{c1_a}^{max}, \forall\, a \in E_{c_1} \tag{4.4}$$

They are further constrained by restricting the total permissible energy consumption j_{tot} in the scheduling period T as

$$\sum_{t=t_s}^{t_f} j_{c_{1a}} = j_{tot}, \forall\, a \in E_{c_1}. \tag{4.5}$$

We may further constrain E_{c_1} by requiring $t \in \tau_a$ to accommodate scheduling constraints to address time shifting and interruptibility.

4.3.2.2. Elastic Loads Class 2

The set of appliances E_{c_2} (unconstrained elasticity; *e.g.,* dimmable lights) are constrained only by minimum and maximum energy consumption within each time interval *t* across the permissible operating window. Thus

$$j_{c_{2a}}^{min} \leq j_{c_{2a}}(t) \leq j_{c_{2a}}^{max}, \forall\, a \in E_{c_2}. \tag{4.6}$$

There is no explicit constraint on j_{tot}, but again, we may further constrain E_{c_2} by requiring $t \in \tau_a$ to accommodate scheduling constraints.

4.3.2.3. Elastic Loads Class 3

The set of appliances E_{c_3} are those whose energy consumption is influenced by perceived environmental factors (setpoint-based; *e.g.,* HVAC, water heater, refrigeration). We restrict ourselves to thermostatically controlled loads. As before, such loads are constrained within each time interval *t* across the permissible operating window first by the minimum and maximum energy consumption, whence

$$j_t^{c_{3a},min} \leq j_t^{c_{3a}} \leq j_t^{c_{3a},max} \quad , t \in \tau_a \ , \forall \ a \in E_{c_3}. \tag{4.7}$$

Compared to earlier notation, the subscript and subscript have been here reversed, for convenience. Second, there is an explicit comfort constraint based on minimum and maximum permissible temperatures

$$\theta_t^{a,min} \leq \theta_t^{a,in} \leq \theta_t^{a,max} \quad , \ t \in \tau_a \ , \forall \ a \in E_{c_3}, \tag{4.8}$$

where θ_t^a is the temperature of system at time interval t.

We use the normalize discrete time model given in [25] to compute the indoor temperature in time interval t as

$$\theta_{t+1}^{a,in}(j_t^a) = \zeta\theta_t^{a,in}(j_t^a) \pm (1-\zeta)(\theta_t^{out} - \eta u_t), \tag{4.9}$$

where the first term on the right gives the contribution from previous time slot and the second term provides the heating (+) or cooling (–) effect; $\theta_t^{a,in} = \left(\theta_t^{a,in} - \theta^d\right)/\theta_e$ is the interior temperature at time interval *t*, with $\theta_e$and θ_d scaling and offset the terms, respectively; and θ_t^{out} is the exterior temperature at time interval *t*. Environmental and appliance-related quantities are *(i)* the factor of inertia $\zeta = e^{-\delta/ℸ}$, where ℸ is duration of each subinterval *t*, and $\delta = mc/A$ is the time constant of the system, where mc is the total thermal mass and *A* is the thermal conductivity; *(ii)* $\eta = (\chi P_a)/(A\theta_e)$ is the steady state temperature gain, where χ is the thermal conversion efficiency, and P_a is the rated energy consumption of appliance a; and *(iii)* $u_t = j_t^a/P^a$ is the control input, where j_t^a is the energy consumption of appliance a in time slot t.

4.3.3. **Inelastic Deferrable Loads (*I*)**

Inelastic Deferrable Loads is the category of appliances with flexibility in their operating time over a permissible operating window τ_a; *i.e.*, appliance $a \in I$ can be deferred in time. A detailed description of their sub-categories along with their operating constraints is discussed below.

4.3.3.1 Inelastic Interruptible Loads

The set of appliances I_{in} are those Inelastic Interruptible Loads that consume a fixed amount of energy within each time interval $t \in \tau_a$ (*e.g.*, PHEV without controllable charging). In terms of demand response potential, I_{in} appliances can be intermittently turned on or off without any performance degradation in response to pricing or DR signals from the utility. In addition, the appliances in I_{in} need an energy consumption constraint to ensure task completion over their operating cycle. This may be given by

$$P_a^{in} \sum_{t=t_s}^{t_f} x_a^t = P_a^{tot} , \tag{4.10}$$

where P_a^{in} is the rated energy consumption of appliance $a \in I_{in}$, P_a^{tot} is the total permissible energy consumption of appliance a over the scheduling period T, and $x_a^t \in \{0,1\}$ is the binary variable of decision vector X_a (zero if the appliance is off and one if the appliance is on).

4.3.3.2 Inelastic Uninterruptible Loads

This set of appliances I_{un} is constrained by a fixed energy consumption in each time interval and a total energy consumption across the scheduling period T (*e.g.,* Washing Machine). The first constraint is therefore the same as (4.10):

$$P_a^{un} \sum_{t=t_s}^{t_f} x_a^t = P_a^{tot} , \tag{4.11}$$

where P_a^{un} is the rated energy consumption of appliance $a \in I_{un}$. The second constraint is the uninterruptibility requirement, which means the operating cycle is over contiguous time invervals. This may be expresses as

$$\sum_{t=t_s}^{t_f - l_p} x_a^{t_s} . x_a^{t_s+1} \dots \dots x_a^{t_s+l_p} = 1 \;\; \forall\, a \in I_{un}, \tag{4.12}$$

where l_p is the length of operating cycle for appliance $a \in I_{un}$.

4.4. Problem Formulation

The overall energy consumption is the aggregate energy consumption across all appliance categories. Thus

$$\boldsymbol{J_{total}} = \sum_{a \in F} \boldsymbol{J_f^a} + \sum_{a \in E} \boldsymbol{J_c^a} + \sum_{a \in I} \boldsymbol{J_i^a}\,, \tag{4.13}$$

where J_f^a, J_c^a, J_i^a are the energy consumption vectors of fixed loads, Elastic Non- Deferrable loads and Inelastic Deferrable loads, respectively. In addition to the constraints discussed above, the aggregate scheduled energy consumption of the house should be less than or equal to the rated capacity of distribution panel (ψ_r). Thus

$$j_{total}^t \leq \psi_r \;\; \forall\, t \in T\,, \tag{4.14}$$

where j_{total}^t is the total scheduled energy consumption of house at each time interval $t \in T$. Note that all the operating constraints of appliances can be defined by the consumer itself or can be computed through intelligent learning algorithms[50], [51].

The optimization problem is formulated as multi-objective problem aimed at minimizing the cost under TOU pricing, comfort conditions, and incentive maximization under DR request.

4.4.1. Design Objectives

We describe objective functions *f* by an appropriate decision vector, which is load category dependent. Customer comfort is treated quadratically with respect to deviation from some nominal uncontrolled, or baseline (*bl*), state. For *E* appliances,

$$f_{comfort} = \sum_{a=1}^{n} \sum_{t=t_s}^{t_f} \left(j_{a,sch}^t - j_{a,bl}^t \right)^2 \qquad \forall\, a \in E, \tag{4.15}$$

where $j_{a,sch}^t$ is the decision variable describing the determined energy consumption of appliance *a* at time *t* and $j_{a,bl}^t$ is the baseline consumption of appliance *a* at time *t*. For *I* appliances,

$$f_{comfort} = \sum_{a=1}^{n} \sum_{t=t_s}^{t_f} j_a^t \left(x_{a,sch}^t - x_{a,bl}^t \right)^2 \,, \forall\, a \in I, \tag{4.16}$$

where $x_{a,sch}^{t}$ is the decision variable describing the determined energy consumption of appliance a at time t and $x_{a,bl}^{t}$ is the baseline consumption of appliance a at time t.

Under general HEMS control (no DR request), electricity cost functions for E and I appliances are defined purely in terms of their decision variables as

$$f_{cost} = \sum_{a=1}^{n} \sum_{t=t_s}^{t_f} P_t \, j_{a,sch}^{t} \quad , \quad \forall \, a \in E, \tag{4.17}$$

$$f_{cost} = \sum_{a=1}^{n} \sum_{t=t_s}^{t_f} P_t \, j_a^t \; x_{a,sch}^{t} \, , \quad \forall \, a \in I, \tag{4.18}$$

respectively, where P_t is the TOU pricing at time t. To address a utility DR request given across a time interval defined by t_{sdr} and t_{fdr}, the starting and finishing times, respectively, we assume a linear response in terms of a pricing signal δ_{dr}^{t}. The DR objective functions are thus

$$f_{dr} = \theta_{dr} \sum_{a=1}^{n} \sum_{t=t_{sdr}}^{t_{fdr}} \delta_{dr}^{t} j_a^t \left(x_{a,bl}^{t} - x_{a,sch}^{t} \right) \forall \, a \in I, \tag{4.19}$$

$$f_{dr} = \theta_{dr} \sum_{a=1}^{n} \sum_{t=t_{sdr}}^{t_{fdr}} \delta_{dr}^{t} \left(j_{a,bl}^{t} - j_{a,sch}^{t} \right) \forall \, a \in E. \tag{4.20}$$

Since DR events are intermittent, this cost function is gated by θ_{dr}, which is either 0 (no DR event) or 1 (DR event). Thus, DR events can be treated as a cost perturbation on TOU pricing or upper power withdrawn limit imposed on participating consumers.

The overall optimization problem is defined as weighted sum of the comfort and cost objectives as

$$f_{objective} = w_1 \, f_{comfort} + w_2 \, (f_{cost} - f_{dr}), \tag{4.21}$$

where $w_1 \geq 0$ and $w_2 \geq 0$ are scalar weights for user comfort and electricity cost, respectively, and $w_1 + w_2 = 1$.The above objective function may be normalized as

$$f_{objective} = w_1\left(\frac{f_{comfort}}{f_{comfort}^{unsch.}}\right) + w_2\left(\frac{(f_{cost} - f_{dr})}{f_{cost}^{max}}\right), \tag{4.22}$$

where $f_{comfort}^{unsch.}$ and f_{cost}^{max} are reference comfort and cost values, respectively.

As discussed in section 3.2.1, we take a penalty function approach to handle constraints and also reject infeasible solutions by replacing them with user-defined boundaries to address the constraints given from equations (4.1) to (4.14). Hence, the objective function (4.22) transforms to

$$f_{objective} + \lambda\left(\sum_{n=1}^{k} p_n + \sum_{i=1}^{m} p_i\right), \tag{4.23}$$

where p_n and p_i is the quadratic loss functions for equality and inequality constraints, respectively. As an example, inequality constraint (4.14) is written as

$$p_i = \max(0, j_{total}^{t} - \psi_r)^2 \; \forall \; t \in T. \tag{4.24}$$

4.5. Load Shifting Strategies

The primary purpose of implementing DR programs is to depress peak power use by actively engaging the end users to reallocate their energy consumption. The above HEMS model is formulated as a mathematical optimization problem whose objective is to minimize consumer cost by scheduling energy demand to lower pricing periods. As each HEMS optimizer aims at minimizing the cost, this may result in the aggregate peak shifting to a previous off-peak period. Furthermore, it will raise the concern of high prices during off-peak periods and overloading of distribution transformer. Thus, it is relevant to investigate the effects of different DR signals like Time of Use (TOU), Critical Peak Rebate (CPR), and power limiting strategies. We propose a load shifting mechanism that can optimize the energy consumption of dwelling based on TOU, CPR or Flat pricing periods and can support the grid by imposing hard constraints such as putting an upper limit on maximum power consumption of individual dwelling. This approach can be easily extended for DR programs based on real time pricing (RTP).

The HEMS model defined by (4.23) allows the consumer to minimize the electricity bill by optimizing to a TOU pricing signal. In addition, it accommodates the additional incentive of a DR signal (either CPR or mandatory DR). Each HEMSC will broadcast its predicted day ahead energy consumption profile to aggregator, which in turn broadcasts the price signal to each HEMSC. Based on this price signal, each HEMS will optimize the energy consumption profile. For DR signals, the optimized results will be communicated back to aggregator, which will then send a yes or no flag by comparing baseline profile with optimized energy consumption profile. In case the flag is yes, HEMS will execute the solution and will wait for new DR strategy (Mandatory or Non-Mandatory) in case of No flag, which is based on distribution transformer loading or emergency power shutdowns. A flowchart illustration of HEMS operation in response to the various price signals is given in Figure 4.4, and a detailed description of the DR strategies is given below.

A non-mandatory DR strategy allows aggregators to encourage their customers to shift/curtail their energy consumption from high peak periods to low peak periods. Under non-mandatory DR strategies, the local DSO does not impose limits on power drawn from the grid. Instead, it uses DR signals such as critical peak rebate (CPR) as an incentive for customers to comply with the request. Customer can participate or opt out from DR request by setting the DR participation parameter δ_{dr}^{t}.

In case of system contingencies or unplanned shutdowns caused by problems in the power network, a mandatory DR event can be communicated to each HEMS, which will shift energy consumption away from the mandatory DR event's time range. The appliances are shifted based on pre-defined priorities set up by the customer. Furthermore, based on the mathematical model discussed in section 4.4, an upper hard restriction on maximum power drawn from the grid can be imposed as per eq. (4.24) on each dwelling to force customers to shift the energy consumption to off-peak periods.

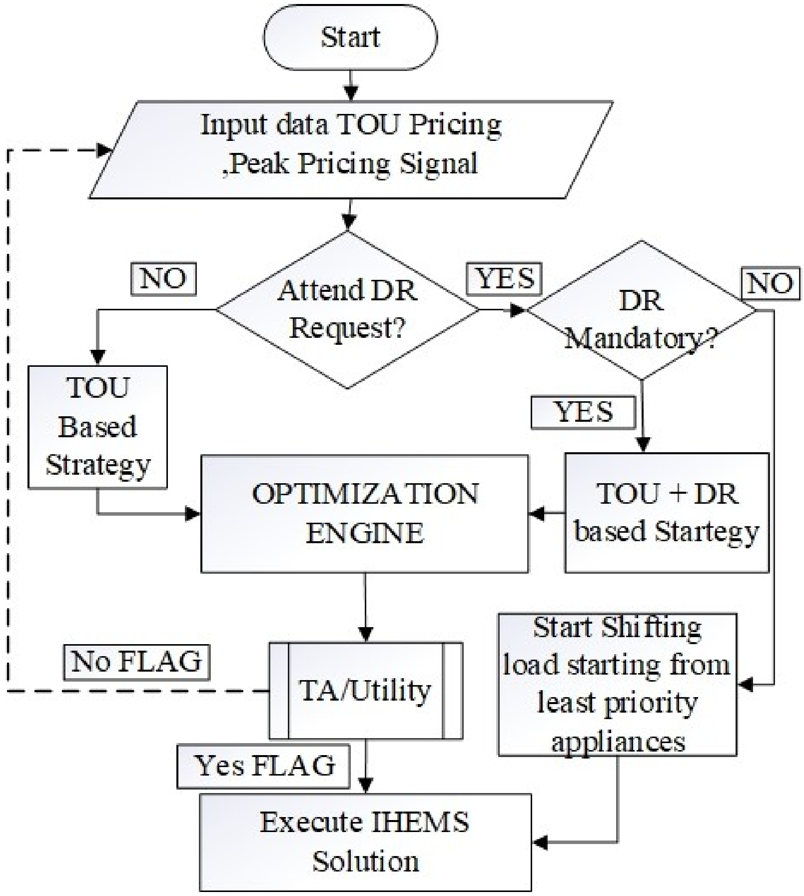

Figure 4.4 Flowchart for Load Reshaping under different scenarios

4.6. Energy Consumption Scheduling using MDPSO

The above optimization problem is a multi-objective knapsack combinatorial problem and is solved using our MDPSO (see section 3.5), with Elastic and Inelastic load classes optimized separately using our canonical and binary versions of PSO, respectively (see section 4.4). The particles move in a multi-dimensional search space to find the best distribution of energy consumption scheduling vectors for all appliances as depicted in Figure 4.5. The sub-swarms coordinate with each other by sharing common local best and global best positions, which means that appliances coordinate among themselves to minimize the fitness function. As an example, let us consider a number a of inelastic deferrable loads (uninterruptible) to be scheduled using binary MDPSO to minimize objective function (4.18) subject to constraints (4.11 – 4.14). A multi-dimensional swarm $S = (S_1, S_2, \dots, S_D)$ is initialized with a random uniform population as per section 3.1.3.

The size of swarm S is equal to number of appliances, D. Furthermore, swarm S consists of $n \in N_S$ particles.

Each particle $n \in N_S$ is represented by the decision vector $\overrightarrow{\boldsymbol{X}_n}$, which is to be optimized, where $\overrightarrow{\boldsymbol{X}_j} = \{\boldsymbol{X}_{j,1}, \boldsymbol{X}_{j,2}, \dots, \boldsymbol{X}_{j,a}\}$ is the j^{th} position vector in swarm S.In this particular optimization problem, $\overrightarrow{\boldsymbol{X}_j}$ is the set of binary scheduling vectors. Each sub-particle $\boldsymbol{X}_{j,a} \triangleq [x_{j,a}^{t_1}, x_{j,a}^{t_2}, \dots, x_{j,a}^{t_N}], \forall\, a \in I$ is a N dimensional vector representing the binary scheduling vector, and each element $x_{j,a}^{t_i} \in \boldsymbol{X}_{j,a}$ is the i^{th} element of $\boldsymbol{X}_{n,a}$ representing the appliance status in the t_i time slot. Note that $x_{j,a}^{t_i} = 0$ means "OFF" and $x_{j,a}^{t_i} = 1$ means "ON".

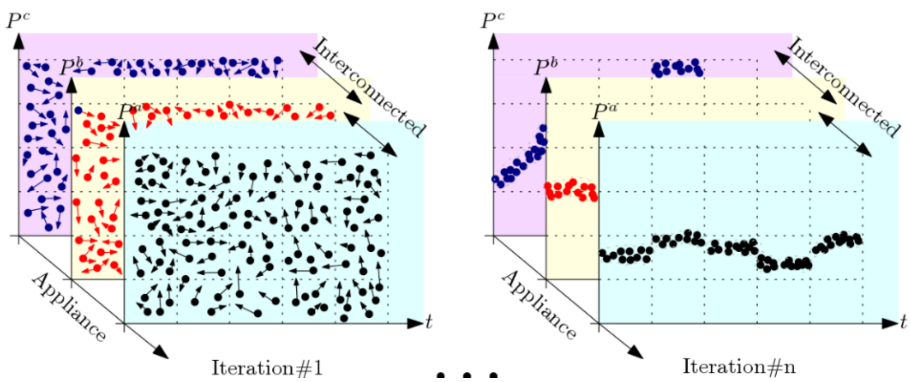

Figure 4.5 Illustration of energy consumption scheduling using MDPSO

The input to our MDPSO is the swarm of N_S position vectors $\overrightarrow{\boldsymbol{X}_n}$ having an associated velocity vectors $\overrightarrow{\boldsymbol{V}_n}$ and *local* best position vectors $\overrightarrow{\boldsymbol{P}_n^l}$, a D element array of N dimensional vectors. The global best position is represented as $\overrightarrow{\boldsymbol{P}^g}$, again a D element array of N dimensional vectors. At each iteration, the particle velocity is updated using the velocity update equations (3.26) and (3.18), and position update equation (3.19). $\overrightarrow{\boldsymbol{P}_n^l}$ is updated based on comparison of the local best cost at current and previous iterations. Furthermore, the global best position $\overrightarrow{\boldsymbol{P}^g}$ is updated based on comparison with the best solution achieved by swarm so far. For this appliance scheduling problem, the local best position and global best position updates are performed as below:

$$\overrightarrow{\boldsymbol{P}_n^l}(t+1) = \begin{cases} \overrightarrow{\boldsymbol{X}_n}(t+1), f\left(\overrightarrow{\boldsymbol{X}_n}(t+1)\right) < f\left(\overrightarrow{\boldsymbol{P}_n^l(t)}\right) \\ \overrightarrow{\boldsymbol{X}_n}(t), \quad f\left(\overrightarrow{\boldsymbol{X}_n}(t+1)\right) \geq f\left(\overrightarrow{\boldsymbol{P}_n^l(t)}\right) \end{cases}, \tag{4.24}$$

$$\overrightarrow{\boldsymbol{P^g}}\,(t+1) = \begin{cases} \overrightarrow{\boldsymbol{P^l_n}}(t+1), f\left(\overrightarrow{\boldsymbol{P}^l_n(t+1)}\right) < \; f(\overrightarrow{\boldsymbol{P^g}}\,(t)) \\ \overrightarrow{\boldsymbol{P}^l_n(t)}, \qquad f\left(\overrightarrow{\boldsymbol{P}^l_n(t+1)}\right) \geq f(\overrightarrow{\boldsymbol{P^g}}\,(t)) \end{cases}. \tag{4.25}$$

The global best value $\overrightarrow{\boldsymbol{P^g}}$ is here the output of our binary MDPSO, and gives the binary decision vector (that is, the scheduling) for the inelastic class of appliances.

