Designing an Adaptive Lighting Control System for Smart Buildings and Homes

Conference Paper · April 2015

DOI: 10.1109/ICNSC.2015.7116079

Yuan Wang

Arizona State University

Designing an Adaptive Lighting Control System for Smart Buildings and Homes

Yuan Wang
Arizona State University
Tempe, AZ, USA
Email: Yuan.Wang.4@asu.edu

Partha Dasgupta
Arizona State University
Tempe, AZ, USA
Email: partha@asu.edu

***Abstract*—Lighting control in smart buildings and homes can be automated by having computer controlled lights and blinds along with illumination sensors that are distributed in the building. However, programming a large building light switches and blind settings can be time consuming and expensive. We present an approach that algorithmically sets up the control system that can automate any building without custom programming. This is achieved by making the system self calibrating and self learning.**

This paper described how the problem is NP hard but can be resolved by heuristics. The resulting system controls blinds to ensure even lighting and also adds artificial illumination to ensure light coverage remains adequate at all times of the day, adjusting for weather and seasons. In the absence of daylight, the system resorts to artificial lighting. Our method works as generic control algorithms and are not preprogrammed for a particular place. The feasibility, adaptivity and scalability features of the system have been validated through various actual and simulated experiments.

I. Introduction

Work environments (and homes) benefit from having even and adequate lighting in spaces that are occupied. Lighting control in large buildings can be challenging to automate, specially as blinds and lights have to be custom programmed for building architecture, geography, weather conditions, seasons and so on. This paper presents an approach to self learning, adaptive lighting control that is not preprogrammed or having a-priori information about the building. Further, such systems can save energy by reducing the use of artificial lighting during daytime hours and unoccupied spaces.

Integrating daylight and artificial lighting in automated system can be challenging. Natural lighting is not stable, even at a fixed location; sunlight's impact varies during times of a day, weather changes, seasons and so on. Our system harvests daylight and then fills in the deficiencies using artificial lights, with attention to provision of even lighting and avoiding light that is too bright (or has glare).

A complete lighting control system contains two interacting modules: daylight control module and artificial lighting control module. The aims of these two modules are different. Daylight control module is mainly used for reducing energy costs while artificial lighting control module is good for providing a more comfortable working environment. Automating and balancing the lighting control system such that uniform and stable lighting is maintained at occupied locations while energy consumptions are kept as low as possible turns out to be a hard problem.

In this paper, a Wireless Sensor Network(WSN)-based lighting control system is introduced. We use static lights that can be turned on/off by the system and venetian blinds on windows, whose angles can be set by the system. The control system is not custom programmed for the environment, i.e. it does not know which light switch controls which light, which blind setting affects which window or even the physical location of rooms and walls. Thus the system is self calibrating, and adaptive to changes in outside lighting, weather, seasons and so on.

The formal model of the problem leads to non-linear integer programming and proves to be NP-Hard. We use a heuristic lighting control algorithm and show that it solves the problem well and efficiently (and is competitive with the optimal solution).

The rest of this paper is organized as follows. Section II introduces the related work of the problem. Section III formalizes the lighting control problem using mathematical model. Section IV presents the heuristic algorithms for lighting control. Experimental work and simulation results are discussed in section V. We conclude the paper in section VI and talk about some future improvements of the system as well.

II. Related Work

To make use of natural sunlight is called daylight harvesting. Electrical blinds are set up at each window and are controlled by feedback systems that adjust blind angles based on daylight levels. It is used in some of the lighting control systems such as [1], [2], etc. for energy savings. Studies show that daylight harvesting can save lighting energy up to 77% [3]. The idea behind is to make use of the sunlight if applicable when light level is not sufficient [1].

The above method can adapt to environmental changes but it usually needs a long adjustment cycle until the blind settings are finally set up, which results in users not liking too frequent blind movements or hunting. [4] proposed a technique called SunCast that can better predict sunlight values by using historical data and approximate simulation results. The drawback of this system is that every time ambient environment or building patterns change, the system needs to collect historical data and rebuild the mathematical model, leading to delays of many months.

WSN technologies have been applied into various areas such as [5]. It consists of portable wireless sensor motes such as Crossbow's TelosB or MICAz to monitor the values of physical conditions, such as temperature, light, humidity, and so on. WSN data collections use several specific protocols such as Collection Tree Protocol [6].

