DeLi2P : A User Centric, Scalable Demand Side Management Strategy for Smart Grids

Syed Muhammad Ali*, Mohammad Naveed*, Fahad Javed†*, Naveed Arshad* and Jahangir Ikram*
* Syed Babar Ali School of Science and Engineering, LUMS
Lahore, Pakistan
Email: naveedarshad@lums.edu.pk
†Department of Computer Science, GIFT University
Gujranwala, Pakistan
Email: fahadedupk@gmail.com

***Abstract*—Demand side management (DSM) has the potential to significantly improve smart grid operations by reducing the peak to average ratio. Proposed DSM schemes are reducing peak load by as much as 30% which can translate to significant cost savings and reduction in green house emissions. But for realistic deployment of this system in the grids there are two very important aspects which need to be considered: Scalability and user acceptability. Since the DSM algorithm is required to control potentially hundreds of thousands of devices, the technique has to be scalable and tractable for such numbers. On the other hand DSM will effect the life style of the consumer and the affect of the system should be as less disruptive as possible. The various techniques proposed in literature attempt to first reduce the cost and then attempt to resolve one of the two aspects. The result is that the techniques are either scalable or are only considerate of the deadlines of the consumers. To this end in this paper we present a unique combination of two solution strategies and propose Deli2P , a user centric and scalable solution. In essence we provide to the consumer a deadline centric interface. The deadlines solutions are generally not scalable. But instead of solving this problem as a scheduling for deadline problem we transform it to a priority based mechanism which is scalable but for the user interface and acceptability is not ideal. Our results show that with this scheme we can reduce the peak load to the priority based mechanism without violating the consumers' deadlines.**

***Index Terms*—Smart grids; Demand Side Management, Peak reduction, demand elasticity.**

I. INTRODUCTION

The continuous increase in electricity demand and the shrinking resources of energy has resulted in scarcity of electricity in the existing setup. In such a scenario conserving and optimally consuming the existing resources has gained paramount importance. One of the major ways for efficient energy management is demand side management (DSM) [9]. Coupling DSM with the future smart grids technologies therefore is being seen as the major resource for the future smart grids [15]. The goal of these future DSM systems is to manage the domestic consumer's load for a more environmental and economic efficient energy generation scheduling. This is usually achieved by offloading the electric consumption from high cost timings to low cost or environment friendly timings.

However, for a DSM strategy to be viable for domestic consumer it is imperative that this load movement is acceptable to the consumers schedules and practical constraints. Historically it has been observed that strategies which do not consider consumers preferences as first class requirement fail to deliver to their promises due to stiff resistance or non-cooperation of the consumers. As Kim and Shcherbakova report on the reasons for DSM failures, the consumer needs to be involved in the DSM activities and her requirements need to be understood and catered for [11]. To cater to the consumer's needs there are two strategies used by the DSM planners and researchers: Either consumer's exact requirements are captured by elicited the deadline within which the task much be achieved [1], or the devices are assigned priorities and these priorities are used for planning [3].

The difference in the two strategies on operational level may seem insignificant on first observation but from computational perspective the two strategies have significant implication in the scalability of the system. The preference based scheduling problem is a very common problem in operations research and computer science. Various job scheduling tasks exist in real world and researchers have analyzed this problem in detail. Scheduling of loads constrained by preferences is very similar to job scheduling problem. However, this problem is classified as NP-complete problem and to-date no scalable algorithm has been proposed to solve this problem. The implication of this NP-complete classification is that a problem without necessary transformations will be intractable for large number of devices. As the number of decision points, in this case number of devices, increases the time to compute will increase exponentially.

On the other hand priority based systems are able to aggregate the priority class demands and the planning decision reduces from controlling hundreds of devices under constraints to the frequency of devices in each priority class. Since the number of priority classes is much smaller than number of devices, planning for this system is tractable. ColorPower is examples of such techniques [16]. This reduction in size makes the problem tractable for a micro-grid or even city level scheduling.