A similar approach is taken to schedule the elastic appliances having continuous decision variables with velocity and position updates performed using equations 3.26 and 3.27 respectively. Figures 4.6 and 4.7 provide very brief pseudocode descriptions of the MDPSO implementation for inelastic deferrable loads and elastic Non-Deferrable appliances, respectively, as a summary of the above discussions.

Algorithm 4.1 MDPSO: Inelastic Appliances

Input: Randomly Create and Initialize $\overrightarrow{\boldsymbol{X_n}}\,\overrightarrow{\boldsymbol{P^l_n}}\,, \overrightarrow{\boldsymbol{V_n}}, \overrightarrow{\boldsymbol{P^g}}$ of size N_S.

Set constant parameters $w_i, w_f, c_{1max}, c_{2max}, c_{1min}, c_{2min}, \delta, x^{t_N}_{max}, x^{t_N}_{min}, t_{max}$

Output: Global best position $\overrightarrow{\boldsymbol{P^g}}$

for $t \; = \; 0 \;\; to \; t_{max}$

Compute $c_1(t), c_2(t)$ & $w(t)$ as per eq. (3.8 – 3.10)

for each particle $n \in \; S$ do

Compute the Fitness function (4.22)

Update $\overrightarrow{\boldsymbol{P^l_n}}, \overrightarrow{\boldsymbol{P^g}}$ as per eq. (4.24 - 4.25)

Update Velocity $\overrightarrow{\boldsymbol{V_n}}$ as per eq. (3.26)

Check upper and lower limit violations $\forall \, v^{S_D}_{n,N} \in \;\; \boldsymbol{v^{S_D}_{n,N}}$ as per eq. (3.6)

Normalize Position elements $x^{S_D}_{n,N} \in \; \boldsymbol{x^{S_D}_n}$ as per eq. (3.18)

Update Position elements $x^{S_D}_{n,N}$ as per eq. (3.19)

Check appliances are ON during permissible operating time window.

End

End

Figure 4.6 Pseudo Code for scheduling Inelastic Appliances

Algorithm 4.2 MDPSO: Elastic Appliances

Input: Randomly Create and Initialize $\overrightarrow{\boldsymbol{J_n}}\, \overrightarrow{\boldsymbol{P_n^l}}, \overrightarrow{\boldsymbol{V_n}}, \overrightarrow{\boldsymbol{P^g}}$ of size N_S.

Set constant parameters $w_i, w_f, c_{1max}, c_{2max}, c_{1min}, c_{2min}, \delta, j_{max}^{t_N}, j_{min}^{t_N}, t_{max}$

Output: Global best position $\overrightarrow{\boldsymbol{P^g}}$

for $t = 0 \;\; to\; t_{max}$

Compute $c_1(t), c_2(t)$ & $w(t)$ as per eq. (3.8 – 3.10)

for each particle $n \in S$ do

Compute the Fitness function $f\left(\overrightarrow{\boldsymbol{J_n}}\right)$ (4.22)

Update $\overrightarrow{\boldsymbol{P_n^l}}, \overrightarrow{\boldsymbol{P^g}}$ as per eq. (4.24 - 4.25)

Update Velocity $\overrightarrow{\boldsymbol{V_n}}$ as per eq. (3.26)

Check upper and lower limit violations $\forall\, v_{n,N}^{S_D} \in \boldsymbol{v_{n,N}^{S_D}}$ as per eq. (3.6)

Update Position elements $j_{n,N}^{S_D}$ as per eq. (3.7)

Check appliances energy consumption is with in limits i.e. $j_{n,N}^{S_D} \in \{ j_{max}^{t_N}, j_{min}^{t_N} \}$.

End

End

Figure 4.7 Pseudo Code for scheduling Elastic Appliances

5. VALIDATION OF PROPOSED APPROACH

This chapter will investigate the independent scheduling of Inelastic (I) and Elastic (E) appliances under TOU pricing, and Mandatory DR and Non-Mandatory DR strategies, in order to validate their optimization problem formulations and to evaluate MDPSO performance. Three different customer comfort scenarios are considered for use case purposes: high comfort, $w_1 = 0.8\,, w_2 = 0.2$; medium comfort, $w_1 = 0.4\,, w_2 = 0.6$; and low comfort, $w_2 = 0.0\,, w_2 = 1.0$. Under high comfort conditions, objective function (4.23) is more biased towards customer comfort, while under low comfort conditions objective function (4.23) optimization is performed to minimize electricity cost. We compare the energy cost deviation between the baseline and scheduled cases under these comfort levels. The sensitivity of the algorithm to time and space is also presented. It should be noted that PSO, as a stochastically seeded optimization process, can produce a different set of decision vectors from run to run. We therefore run the optimization multiple times, taking as our scheduled result that output which has the minimum daily cost, with scheduling times as close to the nominal (*i.e.*, the original, or unscheduled, or preferred) as possible.

5.1. Use Case Analysis

5.1.1. Inelastic Appliances

We consider three appliances from the Inelastic interruptible class (I_{in}), which we simply label as *inelastic*, and two appliances from I_{in} plus two from the Inelastic Uninterruptible class (I_{un}), which we label as *mixed inelastic* . We consider these two groups for use case simulations. We consider the scheduling of loads within a 24-hour time period from 7AM to 7AM, divided into 96 equal time intervals of 15 minutes each. Table 5.1 and Table 5.2 list the rated power, profile length (PL) and length of operating window (τ_a), for fixed and mixed appliances, respectively. The aggregate load profiles are generated from these. The profile length (PL) is defined by number of timeslots required by appliance to finish to finish its duty cycle. We assume that the appliances operate at their rated power when turned 'ON'. We adopted summer TOU prices from Hydro Ottawa [52]. The low peak, mid-peak, and on-peak prices are 0.065$/kWh, 0.094 $/kWh and 0.135 $/kWh, respectively. The input parameters for scheduling inelastic appliances using our binary version of PSO are listed in Table 5.3.

Table 5-1 Inelastic Interruptible Loads

No	Appliance	kW	PL	τ_a
1	Dehumidifier	0.6	6	1 - 96
2	Well pump	1.4	6	1 - 96
3	Sump pump	1.6	8	1 - 96

Table 5-2 Mixed Inelastic Appliances Data

No	Appliance	kW	PL	τ_a
1	Clothes Dryer (I_{in})	3.5	4	1 - 96
2	Motor (I_{in})	2.1	8	1 - 96
3	Dishwasher (I_{un})	1.3	6	1 - 96
4	Washing Machine (I_{un})	0.5	2	1 - 96

Table 5-3 Initial Parameters for discrete PSO

Parameters	**Value**
c_{1max}, c_{2max}	2.5,
c_{1min}, c_{2min}	0.5
v_{max}, v_{min}	4, -4
Iterations	1000
Swarm Size	80
w_i, w_f	0.9,0.4

5.1.1.1. TOU Pricing

Under TOU pricing, appliances are scheduled to minimize cost by shifting the demand to low price periods. Hence, a value of $\theta_{dr} = 0$ is used in (4.19) to turn off the DR functionality in (4.23). A penalty function coefficient of $\lambda = 10$ is used, and the length of optimization window τ_a is set equal to 24 hours. This means that appliances can be scheduled to any time of the day. To remove it from consideration, we take the distribution panel service limit to be greater than the sum of all appliance consumption. We nominally operate the three inelastic (I_{in}) loads on-peak, as shown by the solid "unscheduled" consumption profiles in Figure 5.1. The TOU price signal is shown on the right axis, and unscheduled and scheduled appliance consumption profiles on the left axis. This will be the convention for all subsequent figures. The scheduling has been done for a low comfort level, which is little constrained by operational inconvenience. We find that all profiles have been moved to the lowest TOU region, with complete interruption diversity – every appliance operation is interrupted after the minimum 15 minute PL, and these periods are randomly dispersed for easy appliance throughout the lowest TOU region. This may be contrasted with the medium and high comfort conditions, as shown in Figures 5.2 and Figure 5.3, respectively. Now there is increasing comfort bias to scheduling convenience as the optimization must trade off cost and the need to start and finish appliance operation as close to the nominal scenario as possible. In Figure 5.2, appliance 1 shows full interruption diversity within the lowest TOU region, but appliances 2 and 3 find part of their operation within their nominal operation times. In Figure 5.3, this is even more apparent, with appliance 2 unmoved, only one PL period from appliance 3, and 2 from appliance 1, now scheduled within the lowest TOU region. The scheduled cost is 0.40\$/day, 0.65\$/day and 0.73\$/day under low, medium and high comfort levels, respectively, as compared to the nominal cost of 0.83 \$/day. This validates that the effectiveness of the algorithm in terms of scheduling similar appliances across the full optimization period, the ability of PSO to find the optimal solution for a multi-objective problem, and the reliability of the global best optimum solution.

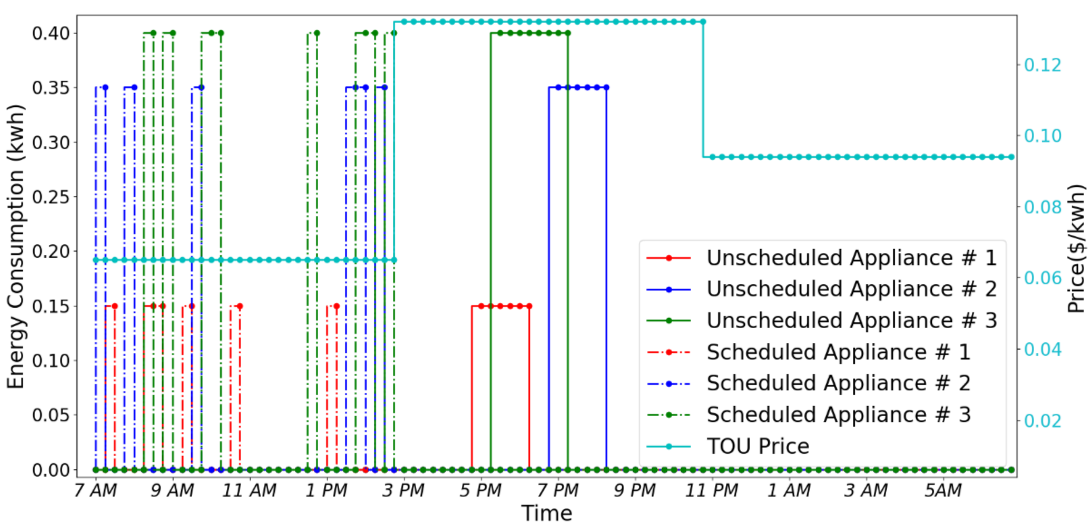

Figure 5.1 Inelastic appliances scheduling under TOU pricing and low comfort level

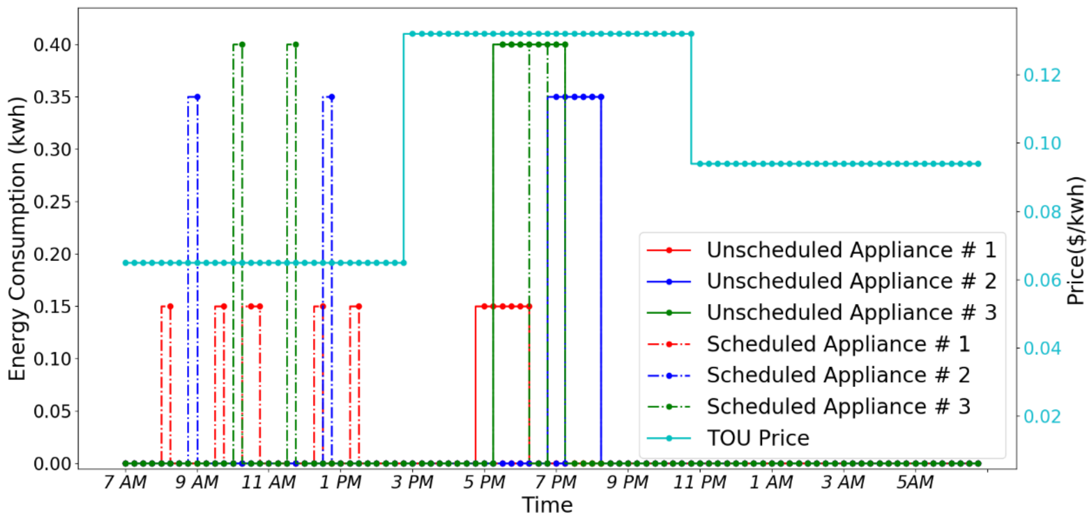

Figure 5.2 Inelastic appliances scheduling under TOU and medium comfort level

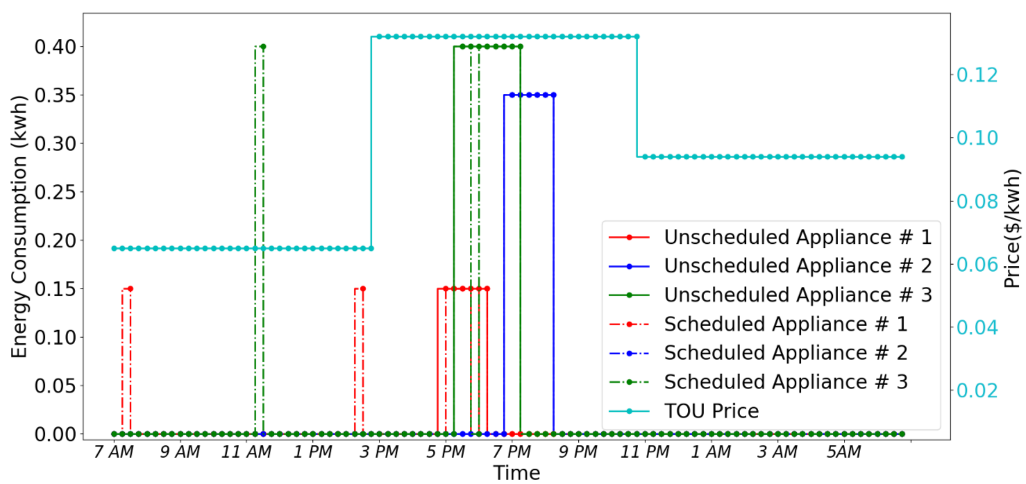

Figure 5.3 Inelastic appliances scheduling under TOU and high comfort level

Figures 5.4 – 5.6 demonstrates the scheduling of the mixed appliances under TOU pricing. Note that the nominal position of appliance 2 (in blue) is within the low TOU region, but that the other three appliances are within the peak TOU region. The low comfort situation of Figure 5.4 shows, as before, all appliances scheduled to operate within the low TOU period, the interruptible appliances (red and blue) again showing profile segmentation into minimal operating periods with full scheduling diversity. The uninterruptible appliances (green and yellow) have no profile segmentation, demonstrating constraint satisfaction. Under medium comfort conditions, Figure 5.5 shows that scheduled operation of interruptible appliance 2 (blue) remains unchanged (no movement being the easiest comfort/cost tradeoff to achieve, so also no profile segmentation), that of uninterruptible appliance 4 (yellow) is only slightly shifted, while appliances 1 and 3 are scheduled to operate in the low TOU region. Under high comfort conditions, Figure 5.6 shows no or little change in all but the smallest uninterruptible appliance 3 (green), placing it within the low TOU region. Some scheduling diversity of appliance two is now noted, even though there is no cost advantage. This is due to the stochastic initialization of the swarm. The scheduled electricity cost is 0.64$/day, 0.77$/day and 1.0$/day under low, medium and high comfort conditions as compared to unscheduled electricity cost of 1.1$/day. This verifies the ability of proposed algorithm for scheduling mixed appliances under cost minimization objective as well as finding optimal trade off solution between cost and comfort objectives.

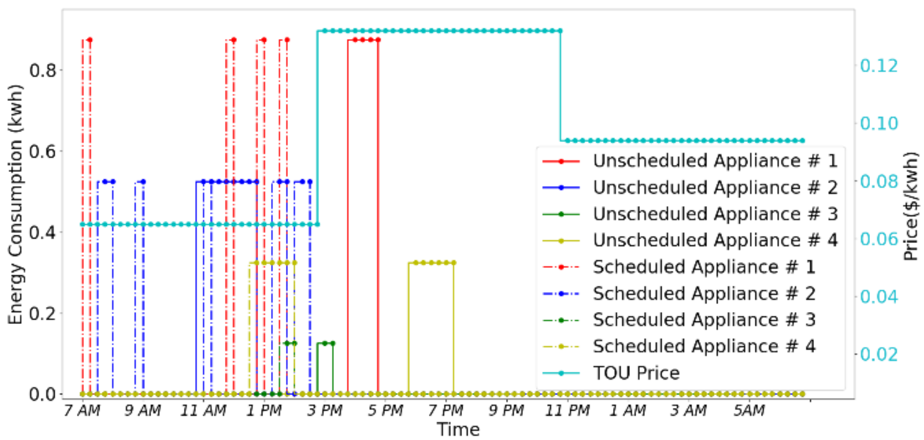

Figure 5.4 Mixed Inelastic Appliances scheduling under TOU and low comfort level

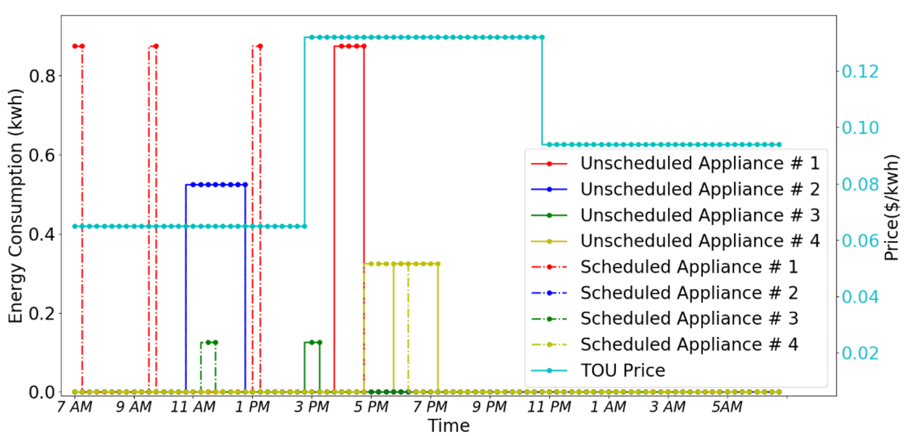

Figure 5.5 Mixed Inelastic Appliances scheduling under TOU and Medium comfort level

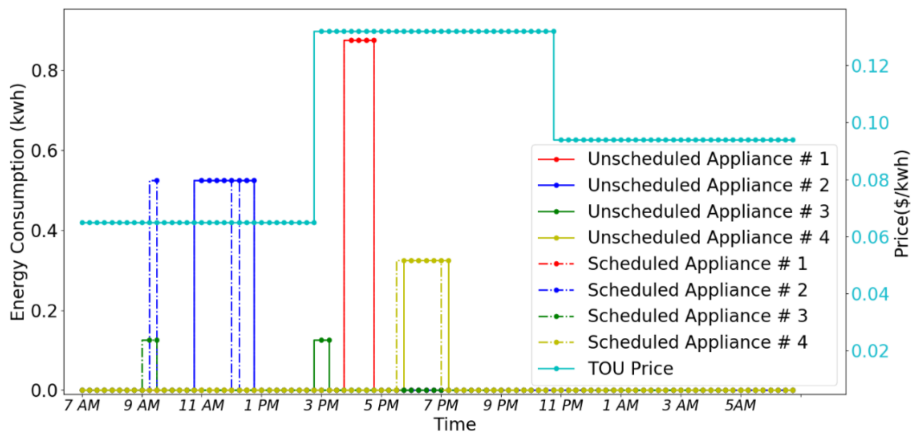

Figure 5.6 Mixed Inelastic Appliances scheduling under TOU and high comfort level

5.1.1.2. Mandatory DR Strategy

Under a mandatory DR event, appliances are forced away from their optimal TOU scheduling scenario by hard constraints on maximum consumption in each time slot. This use case is performed to validate the optimization problem formulation and to prove the reliability of MDPSO to optimally redistribute energy consumption under different conditions. The input to the optimization engine is a nominal load profile of mixed appliances; ψ_r = 1 kWh – the hard constraint, see (4.14); θ_{dr} = 0 – see (4.19); and λ = 10 – see (4.23). The length of the optimization window is 24 hours. Figures 5.7-5.9 compare the nominal and scheduled consumption profiles of the mixed appliances under low, medium and high comfort conditions. Again, solid coloured lines are the nominal (unscheduled) profiles, and dashed the scheduled profiles. Comfort constraint behaviour similar to that described above is seen, with the impact of the consumption constraint demonstrated by the black aggregate profiles. Under no circumstances does the consumption exceed the limit, demonstrating the efficacy of the optimization under this additional constraint. The scheduled costs are 0.64$/day, 0.77$/day and 1.0 $/day under low, medium and high comfort conditions as compared to a unscheduled cost of 1.1 $/day.