Some customized lighting control systems are targeted for special cases. [7] is designed for theater arts area and [8] was mainly designed for entertainment and media production. Some systems use occupancy sensors [9] to switch off lights in unoccupied positions. [10] presented a mathematical model for lighting control problem in which a luminary impact is continuous such as light emitting diodes(LEDs) luminaries rather than discrete values, and the expected illumination level is given as a single value rather than a range. [11] used smart illuminance sensors with infrared ray communication technology to retrieve the lighting ID binding with each lighting fixture. Through analyzing lighting ID information, sensors can recognize nearby luminaries, which is helpful for systems to know the group information for each actuator.

III. Formal Model

In this model we have a set of switches that control arbitrary lights (one per switch) and a set of blind control switches (each sets angle of one blind)and a set of light sensors. We have established that illumination is additive, that is, the impact of two light bulbs is the sum of the impact of each at a fixed point.

Assume a place has n light switches, n' blind switches and m light-level sensors, placed by the human designer of the place. The sensors are connected to the control system via a WSN and the switches are activated via actuators connected to the system. The physical locations of lights, blinds, sensors and their correlations are initially unknown to the control system.

The ultimate task is to compute the positions of switches (lights and blinds). Let $x = \langle x_1, ..., x_n \rangle$ denote the assignment for light switches where $x_i \in \{0, 1\}$ and 0 denotes off and 1 denotes on, $b = \langle b_1, ..., b_{n'} \rangle$ denote the assignment for automatic blinds where each blind has finite settings and 0 denotes fully off, i.e., let B to be the set of discrete blind settings, then $b_i \in B$ and $b_i = 0$ indicates blind i is fully off.

The goal is to optimize the energy $E(x, b)$ and the comfort $C(x, b)$. $E(x, b)$ can be measured by the number of artificial lights on. For $C(x, b)$, there are two criteria needed to satisfy, lighting level and lighting uniformity. The former is satisfied when every sensor reading stays in an accepted range i.e. if sensor j's reading is $L_j(x, b)$, then $min \leq L_j(x, b) \leq max$. The latter can be satisfied by minimizing the standard deviation(represented by σ) of the sensor readings. Hence the problem can be stated as:

$$\begin{aligned}
\underset{x,b}{\text{minimize}} \quad & \langle E(x,b), \sigma(L_1(x,b), ..., L_m(x,b)) \rangle \\
\text{subject to} \quad & min \leq L_j(x,b) \leq max,\ j = 1, ..., m \\
& x_i \in \{0,1\},\ i = 1, ..., n \\
& b_i \in B,\ i = 1, ..., n'
\end{aligned} \tag{1}$$

By applying the ϵ-constraint method designed for solving multi-objective optimization problems, the new objective function can be defined as:

$$\begin{aligned}
\underset{x,b}{\text{minimize}} \quad & \langle \sigma(L_1(x,b), ..., L_m(x,b)) \rangle \\
\text{subject to} \quad & min \leq L_j(x,b) \leq max,\ j = 1, ..., m \\
& E(x,b) \leq \epsilon \\
& x_i \in \{0,1\},\ i = 1, ..., n \\
& b_i \in B,\ i = 1, ..., n'
\end{aligned} \tag{2}$$

One important feature for lighting impacts is sensor readings are additive, i.e. let $Impact_{i_j}$ to be the impact of light i on sensor j when only light i is on, and $Impact_{k_j}^{b_k}$ to be the impact of daylight on sensor j when only blind k is on and set to a particular setting b_k, then we have

$$\begin{aligned}
& L_j(x,b) = \sum_{i=1}^{n} Impact_{i_j} \cdot x_i + \sum_{k=1}^{n'} Impact_{k_j}^{b_k} \cdot f(b_k) \\
& f(b_k) = \begin{cases} 1 & b_k \neq 0 \\ 0 & b_k = 0 \end{cases}
\end{aligned} \tag{3}$$

Note for a specific light i, $Impact_{i_j} (\geq 0, \in \mathbb{Z})$ is always a constant and for a specific short time period and a fixed blind setting, we assume $Impact_{k_j}^{b_k} (\geq 0, \in \mathbb{Z})$ also to be a constant.