However, priority classes are limiting in that deadlines for individual devices are not part of the scheduling. This is the case in the works observed by the authors. This is problematic as this results in lower consumer satisfaction as the needs of the consumer can be violated. From experience in Power7 of UK we know that the consumers life styles are dynamic and such fixed measures are not very useful. To end in this paper we propose a novel transformation of deadlines to priority (DeLi2P) based model.

Deli2P collects user deadlines from devices. The modern electronic devices are fitted with timers to stop or start a device at a specific time in future. Similar to these timers, Deli2P consumer can select the time at which she requires the device's process to be complete by putting in the "putoff" time on the device or device plug interface. For example if a consumer is putting in dishes for washing then the consumer can feed in 6 hours for the dishes to be washed and ready.

Deli2P transforms this deadline into a priority in the following way. For each two hour gap in start time we decrease the priority of the device. That is, if the device is required to complete its operation in 6 hours then the priority level for the device will be yell (Third highest for 2 x 3 hours). With each device assigned a priority based on the time available for execution we can aggregate the priority demands and schedule the device operations in the same way as is done by Ranade and Beal [16]. If the turn to activate the device is not received till the next priority threshold (four hours till deadline in this case) then the priority level of the device is bumped up. With a higher priority level, the device has a higher chance of getting the activation signal. This process is continued till the device is either run or is it at the highest priority level where it is guaranteed execution.

When sufficient supply is available the system allows all the devices to execute as soon as they submit a demand to consume. As the demand grows above the supply, the devices with the least priority are instructed to "wait" for servicing based on a probabilistic model. This ensures that the demand never surpasses the supply. In the early hours of the day the demand is less than supply and all devices are given go ahead for execution but as the wave of demand crosses the supply line our algorithm skims the supply line while the wave of demand grows till the later hours of the day when wave comes crashing down as the overall demand of electricity dies out at later hours of the night. Deli2P in this way is able to reduce the peak demand of the day while satisfying the consumer's preferences. To evaluate our strategy we tested. We used a smart grid city wide device level simulator.

II. Related Work

When we look at the demand side management (DSM) techniques for home consumers we see two goals for the algorithms. A range of algorithms attempt to reduce the cost of electricity for the consumers. These include algorithms which incorporate the time of use pricing in reducing the cost of electricity of the consumer [8][13], and DSM systems which maximizes the benefit of renewable energy sources for home consumer [1][5]. For these systems the goal is not overall supply-demand management but rather it is to minimize the cost of electricity to the consumer. However, this does not guarantee that the demand is shaped according to the global or utility's goals. To achieve this specific goals direct control systems are proposed which aim at reducing the peak load. The difference is that in the first category the goal is cost reduction and peak load reduction is implicit whereas in the second the peak load reduction is the goal and cost savings due to better load profiles are implicit.

The direct control algorithms control thus control the consumer devices remotely. However, shutting down end user device is usually not very acceptable to the end users. This has been studied and expounded by many researchers [11]. To capture the needs of the consumers there are two main processes proposed in the literature.

One stream of research for such DSMs is to gather the preferences or constraints of the consumers. That is, for each device the system predicts or collects from user the range within which the device can be scheduled. These constraint elicitation can be implicit as is the case Du and Lu [6] and Molderink [14] where forecasting and consumer profiling using sensors in the user premises are used to determine the constraints of the consumers. In other cases such as those proposed by Kim and Poor [12] and Arif and colleagues [1], the constraints are explicitly provided by the consumers through some interface. The authors of the systems argue that since we have knowledge of the consumer's constraints and schedule the devices accordingly, the DSM load management will be acceptable to the consumers.

However, as has been discussed by Molderink [14], Arif [1] and Javed in AdOpt [10], such scheduling is NP-complete [17]. To date there has been no polynomial time algorithm to solve the NP-complete problems. The only way to solve such problems is to enumerate all the possible combination which is exponential and thus planning for hundreds of thousands of devices is not possible. Use of artificial intelligence techniques such as genetic algorithms and ant colony optimization decrease the computation time but still take too long for large scale system scheduling and are known to be inaccurate.