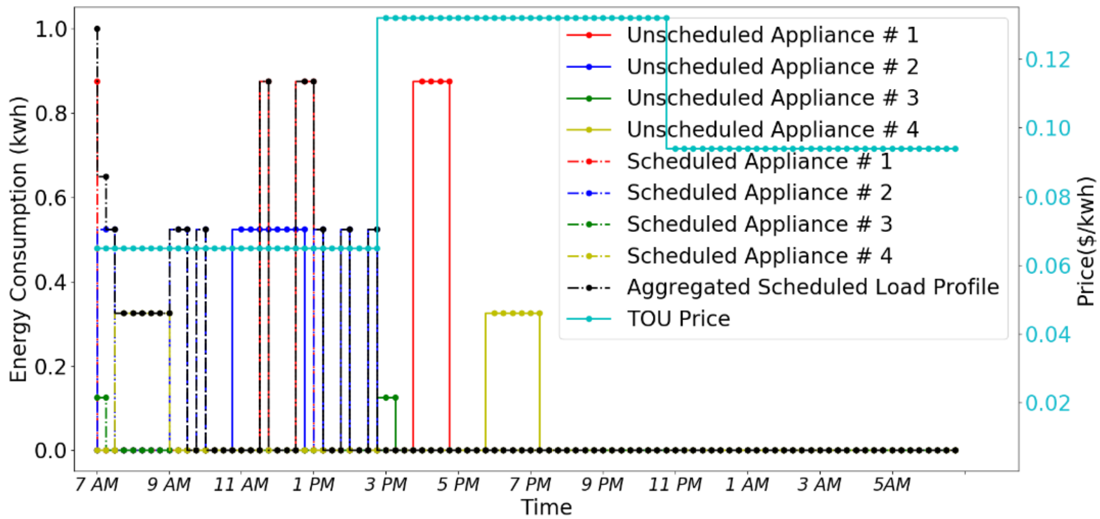

Figure 5.7 Mixed Inelastic appliances scheduling under TOU + DR event (upper power threshold) and low comfort level

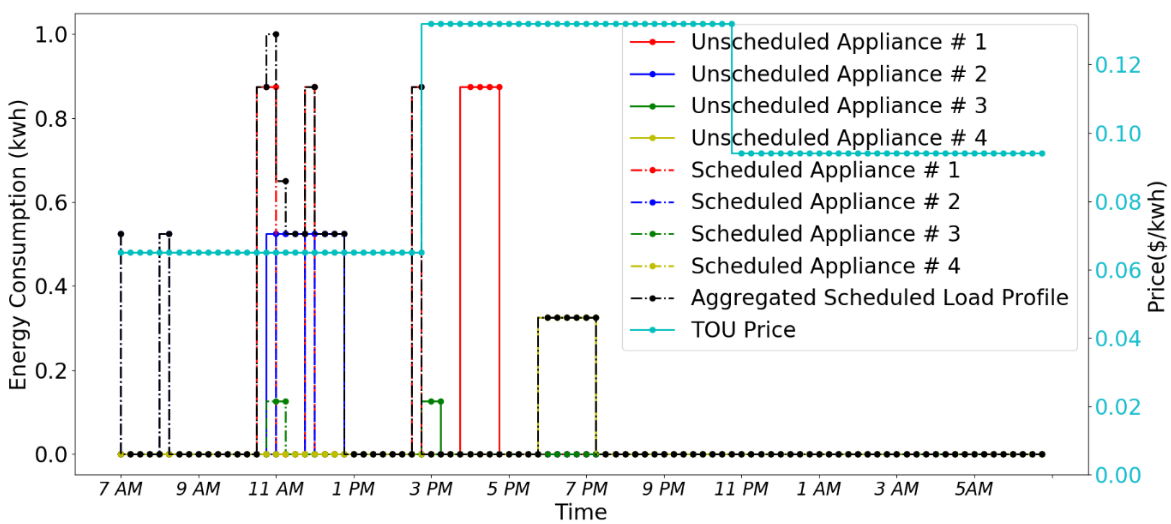

Figure 5.8 Mixed Inelastic appliances scheduling under TOU + DR event (upper power threshold) and medium comfort level

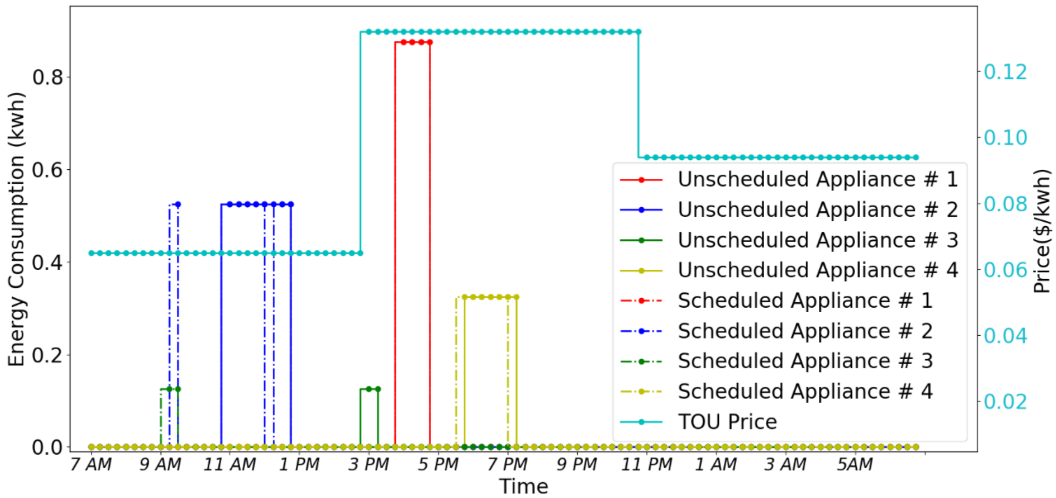

Figure 5.9 Mixed Inelastic appliances scheduling under TOU + DR event (upper power threshold) and high comfort level

5.1.1.3. Non-Mandatory DR Strategy

As discussed in section 4.5, under this strategy the appliances are scheduled under a combination of TOU and CPR signals. We again consider the scheduling of mixed appliances, using $\theta_{dr} = 1$ and a CPR of $\delta_{dr}^{t} = 0.15$ \$/kWh over a select time interval (noted below) sufficient to provide a notable impact. Operationally, this value is determined by the aggregator from more advanced fiscal models. In Figure 5.10, we show the impact of a DR request during a low TOU price period, setting the CPR to override the TOU pricing from 7am to 3pm. The nominal appliance operation is again shown by the solid profiles, which are now dispersed in the low and high TOU periods. The optimization drives the scheduling to the medium TOU region, since it is now the lowest price period. In Figure 5.11, we show the impact of a DR request during the high TOU price period, under medium comfort conditions, setting the CPR to override the TOU pricing from 3pm to 11pm. The nominal appliance operation is again shown by the solid profiles, which are in the same time periods as above. The increased price signal in the high TOU period drives scheduling of all appliances into the low price period, whereas previously the medium comfort constraint permitted some appliance operation in the high price period (*cf.* Figure 5.5). In Figure 5.12, we also show the impact of a DR request during the high TOU price period, under high comfort conditions, again setting the CPR to override the TOU pricing from 3pm to 11pm. With this strong competition between cost and comfort, appliance 1 is scheduled at its nominal time (the profiles overlap) in the high price period, while all other appliances again move to the lowest price period. We conclude the CPR can be used to force comfort constraint elasticity. Compared to an unscheduled cost of 1.1 $/day, the scheduled energy costs are 0.30 $/day, 0.38$/day and 0.58$/day, under low, medium and high comfort, respectively.

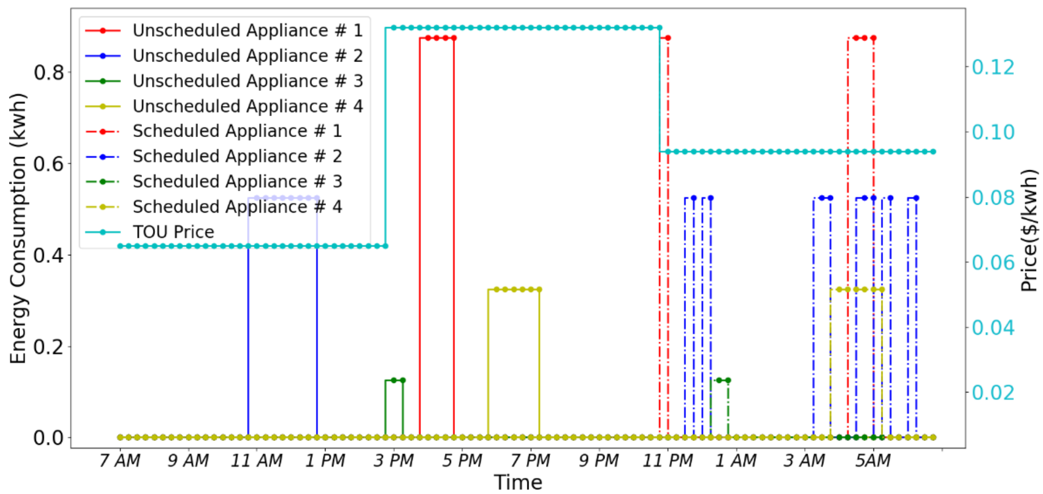

Figure 5.10 Mixed Inelastic Appliances Scheduling under TOU + CPR and low comfort level

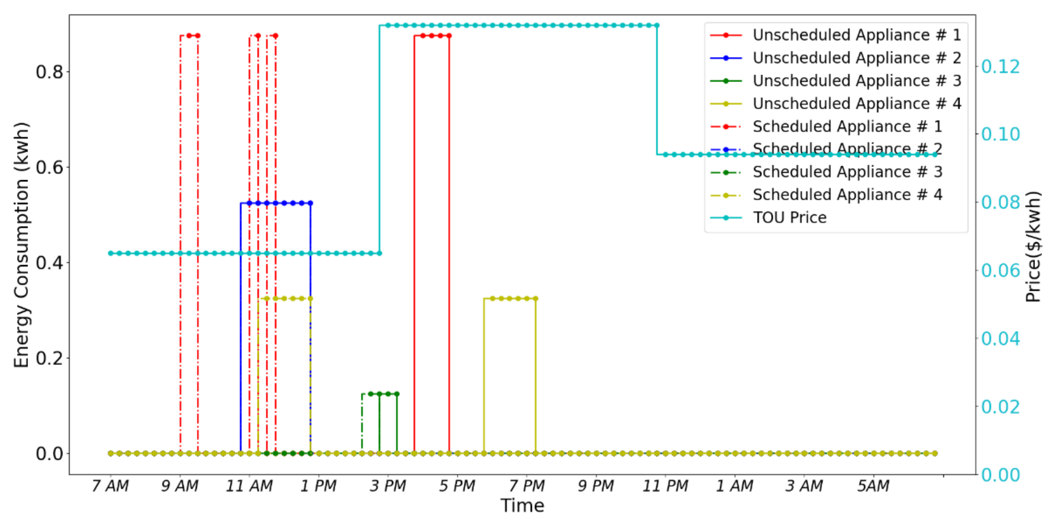

Figure 5.11 Mixed Inelastic Appliances Scheduling under TOU + CPR and medium comfort level

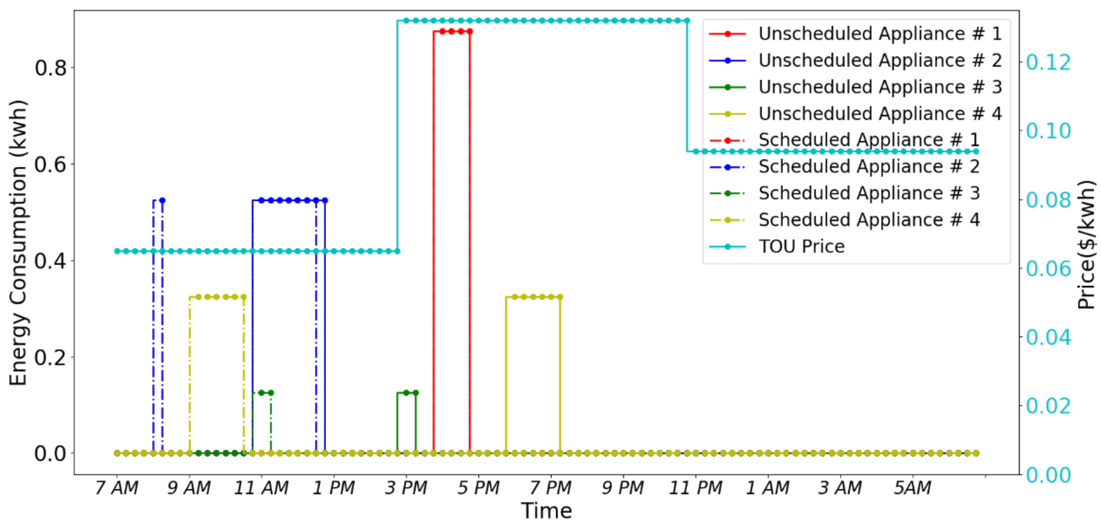

Figure 5.12 Mixed Inelastic Appliances Scheduling under TOU + CPR and high comfort level

5.1.2. **Elastic Appliances**

We now perform use case simulations to optimally schedule elastic appliances. The simulations are again performed under the comfort conditions and load shaping strategies discussed in section 4.5. To remove it from consideration, we take the distribution panel service limit to be greater than the sum of all appliance consumption. We consider three appliances from elastic load class 1 (E_{c_1}), which we will simply refer to as *elastic* appliances for convenience. Table 5.4 lists their maximum rated power consumption, nominal profile length (PL), and permissible operating time periods (τ_a). Their total energy consumptions are fixed, and can be determined from their nominal profiles. We impose no hard operating constraints in each time interval (*i.e.,* minimum consumption is zero and maximum consumption is the distribution panel service limit). We also consider a group of three appliances, one each from elastic load class 1 (E_{c_1}), elastic load class 2 (E_{c_2}), and elastic load class 3 (E_{c_3}), which we referred to as *mixed elastic* appliances. Table 5.5 lists their maximum rated power consumption, nominal profile lengths (PL) and permissible operating time periods (τ_a). The load class 1 appliance is constrained as noted above. The load class 2 appliance has no fixed total energy consumption, but we impose a minimum consumption of 0.05 kW for operation in any time interval; the maximum is again the service limit. PL is computed from the number of time slots required by the appliance to complete its duty cycle, and τ_a is set as noted. The load class 3 appliance is a thermostatic load, whose thermal model is given in section 4.3.2.3. The following parameters are used for computing the indoor temperature using (4.9): $\theta_t^{out} = 85$ °F, $\theta^d = 75$, $\theta^e = 15, ⅂ = 0.25\ h$, χ = -5, and $\delta = 25$. Indoor minimum and maximum temperatures, respectively $\theta_t^{a,min} = 68$ °F and $\theta_t^{a,max} = 72$ °F, are used. The TOU pricing signal remains the same as in section 5.2. The input parameters to the canonical PSO are given in Table 5.6.

Table 5-4 Elastic Appliance data for use case studies

No	Appliance	kW	PL	τ_a	kWh/day
1	PHEV - 1	3.3	4	1 - 96	2.175
2	Battery Charger	2	20	1 - 96	6.6
3	Battery Charger-2	3.3	26	1 - 96	9.715

Table 5-5 Mixed Elastic Appliance data for use case

No	Appliance	kW	PL	τ_a
1	PHEV (E_{c_1})	3.3	4	1 – 65 (7AM -11:45PM)
2	Small Load (E_{c_2})	0.4	7	88 – 96 (5AM-7AM)
3	AC (E_{c_3})	1.5	40	20 – 60 (12 Noon – 10 PM)

Table 5-6 Initial Parameters for PSO

Parameters	Value
$c_{1\max}, c_{2\max}$	2.5, 2.5
$c_{1\min}, c_{2\min}$	0.5, 0.5
Iterations	1000
Swarm Size	80
w_i, w_f	0.9, 0.4
δ_{vmax}	0.02

5.1.2.1. TOU Pricing

We simulate the basic use case to schedule the *elastic* (E_{c_1}) appliances based on TOU pricing and consider different comfort conditions. The objective function (4.23) is optimized with $\theta_{dr} = 0$ in (4.18) to turn off DR functionality. A penalty function coefficient of $\lambda = 10$ is used. The scheduling time window is 24 hours. In the following figures, the nominal consumption profiles of each appliance are shown as "unscheduled" solid lines (placed predominantly in the high TOU period), while the scheduled profiles are dashed lines. Figure 5.13 shows the scheduling under low comfort conditions. Optimization randomly schedules (via the stochastic initialization of the swarm) the appliance profiles throughout the low-price periods, since the comfort constraint is not great enough to keep any operation in the high TOU region. Figure 5.14 shows the scheduling under medium comfort conditions. A significant proportion of appliance operation now occurs throughout the low TOU period. Most of appliance 1 is in the low TOU region, but a small component is in the medium TOU period. This latter, which clearly would be better in the low TOU region, is a consequence of the stochastic initialization of the swarm and the optimization engine termination conditions. Appliances 2 and 3 both have a rough balance in scheduling consumption between the low TOU period and the "preferred" nominal time. Thus, the competition between medium comfort and TOU pricing is roughly even. By contrast, Figure 5.15 shows the scheduling under high comfort conditions. Far less appliance operation is scheduled throughout the low TOU period, with more around the "preferred" nominal time. Thus, the competition between high comfort and TOU pricing is strongly biased toward comfort. There is again, to a similar degree, appliance 1 operation in the medium TOU period, for the reason noted above. Operation of appliances 2 and 3 in the medium TOU period is due to the dominance of the comfort constraint over price, as the scheduling lies close to the nominal. The scheduled electricity cost is 0.60 \$/day, 1.70\$/day and 2.0 \$/day under low, medium and high comfort levels, respectively, as compared to unscheduled cost of 2.34\$/day.