[12] and [13] gave a general definition of nonlinear integer programming problem. It can be stated as:

$$\begin{aligned}
\text{max/min} \quad & f(x) \\
\text{subject to} \quad & h_i(x) = 0,\ i \in I = 1, ..., p \\
& g_j(x) \leq 0,\ j \in J = 1, ..., q \\
& x \in \mathbb{Z}^n
\end{aligned} \tag{4}$$

where x is a vector of decision variables, and some of the constraints $h_i, g_j : \mathbb{Z}^n \rightarrow \mathbb{R}$ or the objective function $f : \mathbb{Z}^n \rightarrow \mathbb{R}$ are non-linear functions. In equation (2), let $g_1(x, b) = L_j(x, b) - max, g_2(x, b) = E(x, b) - \epsilon, h_1(x, b, y) = L_j(x, b) - min - y = 0, y \geq 0, y \in \mathbb{Z}$. It can be transformed to:

$$\begin{aligned}
\underset{x,b}{\text{minimize}} \quad & \langle \sigma(L_1(x,b), ..., L_m(x,b)) \rangle \\
\text{subject to} \quad & g_1(x,b) \leq 0 \\
& g_2(x,b) \leq 0 \\
& h_1(x,b,y) = 0 \\
& x_i \in \{0,1\},\ i = 1, ..., n \\
& b_i \in B,\ i = 1, ..., n' \\
& y \geq 0, y \in \mathbb{Z}
\end{aligned} \tag{5}$$

Since the equivalent version of equation (2) equation (5) satisfies the format of equation (4) where the objective function and partial constrains($g_1(x,b)$ and $h_1(x,b,y)$) are nonlinear, our problem belongs to nonlinear integer programming problem. According to [12], it is NP-hard. Therefore, any polynomial time computed solution would be an approximation. To find the best setting, a naive approach is to try all $2^{n+n'}$ positions for $n+n'$ switches, which is unacceptable due to the time complexity. Therefore, we propose a heuristic algorithm for computing an approximate optimal combination of blind and light settings. This paper focuses on the system control during the day. In the evening when sunlight doesn't exist, artificial lighting becomes the only lighting source and the detailed control approaches can be referred to the method described in [14].

Conceptually, for better saving energy, during the day, daylight will be considered first to provide illuminance. Artificial lights will be used to even and compensate the lighting when necessary. Detailed control approaches will be discussed in section IV.

IV. Control Approaches

A. Calibration

In an earlier publication[14] we described the calibration procedure for artificial lights. The calibration of the blinds follow a similar method and hence we outline it. While there is no daylight, artificial lights are calibrated by turning on one switch at a time and noting the sensor readings or sensors where this light has impact. From this data, we compute zones (or rooms where each light is located). From the impact measures, we are then, heuristically able to decide which light switches should be on for the lighting to be even, even when there is daylighting.

In this section, we describe the calibration steps for electrical blinds, which is used to calculate $Impact_{i_j}^{b_i}$(a.k.a. blind i's impact on sensor j at angle b_i). We assume there is no any other lighting source involved except daylight at this stage. Specifically, suppose there are n' blinds in an area, each blind has k settings. The process of calibration is:

1) Turn off all blinds and check existing values on each sensor
2) Turn on one blind up to a particular angle at a time
3) Calculate the blind's impact on each sensor(sensor value - existing value)
4) Repeat step 2 $n' \times k$ times until all settings are counted

It is clear that the above calibration steps run in linear time($O(kn')$) and the space complexity is ($O(kmn')$). Compared to the exhaustive search solution which runs and records all the combinations of blind settings(complexity is $O(k^{n'})$), our calibration largely saves both time and space complexities. All impact data will be stored in database for further processing.

B. System Control Workflow

The goal is to compute desired blind and light settings for a time point T. The proposed lighting control approach contains three main stages: blind prediction stage, blind adjustment stage and artificial lighting control stage. Blind prediction will be initiated first to compute predicted blind settings based on predicted calibration datasets. Predicted settings will be passed into blind adjustment stage for adjusting electrical blinds in real time. Artificial lighting will be used when inside lighting level is either not sufficient or not distributed evenly.