The other stream is to allow the users to stipulate priority classes to devices or join the device to a contractual obligation group. ColorPower1 and ColorPower 2 allow the consumers to assign priorities to devices [16], [3]. The system then manages the probability of execution for each priority class such that the demand shaping goals are reached while maintaining that the distribution is fair and according to consumer's priorities. Kim and Poor also proposed similar system however their grading of the devices was not as versatile[12]. Pennywise on the other hand allow users to sign contracts with the utility. Since these contracts are standardized the utility can bunch together devices under similar contracts for computation. Escriv and colleagues also used such contracts for their proposed DSM strategy [7].

Since in these systems there is a natural way to combine the devices into consumption classes, the number of decision variable reduces to the number of consumption classes. For few hundred classes the existing algorithms are able to solve within a reasonable time as is shown by Javed and Arshad in

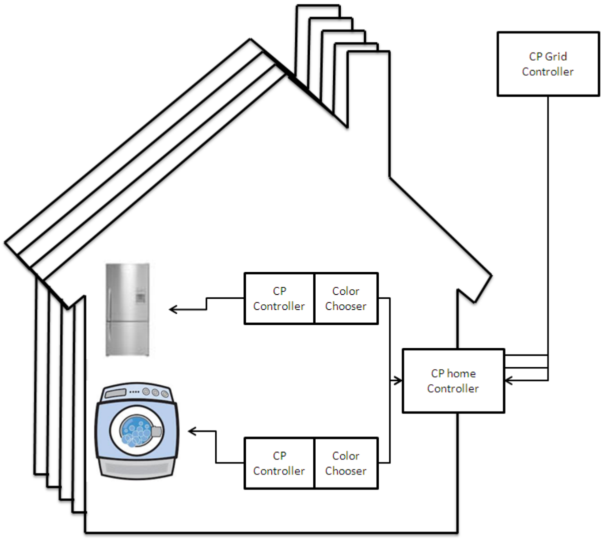

Fig. 1. .

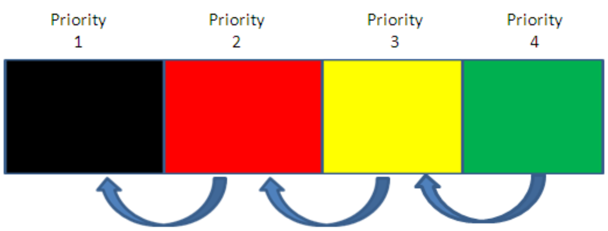

Fig. 2. Priority scale for DeLi2P. As the deadline time approaches the priority color transitions from green to yellow to red and then black, the highest priority level.

AdOpt [10].

However, since the priority systems observed by the authors are somewhat rigid. ColorPower, and Pennywise do not allow the consumers to change the priorities. Colorpower allows consumers to press an emergency button to explicitly demand electricity. However if priority of a device changes then this updation will take 24 hours to come into effect. Similarly re-negotiating contracts for Pennywise or in Escriva's system is a cumbersome task. As has been shown in UK's Power7 package, the consumers do not like to be bounded to certain priorities and contracts. As a biological being, the priorities and needs for the human consumer changes rapidly and regularly. The priorities for consumers are rarely consistent and fixed contracts do not do justice either.