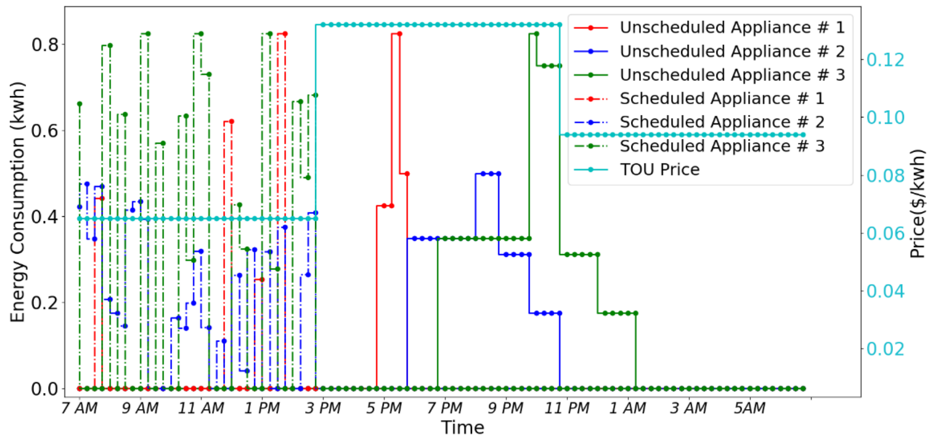

Figure 5.13 Elastic Appliance scheduling under TOU and low comfort level

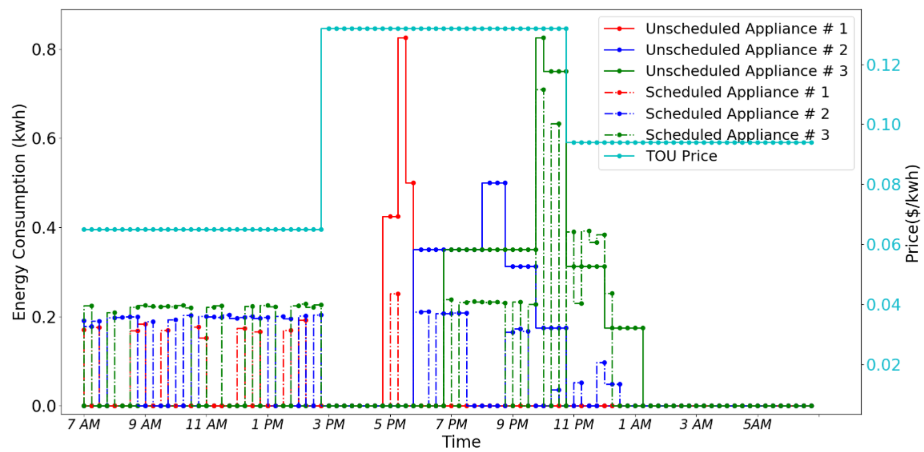

Figure 5.14 Elastic Appliance Scheduling under TOU and medium comfort level

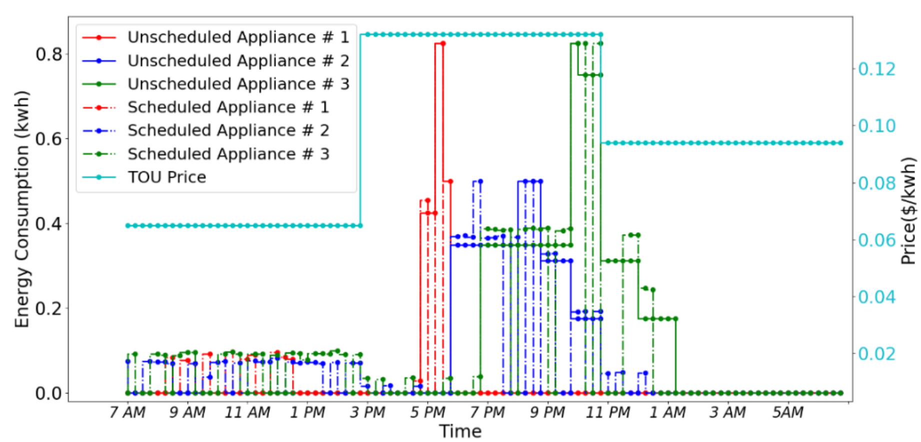

Figure 5.15 Elastic Appliance scheduling under TOU and high comfort level

5.1.2.2. Mandatory DR Strategy

This use case presents the scheduling of *mixed elastic* appliances under TOU pricing and upper power consumption limit of $\psi_r = 0.8$ kWh. We considered scheduling appliances under constrained operating windows (τ_a) as given in Table 5.5. The value of DR activation parameter $\theta_{dr} = 0$ is used in fitness function (4.23). As shown in Figure 5.16, under nominal conditions, appliance 1 operates during the low TOU period (Red), very small appliance 2 operates in the low to high TOU periods (flat profile), and appliance 3 operates between high and low TOU periods. Optimizing scheduling under the low comfort condition sees: appliance 1 still operating in the low TOU period (but now dispersed, lessening its peak); appliance 2 shift its consumption into the permissible operating period, obeying minimum consumption requirements; and appliance 3 disperses its consumption through the permissible operating period, with a slight bias toward the low TOU period, as constrained by the minimum and maximum indoor temperature requirements. The indoor temperature variations (blue) for nominal and scheduled AC (appliance 3) consumption are shown in Figures 5.17 and 5.18, respectively. The temperature drops are seen to correlate with AC operation, as shown by the red profiles. consumption profiles of the air conditioner (AC). The bias toward low TOU operation forces temperatures toward the minimum during that period (see Figure 5.18). Sufficient energy is allocated to the AC to keep the house below the maximum allowable temperature during the permissible operating period, but not to sufficiently pre-cool during this period to keep the indoor temperature below the maximum during the early morning hours. These observations validate the implementation of the thermal model (4.9). Under medium and high comfort conditions, appliances are scheduled more closely with respect to their nominal operation times, as shown in Figure 5.19 and 5.20, respectively. The trends are as we have seen in the other use cases. In all cases, the aggregate profile (black) is within the defined limit of 0.8 kWh. Scheduled daily costs are \$0.50, \$0.60, and \$0.72 under low, medium and high comfort constraints, respectively, as compared to a nominal daily cost of \$0.85.

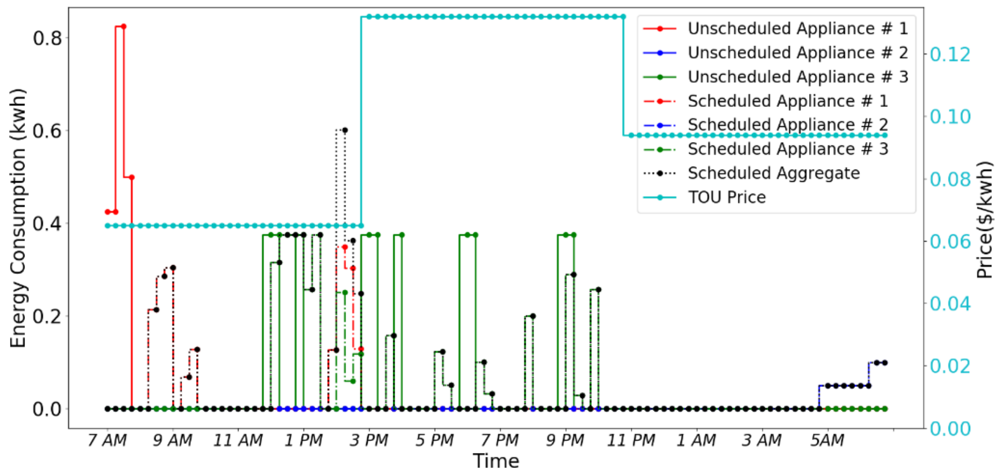

Figure 5.16 Mixed Elastic Appliances Scheduling under TOU + DR (Power Threshold) and minimum comfort level

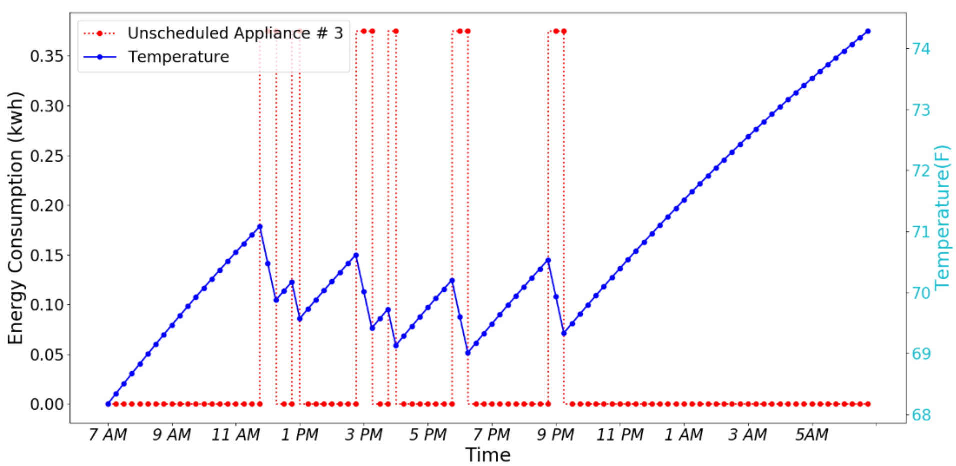

Figure 5.17 Unscheduled AC Energy consumption of Fig. 5.16 vs temperature

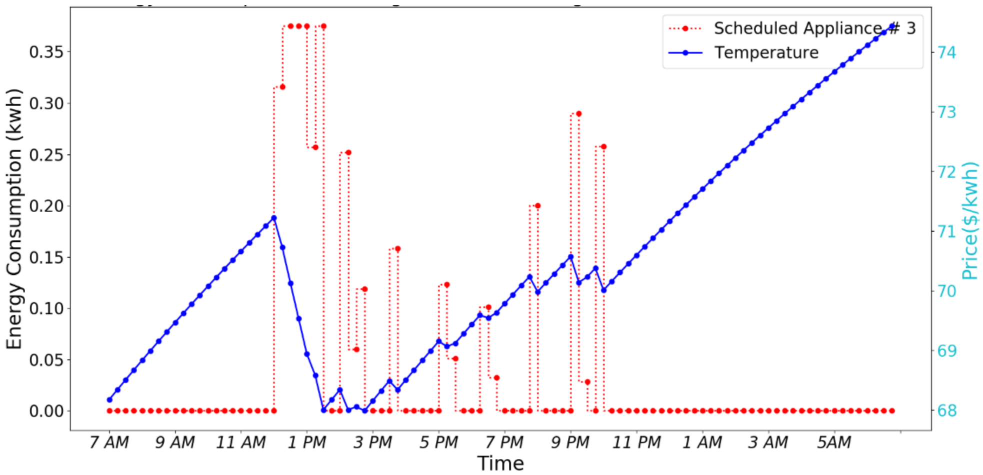

Figure 5.18 Scheduled AC Energy consumption of Fig. 5.16 vs temperature

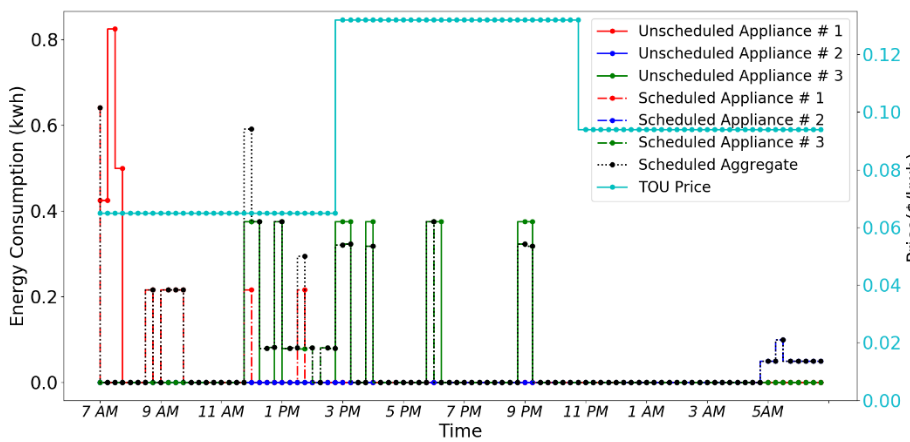

Figure 5.19 Mixed Elastic Appliances Scheduling under TOU + DR (Power Threshold) and medium comfort level

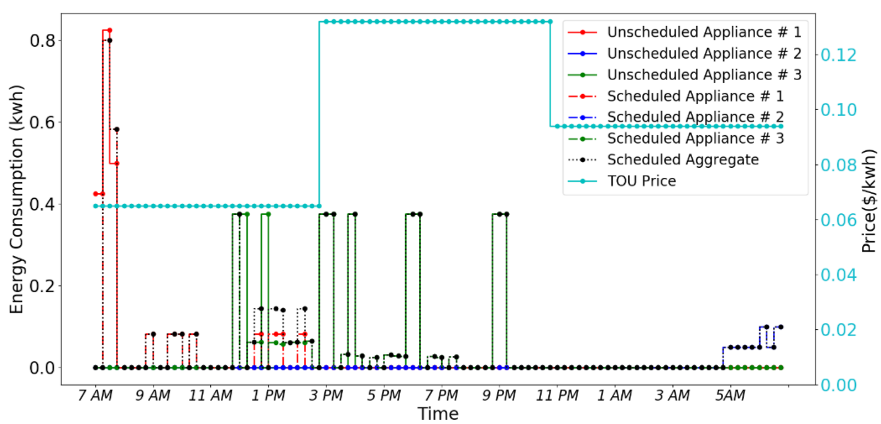

Figure 5.20 Mixed Elastic Appliances Scheduling under TOU and DR (Power Threshold) and maximum comfort level

5.1.2.3. Non-Mandatory DR Strategy

Under a Non-Mandatory DR request, the appliances are scheduled under a combination of TOU and CPR signals. A CPR signal, given by δ_{dr}^{t} =0.15\$/kWh and $\theta_{\mathrm{dr}} = 1$ as before, is now used in (4.20). The DR request is from 7AM to 9:00 AM. All other inputs are as described previously. Figure 5.21 presents the CPR and TOU impact on mixed elastic appliance scheduling under a low comfort constraint. The CPR signal shifts appliance 1 operation out of the CPR period, but still within the low TOU period. The scheduling behaviour of the other appliances is consistent with previous descriptions. Figure 5.22 presents the CPR and TOU impact on mixed elastic appliance scheduling under a medium comfort constraint. Appliance 1 scheduling is still out of the CPR period, but within the low TOU period. A comfort bias to be closer to the nominal profile is not apparent – behaviour appears similar to the low comfort case. Figure 5.23 presents the CPR and TOU impact on mixed elastic appliance scheduling under a high comfort constraint. Appliance 1 scheduling now also occurs within the CPR period and dispersed within the low TOU period.

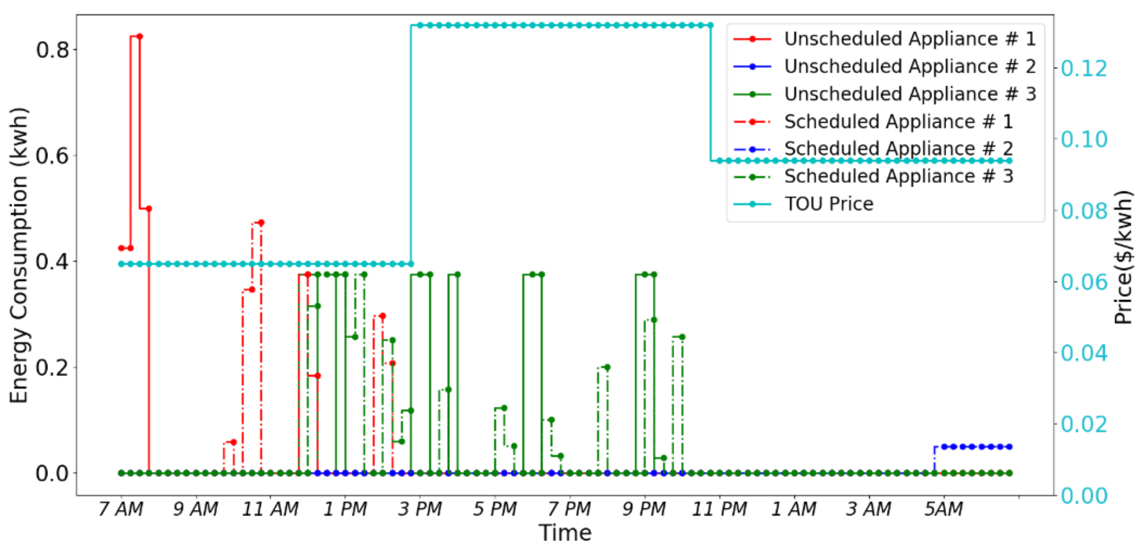

Figure 5.21 Mixed Elastic Appliances Scheduling under TOU + CPR and low comfort level

Now a strong comfort bias to be closer to the nominal profile is apparent. This validates the optimization process. The scheduled daily energy costs are \$0.30, \$0.45, and \$0.70, for low, medium and high comfort constraints, respectively, as compared to the unscheduled daily cost of $0.85.

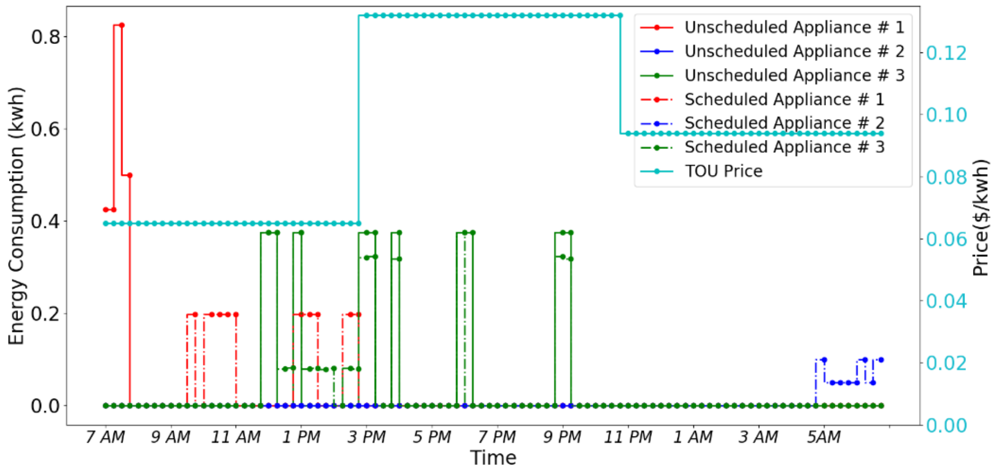

Figure 5.22 Mixed Elastic Appliances Scheduling under TOU + CPR and medium comfort level

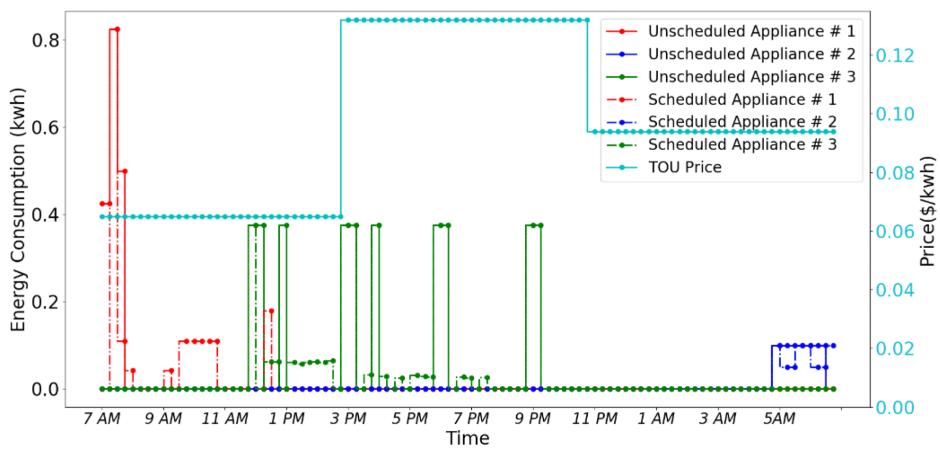

Figure 5.23 Mixed Elastic Appliances Scheduling under TOU + CPR and high comfort level

5.1. Convergence Analysis and Scaling

We will now validate the performance and efficiency of MDPSO to solve the energy consumption scheduling problem with respect to the length of decision vectors and the number of appliances. The performance of PSO is computed as deviation of achieved global best cost from optimum fitness value. This requires prior knowledge of optimal fitness function value; hence we have considered a TOU pricing scenario where appliances are scheduled over full time horizon. We can compute the optimal fitness function value for each case by computing the electricity price for operating appliances in low price periods. We will be testing the algorithm for scheduling *elastic* E_{C_1} appliances and *inelastic* I_{in} appliances, across three different lengths of decision vector (288 time slots ,144 time slots and 96 time slots).The number of time slots are increased by reducing the length of time interval t i.e. 24 hours into 5 min of equal length result in 288 timeslots. Furthermore, sensitivity analysis is performed under increasing number of $elastic$ and $inelastic$ appliances from 5 to 20 , keeping the length of operating time window τ_a equal to 96 time slots. Parameters for velocity update equations for binary and canonical version of PSO are shown in Table 5.3 and 5.6 (see section 5.1 and 5.2) respectively.

5.1.1. Length of Time Interval

For sensitivity analysis, we have considered scheduling of E_{C_1} and I_{in} appliances separately under TOU pricing and varying length of time interval t. The number of particles in each sub swarm and maximum iteration count is set to 100 and 1000 respectively for each simulation. The input load data for *inelastic* appliances and *elastic* appliances is shown in Table 5.1 and 5.4 respectively. The upper limit for power consumption in each time slot is taken as 1.6 KW for inelastic appliances scheduling and 6 KW for elastic appliances scheduling. For *inelastic* appliances, figure 5.24 shows the convergence of *global* best cost with each iteration under different values of time interval t. It is observed that number of iterations required to reach the optimal solution increases linearly with increase in number of time slots. In case, the number of time slots are set to 96 the algorithm converges to optimal solution in 242 iterations. The number of iterations increase to 335 and 522 with increase in number of time slots to 144 and 288 respectively.While the proposed framework generates optimal solution even for smaller time interval lengths, its impact on optimal cost is not obvious. Each simulation under different time interval lengths produced a cost saving of 44.3% under each length of time interval.