The system relies on the calibration datasets stored in database. Those datasets will be passed into data collection and modeling server for further processing. This server is doing some preliminary work, for example, classfication. In this paper, we assume that the historic data used for blind settings computation at particular day has to be picked up from the category of that day. For example, computations in sunny days use the historic data from sunny days. To build a more precise computational model, the system also collects some conditions from outside world such as comfort conditions. New generated results(settings) will be put into database, which are collected by the system itself for better learning scale factors and creating predicted datasets. The system therefore can be viewed as a learning-based closed-loop control system, which will become more robust and precise when dataset is becoming larger.

Figure 1 reflects the system control workflow. Detailed methodologies operated at three main stages are discussed below.

C. Blind-Prediction

Blind prediction relies on the predicted calibration datasets generated by data collection and modeling part. It applies a scale factor(offset) onto the historic calibrated data for similar days and produces the one for blind prediction use.

1) Count the minimum number of blinds needed to be turned on: suppose $C = \{C_i^{b_i}\}, i \in [1, n']$ where

$$\begin{aligned} b_i &= \arg\max_{x \in B} C_i^x \\ C_i^x &= \sum_{j=1}^{m} Impact_{i_j}^x \cdot f(x) \end{aligned} \tag{6}$$

To reach this goal, simply find minimum number of elements in the sorted array C such that sum of them are greater or equal than $lowerbound(m \times min)$.

2) Base-level candidates are computed: From step 1, it is known at least w blinds is needed. Therefore, each base-level candidate setting should have at least w elements selected to be turned on, and the total contribution on sensors under each candidate should be greater or equal than $lowerbound$ value. To solve the problem, it needs w iterations. At each iteration, an unselected blind is picked up whose setting(i, b_i) is generated based on:

$$C_i^{b_i} >= \frac{lowerbound - existing}{w - iteration}, iteration \in [0, w-1]$$

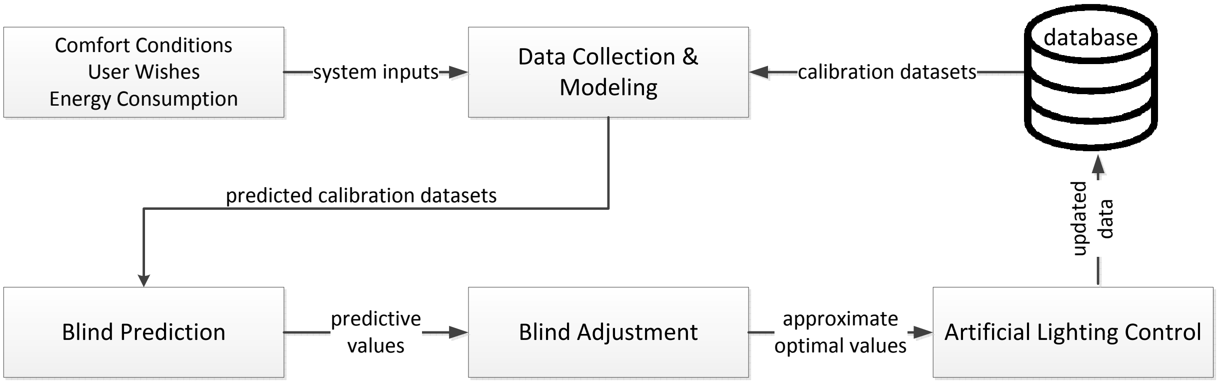


Fig. 1. System Control Workflow

3) More candidates are generated from base-level ones: Since adding δ blinds to compute needs $O(n'^{\delta})$ time, to ensure low response time, we set $\delta = 2$. Suppose each base-level candidate has w elements which have non-zero settings, there are n' blinds in total, each blind has k settings. Then if each candidate wants to pick up an unselected blind, there would be $k(n' - w)$ choices. If there are s base-level candidates, finally there would be $s \times k \times (n' - w)$ new candidates that have $w + 1$ elements which have non-zero settings. Similarly when another unselected blind is trying to be picked up, total amount of candidates that have $w + 2$ elements which have non-zero settings is becoming $s \times k \times (n' - w) \times k \times (n' - w - 1)$. Thus total number of candidates would be $s + s \times k \times (n' - w) + s \times k \times (n' - w) \times k \times (n' - w - 1)$. Note for each candidate q, $\sum\limits_{p \in q,} C_p \leq upperbound(m \times max)$ where $C_p = C_i^{b_i}$ and $p = (i, b_i)$.