To end in this paper we propose a unique transformation to convert a simple deadline based user feedback into a priority based system. Since we collect explicit preference and bound the system by it, we can reap the benefits of better user acceptability. Since we convert these preferences into priorities very similar to the ColorPower scheme, we can compute the solution in reasonably small time as well getting use the benefit of both the systems.

$$\Delta_i^{g-} = \begin{cases} 0 & \text{if } C^g \geq 0 \text{ or } i > b \\ |\hat{EF}_i| & \text{else if } \sum_{j\leq i} |\hat{EF}_j| \leq |C^g| \\ |C^g| - \sum_{j<i}|\hat{EF}_j| & \text{else if } \sum_{j<i}|\hat{EF}_j| < |C^g| \\ 0 & \text{otherwise} \end{cases}$$

Fig. 3. ColorPower [3] controller for device.

$$p_{off,i,a} = \frac{\Delta_i^{g-}+\Delta_i^{p-}+\Delta_i^{c-}}{|EF_i|} \; ; \; p_{on,i,a} = \frac{\Delta_i^{g+}+\Delta_i^{p+}+\Delta_i^{c+}}{|DF_i|}$$

Fig. 4. ColorPower [3] controller for the grid.

III. Proposed Strategy

The goal of DSM is to control consumer devices to reduce the electricity demand to specific thresholds. As previously discussed, there are two non-functional requirements of the DSM system, first is that it should be scalable and second it should be bound by the needs of the consumers. In this regard the priority based algorithms provide adequate solution for the first requirement. For instance, ColorPower is scalable, provides privacy preserving aggregation method and is fair in its distribution of electricity.

The algorithm operates in the following way. The consumer assigns each device a color specifying its priority as shown in figure 2. At every heartbeat the home controller collects its demand from devices of each color. If two devices are yellow then it will add the demand of both devices as yellow. The algorithm provides for a grid wise aggregation mechanism whereby the grid controller has the aggregated demands for each of the four priority colors. The grid controller then assigns a probability to each priority color based on the supply demand equation. The policy is that the highest priority gets to use the supply till its demand is met. If supply is left then the lower priority is provided the supply. If for a priority level the supply is partial then each device in the priority is given a probability based on the amount of electricity available and the demand of that color. This way the lower priority devices run on a probabilistic basis.

However, this scheme does not provide provisions for the consumer to constraint the load movement according to consumer's operations. If a consumer purchases a washing machine then she would wish to have the clothes washed within a specific period of time and if the algorithm does not provide this guarantee then the consumer will be tempted to bump up the device's priority to get the required service. This may result in spiraling up of priorities thereby leaving no space for optimization.

DeLi2P uses the same efficient, scalable and fair ColorPower (CP) controllers to provide control of the device to the utility planner. The main contribution is that instead of consumer assigning priorities, we make these priorities adaptable in that the device controller adapts the devices' color

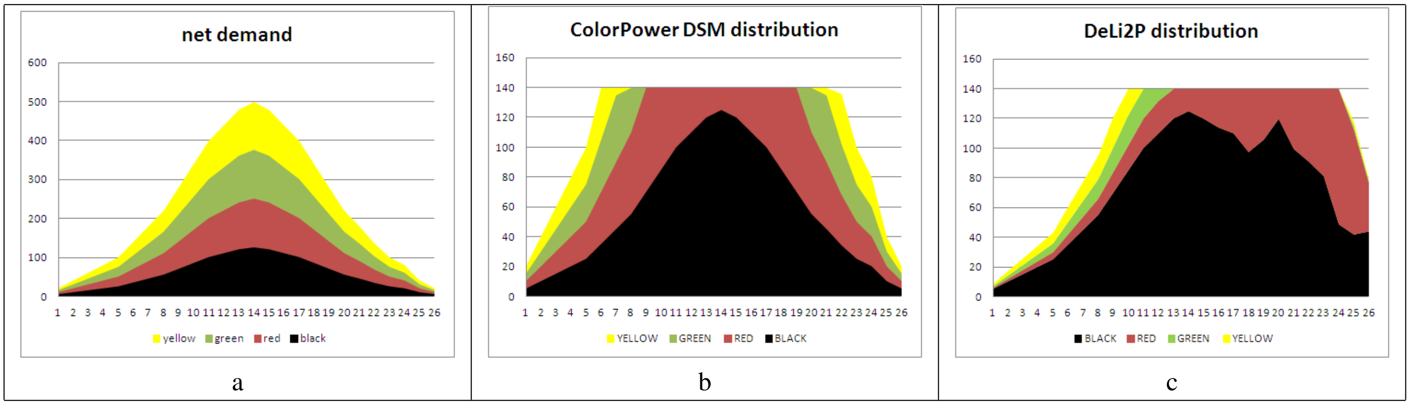

Fig. 5. Graphs showing net demand, demand shaping through ColorPower and through DeLi2P.

according to the amount of time it has to complete the task.