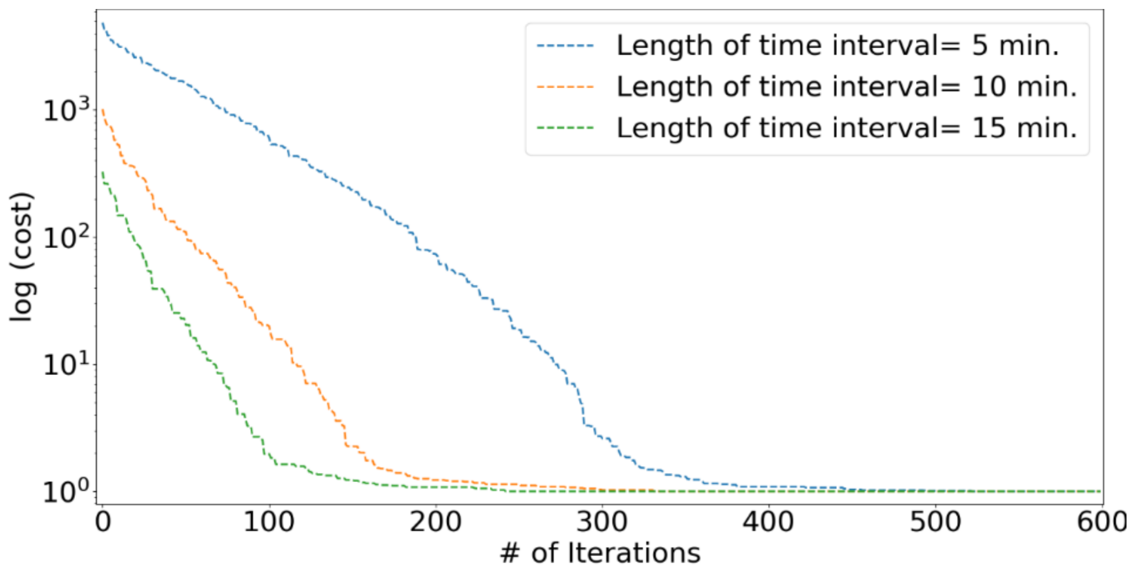

Figure 5.25 shows the convergence to optimal solution for scheduling elastic appliances. The number of iterations required to reach the optimal solution is 673 for 96 time slots, 673 for 144 time slots and 876 for 288 time slots. The number of fitness function evaluations to reach the optimal solution increases with variation in number of time slots from 144 to 288.However, the proposed framework can generate optimal solution even for smaller time intervals (5 min). Under each scenario a maximum cost saving of 46% is achieved. This implies objective function (4.23) is scalable in terms of increasing the length of operating windows for scheduling both *elastic* and *inelastic* appliances using canonical and binary version of PSO.

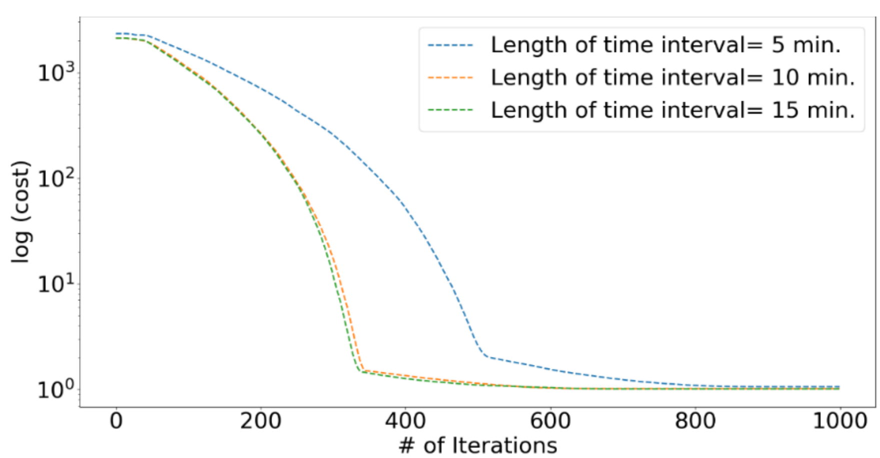

Figure 5.24 Convergence to Optimal solution for elastic appliances scheduling under varying time intervals

5.1.2. Number of Appliances

We will perform the sensitivity analysis to solve the objective function (4.23) under TOU pricing and increasing number of elastic and inelastic appliances. The input data for both category of appliances is adapted from Table 5.1 and 5.4. The number of particles in each sub swarm is set to 200 particles and maximum iteration count is set to 4000 iterations. In case of elastic appliances, the simulations are performed with set of 5 appliances (2 PHEV-1 ,2 Batter charger -1,1 Battery Charger-2), set of 10 appliances (4 PHEV-1 , 4 Batter charger -1,2 Battery Charger-2),set of 15 appliances(6 PHEV-1 , 6 Batter charger -1,3 Battery Charger-2) and set of 20 appliances (8 PHEV-1 , 8 Batter charger -1,4 Battery Charger-2).Table 5.1 compares the cost variation and number of iterations taken by PSO to reach near optimal solution under each simulation.

Table 5-11 Sensitivity w.r.t increasing number of elastic appliances

No. of Appliances	Solution Accuracy	Fitness Evaluations
5	98 %	700
10	95%	1500
15	90 %	4000
20	80%	4000

It is observed that scheduling elastic appliances as per objective function (4.23) yields feasible solution under each simulation, however the number of fitness evaluations required to reach feasible solution increases significantly with increase in number of appliances to be optimized. The accuracy of solution in terms of deviation from optimal solution is also compared in each scenario, which is 7% dropped with maximum number of appliances. However, the feasibility of solution is preserved in each case. This implies that the proposed framework can generate near optimal solution and hence is scalable in terms of number of appliances to be scheduled.

As similar to elastic scenario, we will now perform the validation of proposed framework to schedule inelastic appliances under different number of appliances. The simulations are performed to schedule 5 inelastic appliances (2 Dehumidifier’s, 2 well Pump’s, 1 Sump Pump’s), 10 appliances (4 Dehumidifier’s, 4 well Pump’s, 2 Sump Pump’s), 15 appliances

(6 Dehumidifier's, 6 well Pump's, 3 Sump Pump's), and 20 appliances (8 Dehumidifier's, 8 well Pump's, 4 Sump Pump's).

Table 5-12 Sensitivity w.r.t increasing number of inelastic appliances

No. of Appliances	Solution Accuracy	Fitness Evaluations
5	100 %	400
10	100%	1000
15	100 %	3000
20	97%	4000

Table 5.12 shows the number of fitness evaluations taken by binary PSO to find the feasible solution with increase in number of appliances. The accuracy of solution in terms of deviation from optimal solution is also compared in each scenario. It is observed that the algorithm converges to the near optimal solution in each case with a minimum accuracy of 80% while scheduling 20 appliances. Furthermore, the number of fitness function evaluations also increase as the search space dimensions are increased, but the algorithm returns a feasible solution in each scenario. This implies that the proposed framework is scalable in terms of number elastic of appliances to be scheduled. In addition, it proves the efficiency and performance of PSO for scheduling multiple appliances.

6. RESULTS AND DISCUSSION

6.1. Single house: The local perspective

We have considered a single dwelling having 12 operating appliances, each appliance having a random pre-determined start time (t_s), finish time (t_f), operating window(τ_a) and total energy consumption constraint across a day, as shown in Table 6.1.

Table 6-1 Load Data of single dwelling for simulation

Class	***App.***	***kWh***	***Start Time***	***Finish Time***	***Operating Window***	***kWh /day***
F	CS[1]	1.25	1(7AM)	3(7:45AM)	1-3(7AM-7:45AM)	3.75
F	PC&M[2]	0.050	1(7AM)	96(7AM)	1-96(7AM-7AM)	4.8
$\mathbf{E_{c_1}}$	PHEV	0.825	32(3:00PM)	45(4PM)	38-96(4:30PM-7AM)	2.175
$\mathbf{E_{c_1}}$	BC[3]	0.375	20(12Noon)	43(6PM)	1-60(7AM-10PM)	6.6
$\mathbf{E_{c_2}}$	LL[4]	0.200	52(8PM)	63(11PM)	52-63(8PM-11PM)	1.4
$\mathbf{E_{c_2}}$	CF[5]	0.038	88(5AM)	96(7AM)	88-96 (5AM-7AM)	0.42
$\mathbf{E_{c_3}}$	HVAC	0.375	35(3:45PM)	87(5AM)	35-87(3:45PM-5AM)	8
$\mathbf{E_{c_3}}$	Ref.[6]	0.125	1(7AM)	96(7AM)	1-96(7AM-7AM)	6
$\mathbf{I_{in}}$	CD[7]	0.875	36(4PM)	39(5PM)	36-59(4PM-10PM)	3.5
$\mathbf{I_{in}}$	FM[8]	0.525	16(11AM)	23(1PM)	0-55(7AM-9 PM)	4.2
$\mathbf{I_{un}}$	WM[9]	0.125	32(3PM)	34(3:45PM)	32-59(3PM-10PM)	0.375
$\mathbf{I_{un}}$	DW[10]	0.325	44(6PM)	49(7:30PM)	44-63(6PM-11PM)	1.95

[1]Cooking Stove, [2]PC & Monitor, [3]Battery Charger, [4]Lighting Loads, [5]Controllable Fans, [6]Referigerator, [7]Clothes Dryer, [8]Filter Motor, [9]Washing Machine, [10]Dishwasher.

The baseline load profile of inelastic and fixed appliances *I* is derived from start time, finish time and power consumption (kW) in each time slot. It is assumed that these appliances will consume their rated power during the 'ON' cycle. The objective function (4.23) is computed for *I* and *E* appliances separately, subject to constraints on *I* appliances (4.10 – 4.12), and on *E* appliances (4.4 – 4.9), respectively. In addition, a constraint is imposed for the rated capacity of the distribution panel (4.14). The input parameters of the thermal model for elastic appliances Type 3 are adopted from [25]; considering the cooling operation with maximum and minimum indoor temperatures constrained between 72°F and

68ºF for HVAC. Also, for the refrigerator, the temperature limits are constrained between 42ºF and 39ºF.

The initial PSO parameters for scheduling *I* and *E* appliances are given in Tables 5.3 & 5.6, respectively. The cost and comfort weights in the objective functions are kept similar to the use case studies in previous chapter and are referred to as maximum comfort ($w_1 = 0.8\,, w_2 = 0.2$), medium comfort ($w_1 = 0.4\,, w_2 = 0.6$) and minimum comfort ($w_1 = 0.4\,, w_2 = 0.6$).

The simulations are performed under different comfort conditions to measure the effect on scheduled cost and peak demand under three scenarios:

1. *TOU pricing:* The objective function (4.23) is minimized with value of penalty function coefficient λ = 10 and $\theta_{dr} = 0$ as discussed earlier. The rated capacity of distribution panel is set to 2.5 kWh in (4.14).
2. *Mandatory DR request:* Energy consumption scheduling of individual dwelling is performed considering the combination of TOU and limiting rated capacity of distribution panel ($\psi_r = 1.6$). The value of λ and θ_{dr} remains same as the TOU pricing scenario.
3. *Non-Mandatory DR request:* Optimization is performed considering the combination of TOU and non-mandatory DR event from 5:00 PM to 7:00 PM with reward $\delta_{dr} = 0.15$ \$/kWh and $\theta_{dr} = 1$.

6.1.1. TOU Pricing

Figure 6.1 compares the unscheduled and scheduled energy consumption profile of a house under TOU pricing and different comfort level settings. In an unscheduled case shown with a solid blue line in Figure 6.1, user activities are high during 3:00 PM – 5:00 PM and 6:00PM -7:00PM resulting in high peak demand of 2.2 kWh and electricity cost of 5.11 $/day. Scheduled electricity prices under TOU pricing (i.e. scenario 1) and different comfort levels are listed in column 1 of Table 6-2. The effect on peak energy demand due to redistribution of loads under different comfort levels is depicted in column 1 of Table 6.3.

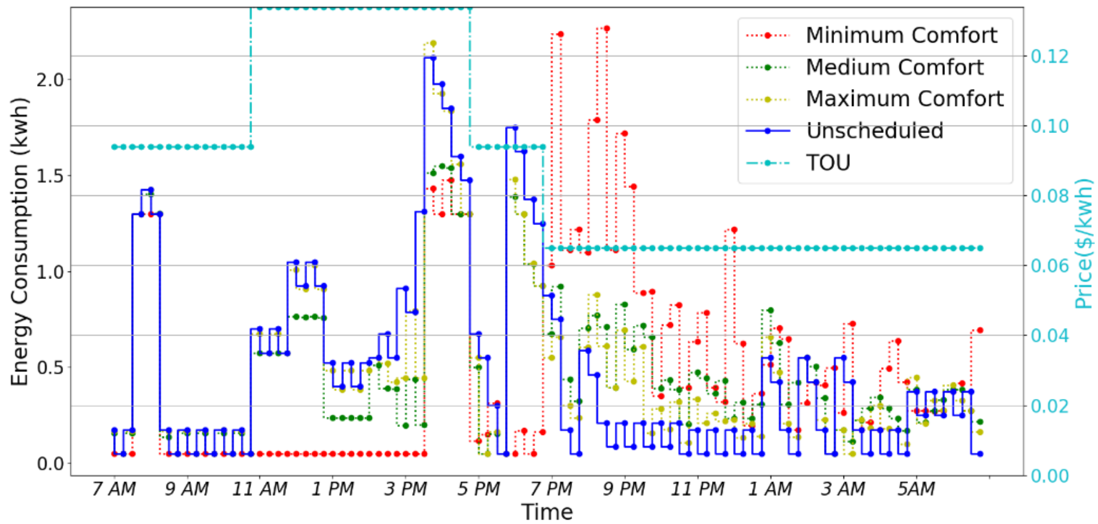

Figure 6.1 Unscheduled vs Scheduled energy consumption profile- Scenario 1

Electricity cost savings are achieved in range of 27 % to 7 % under increasing comfort level. Under low comfort level settings, electricity cost is reduced by 27% and peak demand during high price periods (11AM-5PM) is reduced by 40% as compared to unscheduled case under minimum comfort level. However, a peak rebound effect under minimum comfort conditions is obvious, as the majority of the loads are shifted to low price periods (7PM-9PM). This is because appliances are not governed by customer comfort and hence will move towards low price time intervals. This shows that, under low comfort levels, optimization is biased towards reducing electricity cost.

Table 6-2 Cost comparison of single household under different comfort levels

Comfort Level	**Energy Cost ($/day)**		
	TOU	**Mandatory DR**	**Non-mandatory DR**
Minimum	3.8	3.8	2.5
Medium	4.4	4.3	2.9
Maximum	4.8	4.7	4.1

For medium comfort conditions, the electricity cost and peak demand is reduced by 16% and 28% respectively, as appliances are partially governed by comfort level and hence partially restricted to exploit time varying electricity prices.

The electricity cost and peak demand under maximum comfort conditions is 4.8 $/day and 2.15 kWh which is close to unscheduled case, since appliances tend to follow the baseline energy consumption pattern and hence objective function (4.23) is less biased towards cost minimization objective.

Table 6-3 Scheduled peak energy consumption under different comfort levels

Comfort Level	Peak Energy Demand (kWh)		
	TOU	*Mand. DR*	*Non-Mand. DR*
Minimum	2.2	1.54	2.2
Medium	1.6	1.57	1.60
Maximum	2.15	1.55	2.3

6.1.2. Mandatory DR Event

Figure 6.3 shows the comparison of unscheduled and scheduled energy consumption profile under mandatory DR request with different comfort level settings. The comparison of cost and peak energy demand is listed in column 2 of Table 6.2 and 6.3 respectively. Under minimum comfort level setting, achieved electricity cost savings comes out to be 27% and peak demand is within the specified hard limit of 1.6 kWh between 7:00 PM – 8:00PM. The comparison of cost in scenario 1 & 2 under minimum comfort conditions is almost similar as the temporal flexibility constraints of heavy loads such as PHEV -1 and HVAC are accommodating the mandatory DR request.

In case of medium and maximum comfort level, the algorithm optimally schedules the energy consumption to satisfy the constraint (4.14). Cost and peak energy demand change is minimal under TOU pricing scenario and Mandatory DR request for medium comfort conditions, as the value of hard constraint imposed (1.6 kWh) is equal to the peak demand obtained during TOU pricing. Consumer comfort is overridden by the mandatory DR request in case of maximum comfort; however, the cost of electricity is similar to the TOU pricing scenario as energy consumption scheduling is biased towards consumer comfort and satisfying upper limit constraint. This implies that the proposed algorithm is expected to compute the near optimal solution under mandatory DR request and hence will support the grid to lower the peak demand under system contingency conditions.

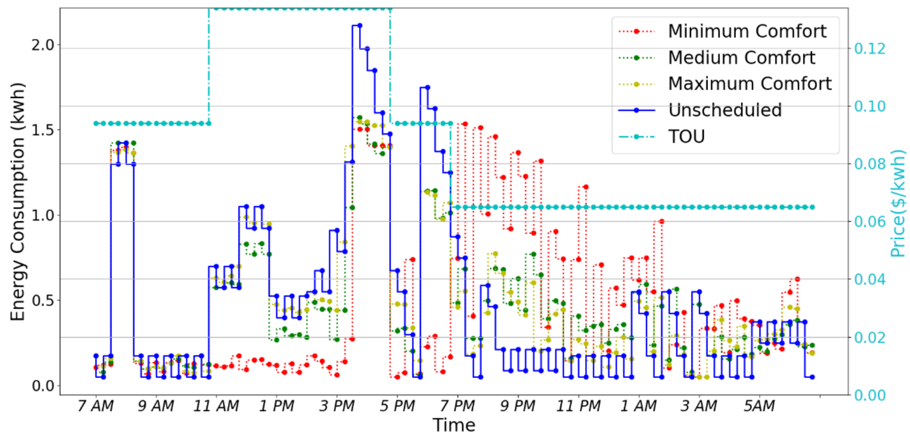

Figure 6.2 Scheduled vs unscheduled energy consumption profile-Scenario 2

6.1.3. Non-Mandatory DR Event

Figure 6.3 compares unscheduled and scheduled energy consumption profiles under the impact of CPR (critical peak pricing) δ_{dr}^{t} = 0.15 \$/kWh (*i.e.,* a financial reward). The Non-Mandatory DR is activated between 5:00 PM and 7:00 PM. Daily energy cost and peak demand under different comfort level settings are summarized in column 3 of Table 6.2 and 6.3 respectively.

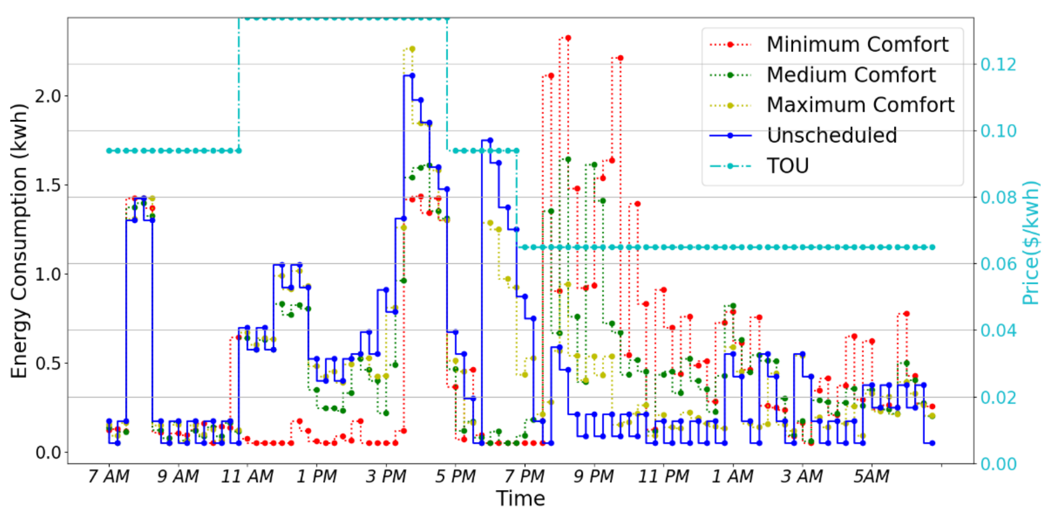

Figure 6.3 Scheduled vs unscheduled energy consumption profile-Scenario 3

Under minimum comfort level settings, the energy cost and peak energy demand are 2.5 \$/day and 2.3 kWh respectively. The peak energy demand during DR time intervals is reduced to 0.5 kWh as HEMS optimizes the objective function (4.23) to maximize the consumer savings.

In case of medium comfort level, the electricity cost is 2.9 \$/kWh and peak energy demand of 1.6 kWh is observed during low TOU pricing periods i.e. between 7:00 PM-9:00PM. The peak energy demand under DR interval is reduced to 0.4 kWh.

Finally, in case of maximum comfort level, the electricity price is price is reduced to 4.1 \$/day and peak energy demand is 2.2 kWh as appliances tend to operate close to baseline state. The peak demand during DR intervals is reduced to 1.3 kWh.

6.2. Multiple Houses: The global Perspective

We have considered six households, each having 12 operating appliances as shown in Appendix section (Table 8.1 – Table 8.6). Each household has different baseline energy consumption profile generated randomly by changing the starting time. Figure 8.7(appendix) shows the baseline energy consumption profile of individual house. Optimization is performed individually for each house under low, medium and high comfort level and aggregate load profile is generated for unscheduled and scheduled case. The aggerate cost and peak energy demand is generated from Table 8.7 (Appendix) and compared under three different pricing signals. It is assumed that each customer participates in the Non-Mandatory DR event. The input parameters of PSO are as given in Table 5.3 and 5.6 for scheduling inelastic and elastic appliances.

6.2.1. TOU Pricing

Under this scenario, the objective function (4.23) is solved by HEMS in each household under TOU pricing signal and different comfort levels. The value of DR activation parameter θ_{dr} = 0 and λ = 10. Figure 6.4 compares the aggregate energy consumption profile for six households under low, medium and high comfort level settings. Solid black line indicates an unscheduled energy consumption pattern. During 5PM – 7PM, maximum energy consumed by aggregate houses under the unscheduled case comes out to be 12.2

kWh. Scheduled scenario shown by red, yellow and green line shows the redistribution of energy consumption profile under low. Medium and high comfort level settings. Table 6.4

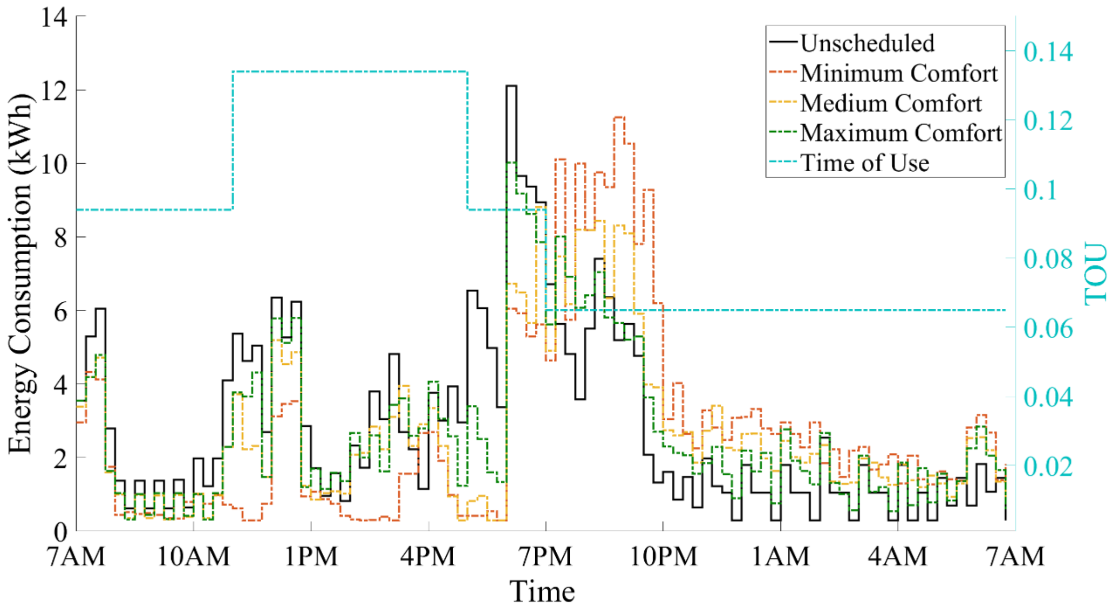

Figure 6.4 Scheduled vs unscheduled energy consumption profile-Multiple Homes under TOU Pricing

and 6.5 shows the comparison of electricity cost and aggregate peak energy demand unscheduled and scheduled case. The unscheduled electricity cost is 26 $/day for multiple houses and peak energy demand of 12.2 kWh observed between 5PM-7PM.Under TOU pricing, a peak reduction of 11% -7% and electricity cost reduction of 24% - 7% is achieved as compared to baseline state. A low peak demand is observed under medium comfort level as compared to low and high comfort level because the appliances are partially governed by comfort and hence, cannot fully exploit the low prices. The electricity cost reduction varies linearly with the comfort level settings. The aggregation of scheduled household load profiles may eliminate the peak rebound effect at distribution transformer level. It should be noted that peak rebound effect under TOU pricing is caused due to lesser assigned flexibility to schedule appliances as appliances are moved to low price periods by HEMS.

Table 4-4 Aggregate Energy Cost comparison under different comfort levels-Multiple Houses

Comfort Level	Energy Cost ($/day)		
	TOU	Mandatory DR	Non-mandatory DR
Unscheduled	26	26	26
Minimum	20	20	14.3
Medium	22.6	22.8	18.7
Maximum	24.2	24.8	21.5

6.2.2. Mandatory DR: Combination of TOU and Upper Power Threshold

We will now investigate the effect of mandatory DR event on aggregate peak energy demand and electricity cost. The upper limit on maximum power withdrawn from each household is set to 1.6 kWh. Objective function (4.23) is solved to find the optimal energy consumption distribution of each household under hard constraint (4.14). Value of DR activation parameter θ_{dr} is set to 0. Unscheduled energy consumption pattern is similar to

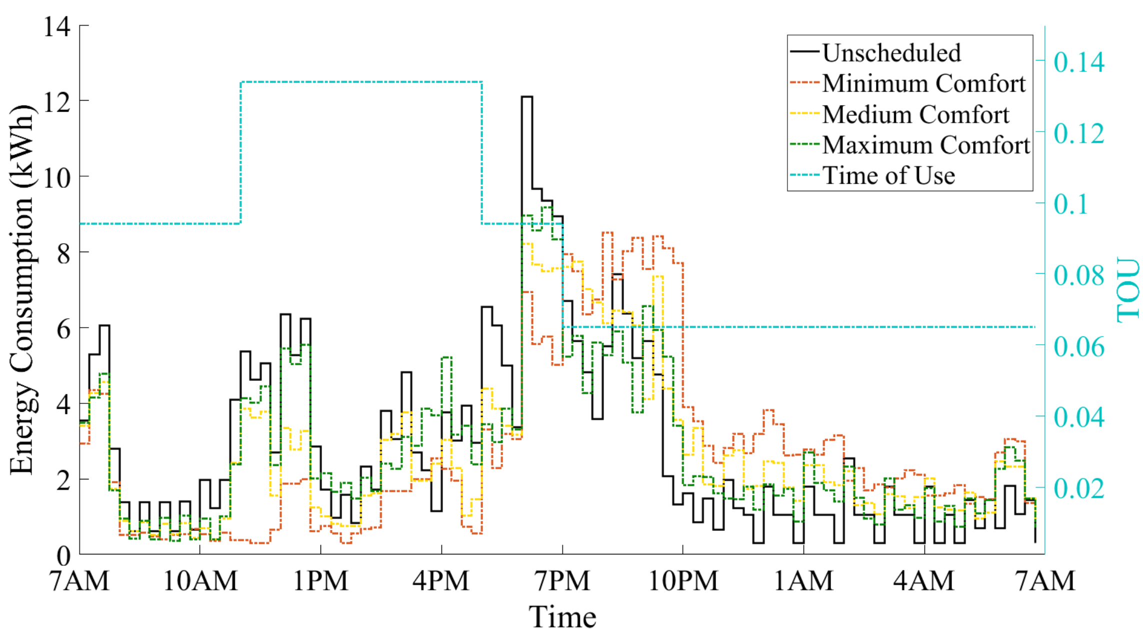

Figure 6.5 Scheduled vs unscheduled energy consumption profile-Multiple Homes under Scenario 2

the above scenario with peak demand of 12.2 kWh. The comparison of unscheduled and scheduled peak energy demand and electricity cost is presented in Table 6.4 and 6.5. The scheduled peak energy demand is within the specified limit for each household. The maximum aggregate peak energy demand is 9.2 kWh under maximum comfort, which is beyond the specified limit of 9.6 kWh. Furthermore, the difference in electricity cost comparison under TOU scenario and Mandatory DR event is not obvious. This implies that proposed algorithm is capable of optimally scheduling the energy consumption of each household under mandatory DR event while satisfying the operating constraints of appliances.

Table 6-5 Aggregate Peak Energy Demand (kWh) under different comfort levels-Multiple Houses

Comfort Level	Peak Energy Demand kWh)		
	TOU	Mandatory DR	Non-mandatory DR
Unscheduled	12.2	12.2	12.2
Minimum	11.3	8.5	11.2
Medium	11	8.2	8.8
Maximum	11.1	9.2	10

6.2.3. Non-mandatory DR: Combination of TOU and CPR Pricing

Under this scenario, objective function (4.23) is minimized under non-mandatory DR request, *i.e.* the value of DR activation parameter θ_{dr} = 1. Table 6.4 and 6.5 shows the comparison of aggregate electricity cost and peak energy demand under unscheduled and scheduled case. The baseline energy consumption pattern is similar to the above scenario with peak energy demand of 12.2kWh and aggregate electricity cost of 26$/day. The DR activation time period is 3PM- 7PM and CPR signal is δ_{dr} = 0.15 $/kWh. Peak demand during DR activation time intervals is reduced by 32 %. However, a peak demand of 11.2 kWh occurs under low comfort level between 8:00 PM-9:00 PM as optimization tends to maximize the reward by exploiting low price periods. Under medium comfort level as appliances are partially governed by cost and comfort a peak demand reduction of 28% is achieved as compared to unscheduled case. Furthermore, under high comfort level settings 18% reduction in peak demand is observed as compared to unscheduled case. However, non-mandatory DR request may result in peak rebounding effect as optimization aims to maximize the reward. The aggregate electricity cost dependence on comfort level is similar to single house simulations. The aggregate cost reduction ranges from 45% to 18% under low to high comfort levels, as customer is rewarded for shifting energy consumption.

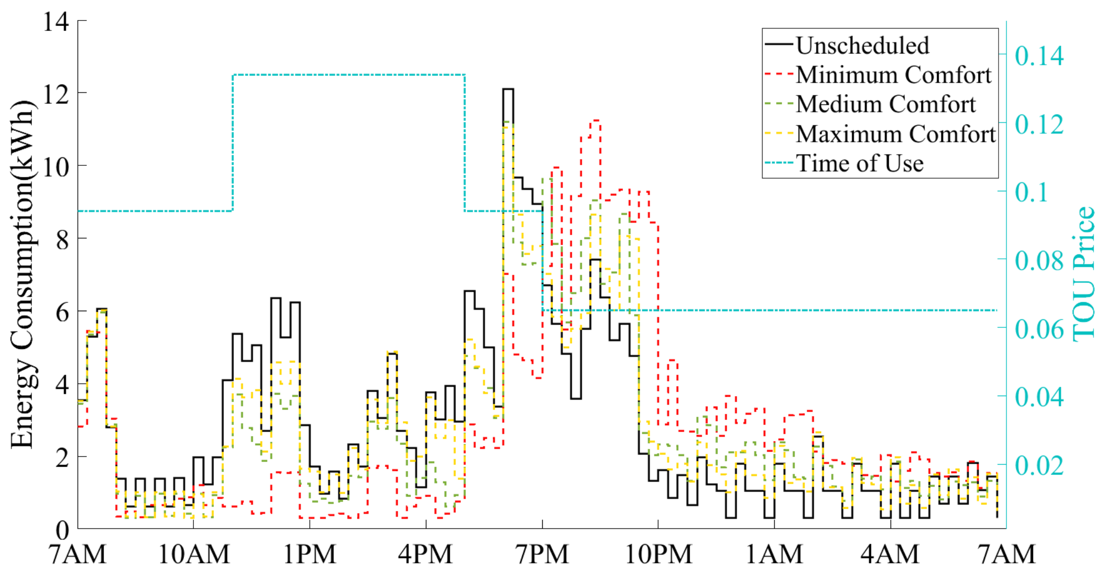

Figure 6.6 Scheduled vs unscheduled energy consumption profile-Multiple Homes under scenario 3

6.2.4. Average and Standard Deviation: Multiple Houses

The simulations are performed under random initial distribution of load profiles for six houses. Furthermore, the simulations are performed subjected to different comfort levels and pricing signals. Table 6.6 shows the average and standard deviation of electricity cost and peak energy demand for six houses under three scenarios. The average baseline electricity cost and peak energy demand is 4.3$/day and 2.7 kWh. The average electricity cost decreases with drop in comfort level under each scenario. However, the variation in electricity cost standard deviation between Scenario 1 and 3 under different comfort levels is obvious. The peak energy demand average and SD increases as comfort level drops. This is because under minimum comfort level appliances are not governed by comfort and hence optimization tend to schedule the appliances away from high TOU prices.

Table 6-6 Average and Standard Deviation - Multiple Houses under Scenario-1,2&3

Comfort Level	**TOU**				**Mandatory DR**				**Non-Mand. DR**			
	$/day		kWh		$/day		kWh		$/day		kWh	
	Avg	SD	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD
Baseline	4.3	0.36	2.7	0.67	4.3	0.36	2.7	0.67	4.3	0.36	2.7	0.67
Maximum	4.1	0.29	2.4	0.28	4.1	0.27	1.9	0.40	4.0	0.34	2.3	0.34
Medium	3.8	0.26	2.4	0.25	3.8	0.27	1.9	0.42	3.3	0.43	2.1	0.35
Minimum	3.3	0.19	2.7	0.54	3.4	0.23	2	0.59	2.8	0.54	2.3	0.40

7. CONCLUSIONS

7.1. Conclusion

Demand side management is an effective way to reduce distribution network peak demand as well as electricity cost. We here have demonstrated an approach to optimally schedule the energy consumption of individual dwellings in order to maximize individual net reward subject to utility-imposed constraints and user preference constraints. We have categorized load resources in terms of their flexibility in energy consumption and operating time. Particle swarm optimization is used for scheduling of appliances. The energy consumption scheduling problem is formulated as a multi-objective optimization problem with an aim to minimize cost and discomfort. We have scheduled elastic and inelastic appliances separately using binary and canonical version of PSO considering modelling of loads. The decision vector for inelastic appliances is binary, whereas for elastic appliances it is a continuous value. However, decision vector can be relaxed to either binary or continuous by changing the load models or categorizing appliances solely as time or energy shift able.

Additionally, we have also presented a use case analysis for scheduling both elastic and inelastic appliances which is validating the problem formulation and is proving performance of PSO to find the optimal solution. The optimization is performed for single and multiple houses. The appliances are scheduled under three load shaping strategies: TOU, combination of TOU and CPR and combination of TOU and hard constraint imposed by utility on rated power consumption. The binary and canonical version of PSO are implemented for multi-dimensional energy consumption scheduling in Python v 3.7. All simulations were performed on Intel Core i7-6700HQ CPU. The results under TOU signal resulted in electricity cost savings in range of 28% to 7% in different comfort conditions and a peak reduction of 28% under medium comfort level. The energy consumption scheduling under combination of TOU and CPR signal reveals that even under modest TOU pricing, it is comfort not CPR that constrains the DR response. This has significant implications for demand side management, consumer engagement and education. In addition, optimization under combination of TOU and hard constraints depresses the peak by 28% with a minimal increase in electricity cost as compared to TOU strategy. This shows that hard constraint imposed on each house can bypass the consumer comfort to support grid during system stress conditions. The energy consumption scheduling of multiple houses under TOU pricing and combination of TOU and hard constraint resulted in peak demand reduction of 17% and 27% respectively. On the contrary, under TOU and CPR, a peak rebound effect causes increase in peak demand by 9 % as HEMS tends to maximize the consumer reward for DR participation by shifting majority of appliances to low price periods.

7.2. Future Research

The future research directions can include extending the existing HEMS model to include grid connected Photo Voltaic System with Battery Management System. As the cost of owning PV system is significantly higher, future research direction will be to study the effectiveness of grid connected PV as compared to islanded mode to develop a more cost-effective solution.

Also, Implementation of heuristic algorithms such as Wind Driven Optimization (WDO), Ant colony optimization (ACO) for HEMS benchmarking could be another possible research of future. Additionally, the proposed scheme is designed for optimizing individual house and may lead to sub-optimal solution at aggregate level. Therefore, use of game theoretic model to include multiple households connected to single distribution feeder can provide optimal solution as HEMS installed in each house can negotiate to balance load quantity and electricity prices.

8. APPENDIX

8.1. Multiple Houses Data

Table 8-1 Load Profile Data for House # 1

Class	***App.***	***kWh***	***Start Time***	***Finish Time***	***Operating Window***	***kWh /day***
F	CS[1]	1.25	1(7AM)	3(7:45AM)	1-3(7AM-7:45AM)	3.75
F	CS[1]	1.25	44(6PM)	46(6:45PM)	44-46(6PM-6:45PM)	3.75
F	PC&M[2]	0.050	1(7AM)	96(7AM)	1-96(7AM-7AM)	4.8
$\mathbf{E_{c_1}}$	PHEV	0.825	36(4PM)	39(5PM)	36-96(4PM-7AM)	2.175
$\mathbf{E_{c_1}}$	BC[3]	0.375	36(4PM)	56(9PM)	36-75(4PM-2AM)	6.6
$\mathbf{E_{c_2}}$	LL[4]	0.200	43(6PM)	51(8PM)	43-51(6PM-8PM)	1.4
$\mathbf{E_{c_2}}$	CF[5]	0.038	32(3PM)	43(6PM)	32-45(3PM-6PM)	0.42
$\mathbf{E_{c_3}}$	HVAC	0.375	26(1:30PM)	55(9PM)	26-55(1:30PM-9PM)	3.375
$\mathbf{E_{c_3}}$	Ref.[6]	0.125	1(7AM)	96(7AM)	1-96(7AM-7AM)	6
$\mathbf{I_{in}}$	CD[7]	0.875	15(10:45AM)	18(11:45AM)	10-60(9:30AM-10:15PM)	3.5
$\mathbf{I_{in}}$	FM[8]	0.525	47(6:45PM)	54(8:45PM)	44-60(6PM-10:15PM)	4.2
$\mathbf{I_{un}}$	WM[9]	0.125	12(10AM)	14(10:45AM)	32-59(3PM-10PM)	0.375
$\mathbf{I_{un}}$	DW[10]	0.325	4(8AM)	9(10AM)	4-63(8AM-11PM)	1.95

[1]Cooking Stove, [2]PC & Monitor, [3]Battery Charger, [4]Lighting Loads, [5]Controllable Fans, [6]Referigerator, [7]Clothes Dryer, [8]Filter Motor, [9]Washing Machine, [10]Dishwasher.