The difference is that when a consumer attempts to use a device, the consumer is provided with a timer to set the time of task completion. This can be achieved by installing a timer on the device. This maybe an external timer for legacy machine or it maybe an internal timer in the future smart devices however, discussion of its deployment is beyond the scope of this work. The consumer sets the time when she requires the task to be completed. Based on the time to deadline DeLi2P calculates the color of the device at runtime using the following formula:

$$C_i = \frac{\lceil (d_i - o_i) - curr \rceil}{len(k)}$$

Where d_i is the deadline set by the user, o_i is the operational execution time or the maximum time it will take the device to complete the task and $curr$ is the current time. C_i is the priority color of the device at time $curr$. $len(k)$ is the length of interval that we give to each color. In our case it is 120 minutes.

As the time moves forward, that is the difference between $(d_i - o_i) - curr$ reduces, the priority of the device increase thereby increasing its chance to complete the task. The operational flow is similar to ColorPower except for the color assignment After each two minute period the color for the device is ascertained. If the consumer has asked for the device to operate and the operation has not yet started then the color is propagated to the CP home controller along with historical consumption value. The CP home controller aggregates the demand for each color, adds it to the data passed by previous house and passes it forward to next house so that it is transmitted to the CP Grid controller as shown in figure 1 and discussed in [16]. The CP grid controller based on its algorithm shown in figure 4 calculate the fractional part for each color. This value is passed to the CP Device controller and the device controller selects the state of the device using formula in figure 3

To illustrate the process, let us assume that a consumer attempts to use washing machine. The consumer wants the operation to be completed in 8 hours time and the operational execution time for the machine is 1 hour. Thus we have (8 - 1) hours to complete the task. This puts the device in yellow or category 3. Based on the global demand the CP grid controller assign probabilities to the colors and these probabilities are propagated to the CP device controllers. If at this time the supply is sufficient to supply yellow priority then this device will run immediately. But if yellow is partially supplied or not supplied at all then this device will use the CP controller using formula in figure 3. If the device gets a chance to run then it will execute otherwise it will wait till C_i is red. With an updated color the device's probability of execution will chance. If the supply is so short that even the red does not execute then after 2 hours the device will turn black. Since black is emergency color the device will be provided with supply and the task will be achieved within the stipulated time.

To illustrate on grid let us consider the demand in figure 5(a). This is hypothetical demand where each of the priority level is equally divided. This is the demand that was assumed by ColorPower authors for their validation [16]. The second figure 5(b) shows the response of ColorPower and similar algorithms. As can be seen the demand is flat lined at demand of 140 units. This may result in a yellow device to be unavailable for as much as 20 hours. To ameliorate this situation in figure 5(c) the response of DeLi2P is shown. As the time moves forward DeLi2P elevates the priority of yellow, green and red. This elevation of priority means that the device has the opportunity to run within the deadline constrain set by the consumer. the deadline will fail only when a part of black is above the threshold line. This in essence is the breaking point of the algorithm, that is, if we require to shut down devices in black then the deadlines for those consumers will be violated. In comparison ColorPower does not cater for this dimension hence evaluating it for deadline failure is not possible.

In this section we have first described how are strategy will consider consumers deadlines. Then we have discussed how we incorporates existing ColorPower algorithm for our planning. We close this section with description of how DeLi2P will impact the consumer and how it will impact the global demand shape and shown how the load shedding targets will be achieved without violating the consumer preferences. In the next section we will discuss the experimental setup and

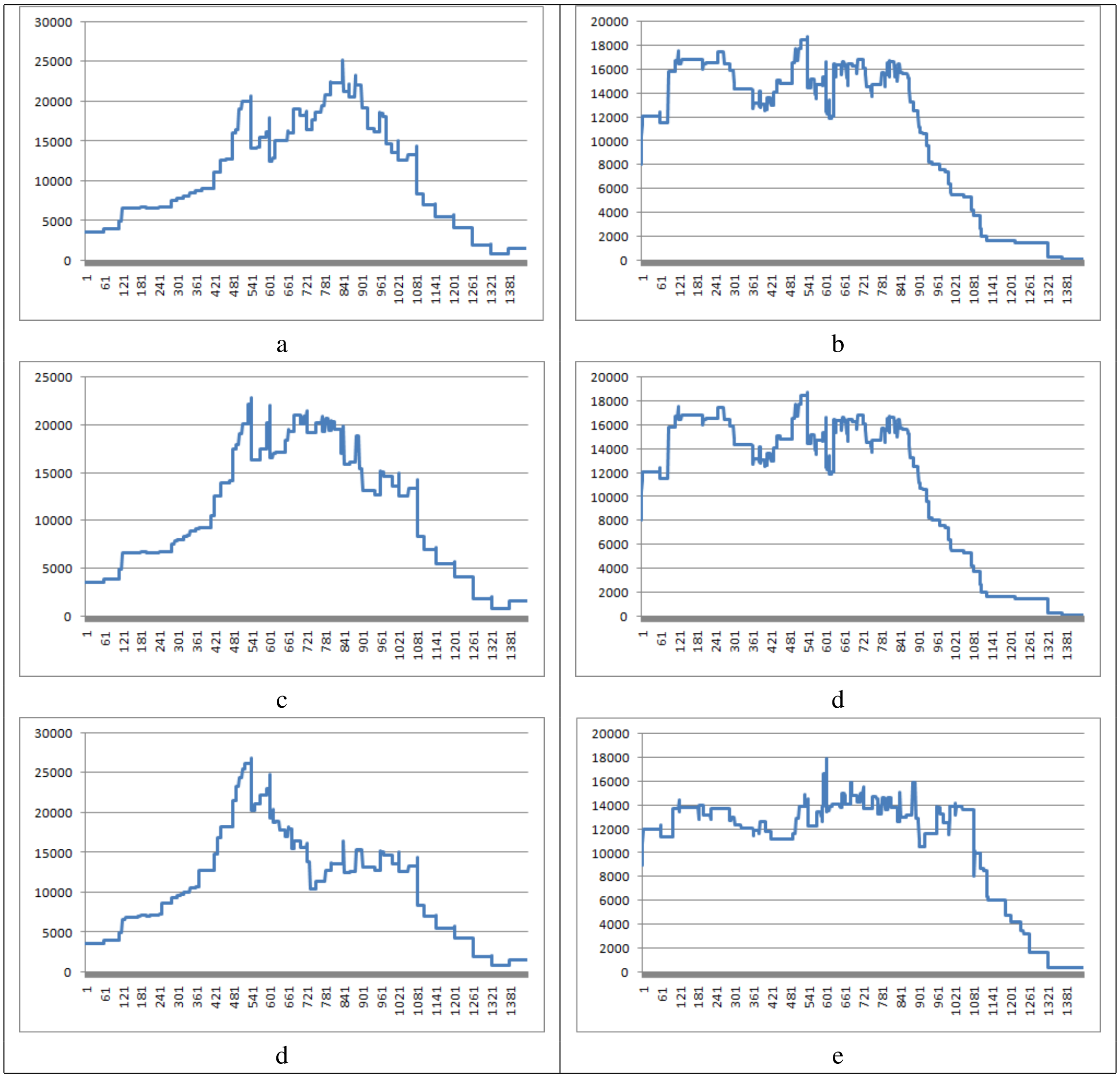

Fig. 6. Graphs showing demand and DSM through DeLi2P results. On the left the graphs show the demand for the day and on the right the demand after application of DeLi2P. The peak demand ranges from 26MW on day 3 to 23MW on day 2. on all these days DeLi2P has restricted the peak demand to 18MW without violating consumer deadlines

our results on the simulation data.