Table 8-2 Load Profile data for House # 2

Class	***App.***	***kWh***	***Start Time***	***Finish Time***	***Operating Window***	***kWh /day***
F	CS[1]	1.25	3(7:45AM)	5(8:30AM)	3-5(7:45AM-8:30AM)	3.75
F	CS[1]	1.25	35(3:45PM)	37(4:30PM)	35-39(3:45PM-4:30PM)	3.75
F	PC&M[2]	0.050	1(7AM)	96(7AM)	1-96(7AM-7AM)	4.8
$\mathbf{E_{c_1}}$	PHEV	0.825	32(3PM)	35(4PM)	32-96(3PM-7AM)	2.175
$\mathbf{E_{c_1}}$	BC[3]	0.375	20(12Noon)	40(5:15PM)	20-75(12Noon-2AM)	6.6
$\mathbf{E_{c_2}}$	LL[4]	0.200	88(5AM)	95(7AM)	88-95(5AM-7AM)	1.4
$\mathbf{E_{c_2}}$	CF[5]	0.038	52(8PM)	62(10:45PM)	52-63 (8PM-11PM)	0.42
$\mathbf{E_{c_3}}$	HVAC	0.375	35(3:45PM)	87(5AM)	35-87(3:45PM-5AM)	6
$\mathbf{E_{c_3}}$	Ref.[6]	0.125	1(7AM)	96(7AM)	1-96(7AM-7AM)	6
$\mathbf{I_{in}}$	CD[7]	0.875	44(6PM)	47(7PM)	30-60(2:30PM-10:15PM)	3.5
$\mathbf{I_{in}}$	FM[8]	0.525	16(11AM)	23(1PM)	15-60(10:45AM-10:15PM)	4.2
$\mathbf{I_{un}}$	WM[9]	0.125	40(5PM)	42(5:45PM)	30-40(2:30PM-5:15PM)	0.375
$\mathbf{I_{un}}$	DW[10]	0.325	44(6PM)	49(8PM)	40-44(5PM-6:15PM)	1.95

[1]Cooking Stove, [2]PC & Monitor, [3]Battery Charger, [4]Lighting Loads, [5]Controllable Fans, [6]Referigerator, [7]Clothes Dryer, [8]Filter Motor, [9]Washing Machine, [10]Dishwasher.

Table 8-3 Load Profile Data of House # 3

Class	***App.***	***kWh***	***Start Time***	***Finish Time***	***Operating Window***	***kWh /day***
F	CS[1]	1.25	1(7AM)	3(7:45AM)	1-3(7AM-7:45AM)	3.75
F	CS[1]	1.25	53(8:15PM)	55(9PM)	53-55(8:15 PM-9PM)	3.75
F	PC&M[2]	0.050	1(7AM)	96(7AM)	1-96(7AM-7AM)	4.8
$\mathbf{E_{c_1}}$	PHEV	0.825	46(6:30PM)	49(7:30PM)	46-96(6:30 PM-7AM)	2.175
$\mathbf{E_{c_1}}$	BC[3]	0.375	36(4PM)	55(9PM)	36-96(4PM-7AM)	6.6
$\mathbf{E_{c_2}}$	LL[4]	0.200	44(6PM)	51(8PM)	44-63(6PM-11PM)	1.4
$\mathbf{E_{c_2}}$	CF[5]	0.038	40(5PM)	51(8PM)	40-51 (5PM-8PM)	0.42
$\mathbf{E_{c_3}}$	HVAC	0.375	44(6PM)	87(5AM)	44-87(6PM-8AM)	4.5
$\mathbf{E_{c_3}}$	Ref.[6]	0.125	1(7AM)	96(7AM)	1-96(7AM-7AM)	6
$\mathbf{I_{in}}$	CD[7]	0.875	15(10:45AM)	19(11:45AM)	15-59(10:45 AM-10PM)	3.5
$\mathbf{I_{in}}$	FM[8]	0.525	48(7PM)	55(9PM)	40-59(5PM-10PM)	4.2
$\mathbf{I_{un}}$	WM[9]	0.125	12(10AM)	14(10:45AM)	10-59(9:30 AM-10PM)	0.375
$\mathbf{I_{un}}$	DW[10]	0.325	54(8:30PM)	59(10PM)	30-63(2:30 PM-11PM)	1.95

[1]Cooking Stove, [2]PC & Monitor, [3]Battery Charger, [4]Lighting Loads, [5]Controllable Fans, [6]Referigerator, [7]Clothes Dryer, [8]Filter Motor, [9]Washing Machine, [10]Dishwasher.

Table 8-4 Load Profile Data for House # 4

Class	***App.***	***kWh***	***Start Time***	***Finish Time***	***Operating Window***	***kWh /day***
F	CS[1]	1.25	30(2:30 PM)	32(3:15PM)	30-32(2:30 PM-3:15PM)	3.75
F	CS[1]	1.25	40(5PM)	42(5:45PM)	40-42(5PM-5:45PM)	3.75
F	PC&M[2]	0.050	1(7AM)	96(7AM)	1-96(7AM-7AM)	4.8
$\mathbf{E_{c_1}}$	PHEV	0.825	36(4PM)	39(5PM)	36-96(4PM-7AM)	2.175
$\mathbf{E_{c_1}}$	BC[3]	0.375	46(6:30PM)	96(7AM)	45-96(6:30PM-7AM)	6.6
$\mathbf{E_{c_2}}$	LL[4]	0.200	40(5PM)	46(6:45PM)	40-63(5PM-11PM)	1.4
$\mathbf{E_{c_2}}$	CF[5]	0.038	44(6PM)	54(8:45PM)	44-63 (6PM-11PM)	0.42
$\mathbf{E_{c_3}}$	HVAC	0.375	44(6PM)	87(5AM)	44-87(6PM-5AM)	4.5
$\mathbf{E_{c_3}}$	Ref.[6]	0.125	1(7AM)	96(7AM)	1-96(7AM-7AM)	6
$\mathbf{I_{in}}$	CD[7]	0.875	48(7PM)	52(8PM)	36-59(4PM-10PM)	3.5
$\mathbf{I_{in}}$	FM[8]	0.525	15(10:45 AM)	22(12:45PM)	15-59(10:45AM-10PM)	4.2
$\mathbf{I_{un}}$	WM[9]	0.125	54(8:30PM)	56(9:15PM)	30-59(2:30PM-10PM)	0.375
$\mathbf{I_{un}}$	DW[10]	0.325	12(10 AM)	63(11:30PM)	12-63(10AM-11PM)	1.95

[1]Cooking Stove, [2]PC & Monitor, [3]Battery Charger, [4]Lighting Loads, [5]Controllable Fans, [6]Referigerator, [7]Clothes Dryer, [8]Filter Motor, [9]Washing Machine, [10]Dishwasher.

Table 8-5 Load Profile Data of House # 5

Class	***App.***	***kWh***	***Start Time***	***Finish Time***	***Operating Window***	***kWh /day***
F	CS[1]	1.25	1(7AM)	3(7:45AM)	1-3 (7AM-	3.75
F	CS[1]	1.25	44(6PM)	46(6:45PM)	44-46(6PM-6:45PM)	3.75
F	PC&M[2]	0.050	1(7AM)	96(7AM)	1-96(7AM-7AM)	4.8
$\mathbf{E_{c_1}}$	PHEV	0.825	38(4:30PM)	41(5:30PM)	38-96(4:30PM-7AM)	2.175
$\mathbf{E_{c_1}}$	BC[3]	0.375	20(12Noon)	39 (5PM)	20-75(12Noon-2AM)	6.6
$\mathbf{E_{c_2}}$	LL[4]	0.200	88(5AM)	96(7AM)	88-96(5AM-7AM)	1.4
$\mathbf{E_{c_2}}$	CF[5]	0.038	52(8PM)	62(10:45PM)	52-63(8PM-11PM)	0.42
$\mathbf{E_{c_3}}$	HVAC	0.375	35(3:45PM)	87(5AM)	35-87(3:45PM-5AM)	6
$\mathbf{E_{c_3}}$	Ref.[6]	0.125	1(7AM)	96(7AM)	1-96(7AM-7AM)	6
$\mathbf{I_{in}}$	CD[7]	0.875	36(4PM)	39(5PM)	36-59(4PM-10PM)	3.5
$\mathbf{I_{in}}$	FM[8]	0.525	16(11AM)	23(1PM)	16-59(11AM-10PM)	4.2
$\mathbf{I_{un}}$	WM[9]	0.125	32(3PM)	35(3:45PM)	30-59(2:30PM-10PM)	0.375
$\mathbf{I_{un}}$	DW[10]	0.325	44(6:00PM)	49(7:30PM)	44-59 (6PM-10PM)	1.95

[1]Cooking Stove, [2]PC & Monitor, [3]Battery Charger, [4]Lighting Loads, [5]Controllable Fans, [6]Referigerator, [7]Clothes Dryer, [8]Filter Motor, [9]Washing Machine, [10]Dishwasher.

Table 8-6 Load Profile Data of House # 6

Class	***App.***	***kWh***	***Start Time***	***Finish Time***	***Operating Window***	***kWh /day***
F	CS[1]	1.25	3(7:45AM)	5(8:15AM)	3-5 (7:45AM - 8:15AM)	3.75
F	CS[1]	1.25	40(6PM)	42(6:45PM)	40-42(6PM- 6:45PM)	3.75
F	PC&	0.050	1(7AM)	96(7AM)	1-96(7AM-7AM)	4.8
$\mathbf{E_{c_1}}$	PHEV	0.825	46(6:30PM)	49(7:30PM)	44-96(6PM- 7AM)	2.175
$\mathbf{E_{c_1}}$	BC[3]	0.375	36(4PM)	55 (9 PM)	36-96(4PM- 7AM)	6.6
$\mathbf{E_{c_2}}$	LL[4]	0.200	44(6PM)	51(8PM)	44-51(6PM-8PM)	1.4
$\mathbf{E_{c_2}}$	CF[5]	0.038	40(5PM)	50(7:45PM)	40-63(5PM- 11PM)	0.42
$\mathbf{E_{c_3}}$	HVAC	0.375	44(3:45PM)	87(5AM)	35-87(3:45PM- 5AM)	4.5
$\mathbf{E_{c_3}}$	Ref.[6]	0.125	1(7AM)	96(7AM)	1-96(7AM-7AM)	6
$\mathbf{I_{in}}$	CD[7]	0.875	15(10:45AM)	18(11:45AM)	15-59(10:45AM - 10PM)	3.5
$\mathbf{I_{in}}$	FM[8]	0.525	48(7PM)	55(9PM)	40-59(5PM- 10PM)	4.2
$\mathbf{I_{un}}$	WM[9]	0.125	12(10AM)	14(10:45AM)	12-59(10AM- 10PM)	0.375
$\mathbf{I_{un}}$	DW[10]	0.325	54(8:30PM)	59(10:00PM)	44-59 (6PM- 10PM)	1.95

[1]Cooking Stove, [2]PC & Monitor, [3]Battery Charger, [4]Lighting Loads, [5]Controllable Fans, [6]Referigerator, [7]Clothes Dryer, [8]Filter Motor, [9]Washing Machine, [10]Dishwasher.

Table 8-7 Cost and Peak Energy Demand Comparison of Multiple Houses

House No.	Cost / Peak	Base line	TOU			Mandatory			Non- Mandatory		
			Max	**Med**	**Min**	**Max**	**Med**	**Min**	**Max**	**Med**	**Min**
House1	$/day	4.2	4.0	3.6	3.2	3.8	3.6	3.2	4.4	2.6	2.5
	kWh	2.6	2.2	2.1	2.7	1.8	1.8	1.8	2.5	2.6	2.7
House2	$/day	4.6	4.4	4.1	3.4	4.4	4	3.5	4.1	3.2	2.5
	kWh	3	2.5	2.4	2.7	1.66	1.67	1.64	2.6	2.5	2.7
House3	$/day	3.9	3.8	3.5	3.1	3.8	3.5	3.1	3.8	3.5	3.2
	kWh	2.45	2.7	2.8	3.3	2.7	2.8	3.3	1.88	1.82	1.83
House4	$/day	4.1	3.7	3.5	3.1	4.1	3.9	3.6	4.1	3.9	3.6
	kWh	2.175	2.7	2.5	3.3	1.5	1.7	1.8	2.7	2.5	2.3
House5	$/day	4.9	4.4	4	3.5	4.4	4	3.7	4	3.2	2.1
	kWh	3.9	2.3	2.2	2.3	1.8	1.75	1.8	2.3	2.2	2.6
House6	$/day	4.3	4	3.8	3.5	4.2	3.8	3.5	3.4	3.1	2.9
	kWh	2.1	2	2.23	1.93	1.8	1.7	1.8	1.94	1.84	1.9