IV. Problem Model

In this section we model the demand of electricity according to its priorities. The time in our algorithm is divided into t time slices and we have j priorities. Demand of electricity for j priority class generated in a time slice t is given as:

$$\forall_j d_t^j = \text{Demand for } j^{th} \text{ priority at } t \text{ time}$$

For ColorPower, this will also be net demand or D_t^j of the respective time slice and the priority class since there is no concept of carrying over load which has not been provided supply. But for DeLi2P, load which is not provided supply is considered is carried over thus net demand at time t for j priority class is:

$$\forall_j D_t^j = d_t^j + (D_{t-1}^j \times p_{t-1}^j + \Delta_t^{j-1}) - \Delta_t^j$$

Here Δ_t^{j-1} is the demand for the lower priority level which has expired and needs to elevated to the next level and Δ_t^j is the expired demand from j^{th} level which needs to be elevated to the $j+1^{th}$ level.

p_t^j is the probability of selection for the j^{th} priority at time t. DeLi2P and ColorPower use the same formulation for calculating p_t^j gieven as:

$$p_t^j = 1 \text{ if } \sum_{i=j}^{max(j)} D_t^i \leq S$$

$$p_t^j = 0 \text{ if } \sum_{i=j+1}^{max(j)} D_t^i \geq S$$

$$p_t^j = \frac{S - \sum_{i=j+1}^{max(j)} D_t^i}{D_t^j} \text{ otherwise}$$

Where S is the total supply to the system.

The graphs in figure 6 show this transformation of priority level induced by the transfer of Δ load from load priority levels to higher levels. Though the probability assignment in ColorPower and DeLi2P is same, the underlying total demand D_t^j varies in DeLi2P to incorporate the concept of deadline but in ColorPower, the deadline is not accommodated.

V. Experimental Setup

To carry out analysis, simulations for the proposed algorithm and related work were developed in C Sharp programming language. These simulations were conducted on Intel i5 Processor with 4GB of physical memory having 2.4GHz of clock speed. Power consumption data for 100 different devices was used in simulations for extensive testing of algorithms. The data is generated at the rate of 2 samples / minute by the configurable simulator presented in [2]. The aggregated consumption of each device at a articular time window was used to compute average power for that specific appliance. The average powers were then put to use in demand response algorithms corresponding to the time window for which they were computed to obtain results close to the actual environment.

VI. Results

In this section we show the results of applying DeLi2P over the house devices in 200 houses. As described earlier, when a consumer attempts to use a machine and the system asks the consumer to provide a deadline by which time she requires the task done. Based on the available time the algorithm assigns a priority level. Based on the overall supply-demand equation, the device is run when based on probability provided by the grid controller the device controller has the chance to run. In case if the algorithm was not present then the device will run at the very minute the consumer attempts to use the machine.

The results of application of DeLi2P are shown in figure 6. The graphs on the left show the demand of the system without DeLi2P and on the right the results are with DeLi2P with 18000 KW (18MW) as the target maximum load. As can be seen, DeLi2P is able to maintain this target even though the demand ranged from 25 MW on day 3 to 23 MW on day 2 while satisfying the deadlines set by the consumers.

VII. Conclusion and Future Work

The need to incorporate consumer's priorities and requirements in to demand side management system is critical to its acceptance and thereby its use. In this paper we have considered two key requirements for increasing the acceptability of the consumer. First we have provided a way for the consumer to be an active part of the DSM. This provides a satisficing feeling to the consumer as proposed by Breukers and colleagues [4]. Secondly, we have provided a way for the consumer's deadlines to be catered for in a scalable manner by converting those deadlines to priority levels and thereby making the algorithm scalable, secure and fair as well.

We see this as a very viable solution for future DSM systems. The future of this work is to apply it to a residential area and observe the consumer's response to this strategy. A second path of research is to incorporate machine learning to learn consumer behavior in order to automate the timer task in order to aid the consumer. Another avenue is to incorporate time of use pricing in this system such that the consumer is informed of the price savings that she can achieve by setting the timer to a later time.

References

[1] Ahmer Arif, Fahad Javed, and Naveed Arshad. Integrating renewables economic dispatch with demand side management in micro-grids: a genetic algorithm-based approach. *Energy Efficiency*, pages 1–14, 2013.

[2] Naveed Arshad, Usman Ali, and Fahad Javed. A highly configurable simulator for assessing energy usage. *Energy Procedia*, 42:308–317, 2013.

[3] Jacob Beal, Jeffrey Berliner, and Kevin Hunter. Fast precise distributed control for energy demand management. In *Self-Adaptive and Self-Organizing Systems (SASO), 2012 IEEE Sixth International Conference on*, pages 187–192. IEEE, 2012.

[4] SC Breukers, E Keiskanen, B Brohmann, RM Mourik, and CFJ Feenstra. connecting research to practice to improve energy demand-side management (dsm). *Energy*, 36(4):2176–2185, 2001.

[5] Liu Daoxin, Li Lingyun, Chen Yingjie, and Zeng Ming. Market equilibrium based on renewable energy resources and demand response in energy engineering. *Systems Engineering Procedia*, 4(0):87 – 98, 2012. ¡ce:title¿Information Engineering and Complexity Science - Part II¡/ce:title¿.

[6] Pengwei Du and Ning Lu. Appliance commitment for household load scheduling. *Smart Grid, IEEE Transactions on*, 2(2):411 –419, june 2011.

[7] Guillermo Escriv-Escriv, Isidoro Segura-Heras, and Manuel Alczar-Ortega. Application of an energy management and control system to assess the potential of different control strategies in hvac systems. *Energy and Buildings*, 42(11):2258 – 2267, 2010.

[8] Zhong Fan. Distributed demand response and user adaptation in smart grids. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 726 –729, may 2011. Internet traffic method for DR. Modeling user preference as willingness to pay. Simulation based. no rebound calculations.

[9] C.W. Gellings. The concept of demand-side management for electric utilities. *Proceedings of the IEEE*, 73(10):1468 – 1470, oct. 1985.

[10] Fahad Javed and Naveed Arshad. Adopt: An adaptive optimization framework for large-scale power distribution systems. In *Proceedings of the 2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, SASO '09, pages 254–264, Washington, DC, USA, 2009. IEEE Computer Society.

[11] Jin-Ho Kim and Anastasia Shcherbakova. Common failures of demand response. *Energy*, 36(2):873 – 880, 2011.

[12] T.T. Kim and H.V. Poor. Scheduling power consumption with price uncertainty. *Smart Grid, IEEE Transactions on*, 2(3):519 –527, sept. 2011.

[13] Jang-Won Lee and Du-Han Lee. Residential electricity load scheduling for multi-class appliances with time-of-use pricing. In *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pages 1194 –1198, dec. 2011. Scheduling algorithm using time of use to reduce cost. No rebound in simulation.

[14] A. Molderink, V. Bakker, M.G.C. Bosman, J.L. Hurink, and G.J.M. Smit. Management and control of domestic smart grid technology. *Smart Grid, IEEE Transactions on*, 1(2):109 –119, sept. 2010.

[15] F. Rahimi and A. Ipakchi. Demand response as a market resource under the smart grid paradigm. *Smart Grid, IEEE Transactions on*, 1(1):82 –88, june 2010. Introductory case building paper.

[16] Vinayak V. Ranade and Jacob Beal. Distributed control for small customer energy demand management. In *SASO'10*, pages 11–20, 2010.

[17] Jeffrey D. Ullman. ¡ i¿ np¡/i¿-complete scheduling problems. *Journal of Computer and System sciences*, 10(3):384–393, 1975.