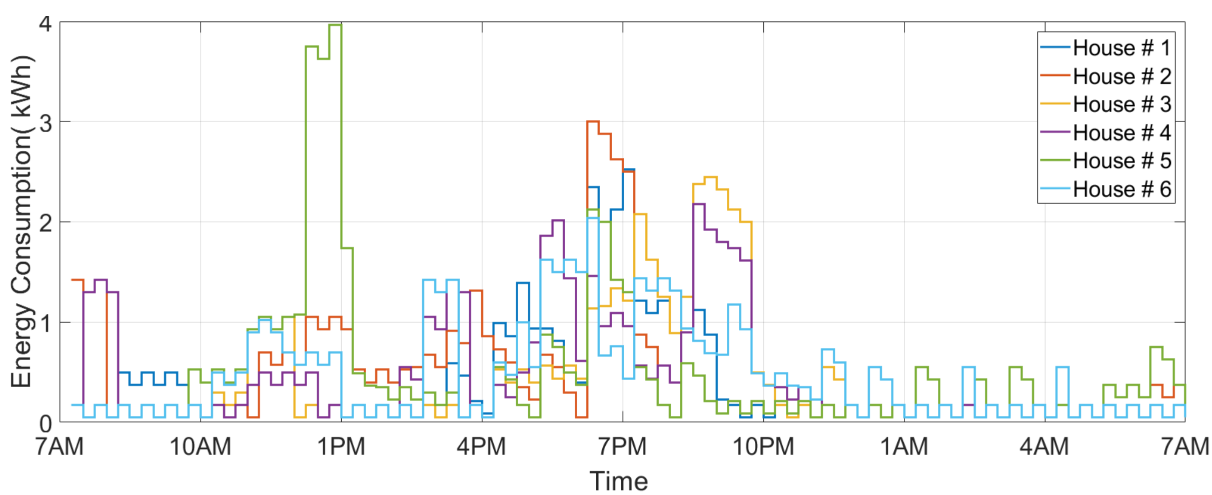

Figure 8.1 Baseline Energy Consumption Profiles of Multiple Houses

8.2. Source Code

```
"""
@author: Rohit Singh Rana
@Continuous and Binary MDPSO with Time Varying Acceleration
Coefficients
@ coded in Python3.7
"""

#----------------Import Libraries----------------------------+
import numpy as np
#-----------------Class Particle ----------------------------+
class Particle:
    def __init__(self,en_min,en_max):
        self.position  = np.zeros([Na,Np,Nd])
        self.v = np.zeros([Na,Np,Nd])
        self.cost = [0] * Np
        self.pbest_cost = [np.inf] * Np
        self.pbest_position = self.position
        self.vmax = 0.01
        self.vmin = -self.vmax
        self.vmax = [0] * Na
        self.vmin = [0] * Na
        for i in range (0,Na):
            self.position[i] = \
np.random.uniform(en_min[i],en_max[i],size = (Np,Nd))
            self.position[i] = np.random.normal(sigma,mu,(Np,Nd))
            self.vmax[i] = 0.02 * en_max[i]
            self.vmin[i] = -self.vmax[i]
# Evaluate Objective Function
    def evaluate_objfcn(self,objective_fcn):
        self.cost = objective_fcn(self.position)
        for i in range(0,Np):
            if self.cost[i] < self.pbest_cost[i]:
# Update Personal Best Here
                self.pbest_cost[i] = self.cost[i]
                for k in range (0,Na):
                    self.pbest_position [k][i][:]= self.position
[k][i][:]
# Update Velocity
    def update_velocity(self,gbest_position,maxiter,itr):
        c1_max = 2.5              # Acc Coefficient Max
        c1_min = 0.5              # Acc Coefficient Min
        c2_min = c1_min

        c2_max = c1_max
        w0 = 0.9                    # Initial Inertia weight
        wf = 0.9                     # Final Inertia Weight
        w =  (w0 - wf) *  ((maxiter - itr) / maxiter) + wf
        c1 = ((c1_min - c1_max) * itr / maxiter ) + c1_max
        c2 = ((c2_max - c2_min) * itr / maxiter ) + c2_min
      for p in range(0,Na):
```

```
            for i in range(0,Np):
                for j in range(0,Nd):
                    self.v[p][i][j] =  w * self.v[p][i][j]+ c1 *
np.random.uniform() * \
                    (self.pbest_position [p][i][j] - self.position
[p][i][j]) \
                    + c2 * np.random.uniform() * (gbest_position
[p][j] - self.position [p][i][j])
                    if self.v[p][i][j] > self.vmax[p] :
self.v[p][i][j] = self.vmax[p]
                    if self.v[p][i][j] < self.vmin[p] :
self.v[p][i][j] = self.vmin[p]
    def update_position(self):
       for p in range(0,Na):
            for i in range(0,Np):
                for j in range(0,Nd):
                    self.position[p][i][j] = self.v[p][i][j] +
self.position[p][i][j] # update position according to pso
    def validate_constraints(self,bounds_max,bounds_min):
        for p in range(0,Na):
            for i in range(0,Np):
                for j in range(0,Nd):
                    if self.position[p][i][j] > bounds_max[p][i][j]:
self.position[p][i][j] = bounds_max[p][i][j]
                    if self.position[p][i][j] < bounds_min[p][i][j]:
self.position[p][i][j] = bounds_min[p][i][j]

# -------Continuous MDPSO------------------------------------------#

class PSO():
      def__init__(self,pr_tou,en_min,en_max,bounds_max,bounds_min,ma
xiter,num_slots,objective_fcn):

        global Nd
        global Na
        global Np
        global mu
        global sigma
        sigma,mu = 0.0 , 0.24
        Nd = num_slots
        Na = len(en_min)
        Np = 100
        self.gbest_cost = np.inf
        self.gbest_position = np.zeros((Na,Nd))     # 1x10
        swarm = Particle(en_min,en_max)
        itr = 0
        self.gbest_pos_store = np.zeros((maxiter,Na,Nd))
        self.gbest_cost_store = np.zeros((maxiter,Na))
        self.pbest_position_store = np.zeros((4,Na,Np,Nd))
        self.current_position_store = np.zeros((4,Na,Np,Nd))
        while itr < maxiter:
            swarm.validate_constraints(bounds_max,bounds_min)
            swarm.evaluate_objfcn(objective_fcn)
```

```
            for i in range(0,Np):
# Update global Best Here
                if swarm.pbest_cost[i] <  self.gbest_cost:
                    self. gbest_cost = swarm.pbest_cost[i]
                    for k in range (0,Na):
                        self.gbest_position[k][:] = \
swarm.pbest_position[k][i][:]
            swarm.update_velocity(self.gbest_position,maxiter,itr)
            swarm.update_position()
            self.gbest_pos_store[itr, :, :] = self.gbest_position
            self.gbest_cost_store[itr] = self.gbest_cost
            itr+=1

*************************End of Class*************************

#------------Class Particle for Binary Version --------------------+
class Particle:
    def __init__(self):
# Initialization of Particles
        self.position  = np.random.randint(2, size=(Na,Np,Nd))
        self.v = np.zeros([Na,Np,Nd])
        self.sigmoid = np.zeros([Na,Np,Nd])
        self.cost = [0] * Np
        self.pbest_cost = [np.inf] * Np
        self.pbest_position = self.position
        self.vmax = 4
        self.vmin = - 4
    def validate_constraints(self,S_app,F_app):
        for r in range(0,Na):
# Handling boundary constraints
           for s in range(0,Np):
               for t in range(0,S_app[r]):
                   if self.position[r][s][t]==1:
self.position[r][s][t]=0
               for t in range(F_app[r],Nd):
                   if self.position[r][s][t]==1:
self.position[r][s][t]=0

    def evaluate_objfcn(self,objective_fcn):
# Update objective function
        self.cost = objective_fcn(self.position)
        for i in range(0,Np):
            if self.cost[i] < self.pbest_cost[i]:
# Update Personal Best Position
                self.pbest_cost[i] = self.cost[i]
# Update Personal Best cost
                for k in range (0,Na):
                    self.pbest_position [k][i][:]  = self.position
[k][i][:]

    def update_velocity(self,gbest_position,maxiter,itr):
      # acceleration coefficients
        c1_max = 2.5
```

```
        c1_min = 0.5
      # acceleration coefficients

        c2_min = c1_min
        c2_max = c1_max
        w0 = 0.9
# Initial inertia weight
        wf = 0.4
# Final Inertial Weight
        w = ((w0 - wf) * (maxiter - itr) / maxiter) + wf
# Linearily decreasing inertia weight
        c1 = ((c1_min - c1_max) * itr / maxiter ) + c1_max
# pbest acceleration coefficients (c1 * r1)
        c2 = ((c2_max - c2_min) * itr / maxiter ) + c2_min
# gbest acceleration coefficients (c2 * r2)
        for p in range(0,Na):
# Update velocity here
            for i in range(0,Np):
                self.v[p][i] = w * self.v[p][i] + c1 *
np.random.uniform() * (self.pbest_position [p][i] - self.position
[p][i]) \
                + c2 * np. random.uniform() * (gbest_position [p][:]
- self.position [p][i])
        for p in range(0, Na):
            for i in range(0,Np):
                for j in range(0,Nd):
                    if self.v[p][i][j] > self.vmax : self.v[p][i][j]
= self.vmax # velocity clamping here
                    if self.v[p][i][j] < self.vmin : self.v[p][i][j]
= self.vmin # velocity clamping here

    def update_position(self,S_app,F_app,r_3):
        for p in range(0,Na):
            for i in range(0,Np):
                for j in range(0,Nd):
                    self.sigmoid[p][i][j] = 1 / (1 + np.exp(-
self.v[p][i][j])) # update position according to bpso
                    if r_3[p][i][j] < self.sigmoid[p][i][j]:
                        self.position[p][i][j] = 1
                     else :
                        self.position[p][i][j] = 0

# ----------Binary MDPSO--------------------------------------------
class PSO():
      def__init__(self,pr_tou,en_app_un,S_app,F_app,num_slots,maxite
      r,objective_fcn):

        global Nd
# scheduling horizon
        global Na
# of appliances
        global Np
# of Particles
        Nd = num_slots
```

```
        Na = len(S_app)
        Np = 100
        self.pr_tou = pr_tou
        self.en_app_un = en_app_un
        self.gbest_cost = np.inf
# global best cost
        self.gbest_position = np.zeros([Na,Nd])
# global best position
        swarm = Particle()
# swarm initialization
        self.convergence_curve = []
# convergence curve initialization
        itr = 0
        self. gbest_pos_store = np.zeros((maxiter,Na,Nd))# Row
vector numappliances X timeslots
        self.gbest_cost_store = np.zeros((maxiter,Na))
      #Column vector
        while itr < maxiter:
            r_3 = np.random.uniform(0,1,[Na,Np,Nd])
            swarm.validate_constraints(S_app,F_app)# Maximum number
of iterations
            swarm.evaluate_objfcn(objective_fcn)
# Evaluate objective function
            for i in range(0,Np):
# Update global Best cost and position here
                if swarm.pbest_cost[i] <  self.gbest_cost:
                    self.gbest_cost = swarm.pbest_cost[i]
                    for k in range (0,Na):
                        self.gbest_position[k][:] =
swarm.pbest_position[k][i][:]
            swarm.update_velocity(self.gbest_position,maxiter,itr)
            swarm.update_position(S_app,F_app,r_3)
            self.gbest_pos_store[itr, :, :] = self.gbest_position
            self.gbest_cost_store[itr] = self.gbest_cost
            itr+=1
*******************************End of the class *******************

""" objective function for elastic Appliances """

def objective_fcn_el(x):
    Na = len(x)
    j = [f_per_particle_el(x[:,i,:],Na)for i in range(Np)]
    return np.array(j)

def f_per_particle_el(x_s,Na):
    eq_cons_ac1 = utility_fcn_ac(x[i])
    eq_cons_ref = utility_fcn_ref(x[i])
    thermal_cons = eq_cons_ac1 + eq_cons_ref
    x = np.sum(x_s,axis=0)
    f_cost = [0] * Na
    dr = [0] * Na
    for i in range(0,Na):
        f_cost[i]= np.dot(x_s[i],pr_tou)
```

```
        dr[i] =
np.subtract(b_state_el[i][ts_dr:tf_dr],x_s[i][ts_dr:tf_dr])
# cost of each appliance * tou_price
    comfort = np.square(np.subtract(x_s,b_state_el))
# (bl_state -sch_state)^2 - comfort
    comfort_sum = np.sum(comfort,axis=0)
# comfort sum over appliances
    f_comfort = sum(comfort_sum)
# comfort sum over time_slots
    dr_sum = np.sum(dr,axis=0)
    f_dr = sum(dr_sum)
    f_total_cost = sum(f_cost) - (theta_dr * delta_rew * f_dr)
    eq_constraint = equality_constraint_el(x_s,en_req_el)
    eq_cost = in_cons_cost(sum(f_cost),unsch_cost_el)
    ineq_cons = in_cons(x,x_max_s)
    j = w1 * (f_comfort/unsch_comfort_el ) + w2 *
(f_total_cost/min_cost_el)  + gamma * (eq_constraint + ineq_cons +
thermal_cons + eq_cost)
    return j
# Utility function for HVAC Loads
def utility_fcn_ac (x):
    utility_theta_max = np.zeros(num_slots)
    utility_theta_min = np.zeros(num_slots)
    tc = XX
    tow = XX
    x_cap = XX
    theta_d = XX
    chi = XX
    epsilon = math.exp(-tow / tc)
    theta_ind = [XX]
    for t in range(1,num_slots+1):
        theta_ind.append(XX * ((epsilon * ((theta_ind[t-1] -
theta_d)/x_cap)) + ( 1 - epsilon) * (((theta_outdoor_1[t-1] -
theta_d)/x_cap) + (chi * (x[t-1]/en_max_thl[0])))) + XX)
    theta_indoor = theta_ind[1:97]
    for i in range(S_app_thl[0],F_app_thl[0]+1):
        if theta_indoor[i] > theta_max[0]:
            utility_theta_max[i] = (theta_indoor[i]-theta_max[0])**2
        elif theta_indoor[i] < theta_min[0]:
            utility_theta_min[i] = (theta_min[0]-theta_indoor[i])**2
    sum_ = sum(utility_theta_max) + sum(utility_theta_min)
    return sum_,theta_indoor

#------------objective function for appliances  ------+
def objective_fcn_inel(x):             # Input argument is binary
decision variable
    Na = len(S_app_inel)
    x_f = np.zeros([Na,Np,Nd])
    for i in range(0,len(S_app_inel)):
        x_f[i] = en_app_inel[i] * x[i]
    j = [f_per_particle_inel(x_f[:,i,:],Na,x[:,i,:])for i in
range(Np)]
    return np.array(j)
```

```
def f_per_particle_inel(x_s,Na,x_b):
    constraint_un_inel = [0] * Na
    cost_inel = [0] * Na
    un_app = len(constraint_un_inel) - 2
    for i in range(0,Na):
        cost_inel[i] = np.dot(x_s[i],pr_tou)
    for i in range(un_app,len(constraint_un_inel)):
        constraint_un_inel[i] = uninterruptible(x_b[i],i)
    x = np.sum(x_s,axis=0)
    comfort = np.square(np.subtract(x_s,b_state_unsch_inel))
    comfort_sum = np.sum(comfort,axis=0)
    obj_comfort = sum(comfort_sum)
    obj_cost = sum(cost_inel) - theta_dr * delta_rew *
sum((np.subtract(b_state_unsch_aggregate[ts_dr:tf_dr],x[ts_dr:tf_dr]
)))
    eq_constraint = equality_constraint(x_s,en_req_inel)
    eq_cost = in_cons_cost(sum(cost_inel),unch_cost_inel)
    ineq_cons = in_cons(x,x_max_1)
    j = w1 * (obj_comfort/unsch_comfort_level) + w2 *
(obj_cost/min_cost_inel)  + gamma * (eq_constraint + ineq_cons +
eq_cost + sum(constraint_un_inel))
    return j

#%%%%%Equality and Inequality Constraints Handling
functions%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
def equality_constraint(x,en_req):
    constraint = 0
    for i in range(0,len(x)):
        constraint += (sum(x[i])-en_req[i])**2
# Energy requirement over complete scheduling horizon
    return constraint
def equality_constraint_el(x,en_req):
    constraint = 0
    Na_c3 = 2
    for i in range(0,Na_c3):
        constraint += (sum(x[i])-en_req[i])**2
# Energy requirement over complete scheduling horizon
    return constraint
def in_cons(x,x_max):
    constraint = 0
    rated_panel = np.sum([x,x_max],axis=0)
    max_rated_panel = max(rated_panel)                          #
Rated power consumption of panel should not exceed a pre-defined
limit
    constraint = max(0,(max_rated_panel - panel_capacity))
    return constraint
# Rated power consumption of panel should not exceed a pre-defined
limit
def in_cons_cost(x,unsch_cost):
    constraint = max(0,(x - unsch_cost))
    return constraint
# Uninterruptiblity constraint
def uninterruptible(x,appnum):
    L_x_6 = L_app_inel[appnum]
```

```
    len_phase = num_slots - L_x_6
    sum_x6 = np.zeros(len_phase)
    for i in range(0,len_phase):
        sum_x6[i] = sum(x[i : i + L_x_6])
    max_x_6 = max(sum_x6)
    constraint = (max_x_6 - L_x_6)**2
    return constraint

#%%%%%%%%This is baseline for inelastic and elastic
appliances%%%%%%%%%%%%%%%%%%%%%%%%%%%%
class base_state():                                            #
class to generate the unscheduled load profile of inelastic
appliances
    def __init__(self,S_app,F_app,L_app):
        self.Na = len(S_app)
        self.b_state = np.zeros([self.Na,Nd])
        for i in range(0,self.Na):
            for j in range(S_app[i], S_app[i] + L_app[i]):
                self.b_state[i][j] = 1
    def get_output(self):
        return self.b_state
class base_state_el():                                         #class
to generate the unscheduled load profile of elastic appliances
    def __init__(self,S_app,len_,load_profile):
        self.Na = len(S_app)
        self.b_state = np.zeros([self.Na,Nd])
        for i in range(0,self.Na):
            self.b_state[i][S_app[i]:len_[i]+S_app[i]] =
load_profile[i]
    def get_output(self):
        return self.b_state
#%%%%%%%%%%%%%%%%%%%%%%Function to set contraint on each time slot
energy consumption for class 3,4,5 ---------------
"""class to generate minimum and maximum bound matrix on each time
slot energy consumption """
class set_bounds():
    def __init__(self,Np,en_max,en_min,S_app,F_app):
        self.Na = len(S_app)
        self.bounds_min = np.zeros([self.Na,Np,Nd])
#minimum energy consumption in each time slot
        self.bounds_max = np.zeros([self.Na,Np,Nd])
# maximum energy consumption in each time slot
        for i in range(0,self.Na):
            for j in range(0,Np):
                for k in range(S_app[i],F_app[i]+1):
                    self.bounds_min[i][j][k] = en_min[i]
        for i in range(0,self.Na):
            for j in range(0,Np):
                for k in range(S_app[i],F_app[i]+1):
                    self.bounds_max[i][j][k] = en_max[i]
    def get_output(self):
        return self.bounds_max,self.bounds_min
```

**********************END OF THE CODE******************************

References

[1] A. Atkeson and P. J. Kehoe, “The transition to a new economy after the Second Industrial Revolution,” Federal Reserve Bank of Minneapolis, 606, 2001.

[2] H. Hondo, “Life cycle GHG emission analysis of power generation systems: Japanese case,” *Energy*, vol. 30, no. 11–12, pp. 2042–2056, Aug. 2005.

[3] C. Greer *et al.*, “NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0,” National Institute of Standards and Technology, NIST SP 1108r3, Oct. 2014.

[4] A. Ipakchi and F. Albuyeh, “Grid of the future,” *IEEE Power Energy Mag.*, vol. 7, no. 2, pp. 52–62, Mar. 2009.

[5] M. Lauby, “Reliability Assessment Guidebook,” p. 83.

[6] C. W. Gellings, “The concept of demand-side management for electric utilities,” *Proc. IEEE*, vol. 73, no. 10, pp. 1468–1470, Oct. 1985.

[7] “2010 Assessment of Demand Response and Advanced Metering - Staff Report,” p. 117.

[8] A. Gomes, C. H. Antunes, and A. G. Martins, “A Multiple Objective Approach to Direct Load Control Using an Interactive Evolutionary Algorithm,” *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1004–1011, Aug. 2007.

[9] Chu, C., Jong, T., & Huang, Y. (2005). A direct load control of air-conditioning loads with thermal comfort control. *IEEE Power Engineering Society General Meeting, 2005*, 664-669 Vol. 1.

[10] N. Ruiz, I. Cobelo, and J. Oyarzabal, “A Direct Load Control Model for Virtual Power Plant Management,” *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 959–966, May 2009.

[11] D. D. Weers and M. A. Shamsedin, “Testing a New Direct Load Control Power Line Communication System,” *IEEE Trans. Power Deliv.*, vol. 2, no. 3, pp. 657–660, Jul. 1987.

[12] A. Faruqui and S. Sergici, “Household response to dynamic pricing of electricity: a survey of 15 experiments,” *J. Regul. Econ.*, vol. 38, no. 2, pp. 193–225, Oct. 2010.

[13] A. Safdarian, M. Fotuhi-Firuzabad, and M. Lehtonen, “A Distributed Algorithm for Managing Residential Demand Response in Smart Grids,” *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2385–2393, Nov. 2014.

[14] G. Lobaccaro, S. Carlucci, and E. Löfström, “A Review of Systems and Technologies for Smart Homes and Smart Grids,” *Energies*, vol. 9, no. 5, p. 348, May 2016.

[15] A. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, “Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid,” *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 320–331, Dec. 2010.

[16] Z. Zhu, J. Tang, S. Lambotharan, W. H. Chin, and Z. Fan, “An Integer Linear Programming Based Optimization for Home Demand-side Management in Smart Grid,” in *Proceedings of the 2012 IEEE PES Innovative Smart Grid Technologies*, Washington, DC, USA, 2012, pp. 1–5.

[17] C. O. Adika and L. Wang, “Autonomous Appliance Scheduling for Household Energy Management,” *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 673–682, Mar. 2014.

[18] A. Agnetis, G. de Pascale, P. Detti, and A. Vicino, “Load Scheduling for Household Energy Consumption Optimization,” *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2364–2373, Dec. 2013.

[19] S. Koch, D. Meier, M. Zima, M. Wiederkehr, and G. Andersson, “An active coordination approach for thermal household appliances — Local communication and calculation tasks in the household,” in *2009 IEEE Bucharest PowerTech*, 2009, pp. 1–8.

[20] N. Li, L. Chen, and S. H. Low, “Optimal demand response based on utility maximization in power networks,” in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–8.

[21] S. Tiptipakorn and W. Lee, “A Residential Consumer-Centered Load Control Strategy in Real-Time Electricity Pricing Environment,” in *2007 39th North American Power Symposium*, 2007, pp. 505–510.

[22] S. Salinas, M. Li, and P. Li, “Multi-Objective Optimal Energy Consumption Scheduling in Smart Grids,” *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 341–348, Mar. 2013.

[23] D. Livengood and R. Larson, “The Energy Box: Locally Automated Optimal Control of Residential Electricity Usage,” *Serv. Sci.*, vol. 1, no. 1, pp. 1–16, Mar. 2009.

[24] H. Bagge and D. Johansson, “Measurements of household electricity and domestic hot water use in dwellings and the effect of different monitoring time resolution,” *Energy*, vol. 36, no. 5, pp. 2943–2951, 2011.

[25] P. Constantopoulos, F. C. Schweppe, and R. C. Larson, “Estia: A real-time consumer control scheme for space conditioning usage under spot electricity pricing,” *Comput. Oper. Res.*, vol. 18, no. 8, pp. 751–765, Jan. 1991.

[26] J. Y. Lee and S. G. Choi, “Linear programming based hourly peak load shaving method at home area,” in *16th International Conference on Advanced Communication Technology*, 2014, pp. 310–313.

[27] F. A. Qayyum, M. Naeem, A. S. Khwaja, A. Anpalagan, L. Guan, and B. Venkatesh, “Appliance Scheduling Optimization in Smart Home Networks,” *IEEE Access*, vol. 3, pp. 2176–2190, 2015.

[28] N. G. Paterakis, O. Erdinç, A. G. Bakirtzis, and J. P. S. Catalão, “Optimal Household Appliances Scheduling Under Day-Ahead Pricing and Load-Shaping Demand Response Strategies,” *IEEE Trans. Ind. Inform.*, vol. 11, no. 6, pp. 1509–1519, Dec. 2015.

[29] F. B. Saghezchi, F. B. Saghezchi, A. Nascimento, and J. Rodriguez, “Quadratic Programming for Demand-Side Management in the Smart Grid,” in *Wireless Internet*, 2015, pp. 97–104.

[30] B. Jeddi, Y. Mishra, and G. Ledwich, “Dynamic programming based home energy management unit incorporating PVs and batteries,” in *2017 IEEE Power Energy Society General Meeting*, 2017, pp. 1–5.

[31] H. Roh and J. Lee, “Residential Demand Response Scheduling With Multiclass Appliances in the Smart Grid,” *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 94–104, Jan. 2016.

[32] W. Saad, Z. Han, H. V. Poor, and T. Başar, “Game Theoretic Methods for the Smart Grid,” *ArXiv12020452 Cs Math*, Feb. 2012.

[33] Zhuang Zhao, Won Cheol Lee, Yoan Shin, and Kyung-Bin Song, “An Optimal Power Scheduling Method for Demand Response in Home Energy Management System,” *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1391–1400, Sep. 2013.

[34] M. Rasheed, N. Javaid, A. Ahmad, Z. Khan, U. Qasim, and N. Alrajeh, "An Efficient Power Scheduling Scheme for Residential Load Management in Smart Homes," *Appl. Sci.*, vol. 5, no. 4, pp. 1134–1163, Nov. 2015.

[35] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, "Particle swarm optimization for integer programming," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, 2002, vol. 2, pp. 1582–1587 vol.2.

[36] "A Comparison of Particle Swarm Optimization and the Genetic Algorithm | Structures, Structural Dynamics, and Materials and Co-located Conferences." [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2005-1897. [Accessed: 20-Jun-2019].

[37] "Demand response for home energy management system | Elsevier Enhanced Reader." [Online]. Available: https://reader.elsevier.com/reader/sd/pii/S0142061515002379?token=BC8F250A5CBFFADA6E0BD81A1D1D1D519E5448376A61D09C3678EBF2F2B113A6C72B2C3C9B16320ABB84B7E714FC8151. [Accessed: 19-Jun-2019].

[38] Yimin Zhou, Yanfeng Chen, G. Xu, Qi Zhang, and L. Krundel, "Home energy management with PSO in smart grid," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, 2014, pp. 1666–1670.

[39] H. Yang, C. Yang, C. Tsai, G. Chen, and S. Chen, "Improved PSO based home energy management systems integrated with demand response in a smart grid," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 275–282.

[40] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948 vol.4.

[41] M. Clerc, "Standard Particle Swarm Optimisation," Sep-2012.

[42] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, Jun. 2004.

[43] "A modified particle swarm optimizer - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/699146. [Accessed: 28-Jun-2019].

[44] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, “Practical distribution state estimation using hybrid particle swarm optimization,” in *2001 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.01CH37194)*, 2001, vol. 2, pp. 815–820 vol.2.

[45] Y. Shi and R. C. Eberhart, “Parameter selection in particle swarm optimization,” in *Evolutionary Programming VII*, 1998, pp. 591–600.

[46] Zwe-Lee Gaing, “Particle swarm optimization to solving the economic dispatch considering the generator constraints,” *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1187–1195, Aug. 2003.

[47] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, “A particle swarm optimization for reactive power and voltage control considering voltage security assessment,” *IEEE Trans. Power Syst.*, vol. 15, no. 4, pp. 1232–1239, Nov. 2000.

[48] Chin Aik Koay and D. Srinivasan, “Particle swarm optimization-based approach for generator maintenance scheduling,” in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS’03 (Cat. No.03EX706)*, 2003, pp. 167–173.

[49] J. Kennedy and R. C. Eberhart, “A discrete binary version of the particle swarm algorithm,” in *Computational Cybernetics and Simulation 1997 IEEE International Conference on Systems, Man, and Cybernetics*, 1997, vol. 5, pp. 4104–4108 vol.5.

[50] U. Wilke, F. Haldi, J.-L. Scartezzini, and D. Robinson, “A bottom-up stochastic model to predict building occupants’ time-dependent activities,” *Build. Environ.*, vol. 60, pp. 254–264, Feb. 2013.

[51] A. Barbato, A. Capone, M. Rodolfi, and D. Tagliaferri, “Forecasting the usage of household appliances through power meter sensors for demand management in the smart grid,” in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2011, pp. 404–409.

[52] “Accounts & Billing - Residential - Rates and Conditions - Hydro Ottawa.” [Online]. Available: https://hydroottawa.com/accounts-and-billing/residential/rates-and-conditions. [Accessed: 26-Jul-2019].