4) Standard deviation of sensor readings generated by each candidate is calculated: According to equation 3, for each candidate c, we are able to calculate its impact on sensor j $L_j(c)$(note there is no artificial lighting at this point). Then it is easy to know the standard deviation of all sensors' readings under c. The candidate that generates the lowest standard deviation of sensor readings would be selected as the final blind setting, at the blind prediction stage.

Suppose there are n' blinds in total. Since step 1 has a time complexity $O(n')$, step 2, step 3 and step 4 all have a time complexity $O(n'^2)$, *Blind-Prediction* has a time complexity $O(n'^2)$. Compared to feedback system, our approach predicts and finds the approximate blind settings in a very short time without having physical blind movements, which results in more satisfactory feelings to users.

D. Blind-Adjustment

Since the calibration data used in the prediction stage is predicted based on historical data, there exists errors at the prediction stage. The adjustment step is added into the control plan to reduce the offset in real time. To minimally reduce the number of blind movements, each blind(if selected) will be adjusted to maximum impact angle(when lighting is not sufficient) or minimum impact angle(when lighting is beyond expectation) only. Algorithm 1 describes the steps of blind adjustment. Specially, $setting$ denotes the predicted blind setting generated by blind prediction stage, lu_r denotes the real sensor readings under $setting$, lu_p denotes the predicted sensor readings under $setting$. With the blind-adjustment step, the system is able to adapt the environmental changes more quickly and make corresponding adjustments more properly compared to a pure learning system.

Algorithm 1 Blind Adjustment

```
Input: $setting$, $lu_r$, $lu_p$
Output: new blind settings
$scale = \lambda \times (lu_r \ / \ lu_p)$
for $(i, b_i) \in setting$ do
    $max = \max_{x \in B} C_i^x \times scale, min = \min_{x \in B} C_i^x \times scale$
    $C_i^{b_i} = C_i^{b_i} \times scale$
    $diff_{max} \leftarrow |C_i^{b_i} - max|$
    $diff_{min} \leftarrow |C_i^{b_i} - min|$
end for
if $lu_r < lowerbound$ then
    choose values from $diff_{max}$, sum of the values $\geq$ offset
else
    if $lu_r > upperbound$ then
        choose values from $diff_{min}$, sum of the values $\geq$ offset
    end if
end if
adjust blind angles based on the chosen values
```

E. Artificial Lighting Involved into the System

The biggest advantage of involving blind control into the system is saving lighting resources. However, artificial lighting cannot be fully ignored in some circumstances. When sunlight is not sufficient, artificial lighting is required for completing the inside lighting levels. When there are variations among sensor readings, artificial lighting is needed to reduce them. According to the objective function described in section III, the selection of value ϵ introduced in equation 2 needs to be adjusted based on real environments. Since computational time is important and low time complexity is desired, ϵ is always set to be minimum required lighting + 2 artificial lights(unless there is a user-specific requirement), and thus the total time complexity would be no more than $O(N^2)$(N is number of actuators). The extension work described in [14] detailedly describes how to adjust artificial lights to get a more even and balanced environment.

V. Experimental Work and Simulation Results

A. Implementation Details

To demonstrate the feasibility and effectiveness of our approach, we used an experimental setup. A 8ft × 8ft test cell was instrumented with 9 lights ($15W$ incandescent, $120V$), two automated blinds and 9 sensors connected via a WSN and actuators to a computer [14]. The blinds are placed on two windows and communicate with the control server through serial port. They can be controlled by slave commands sent from the server. Each blind can be adjusted from 0 degree(fully off) to 90 degree(fully on). Blind control programs are written in Java.

B. Feasibility

The experiment is run during the day. It is to verify the feasibility of our proposed lighting control algorithm. Control strategies are based on the contents introduced in section IV. One special sensor is put on the window to detect the offset (scale) between the experimental day's sunlight level and the previous days. Blinds' impacts on sensors at previous days are stored in the database. By applying the scale factor, a predicted calibration dataset is obtained for blind prediction. Min value is set 100. Max value is set 130.

To check the difference between our computed results and optimal ones, we first run our proposed method at 8 am. Then with the same configurations, we run a brute force algorithm checking all combinations of all actuators using real data rather than predicted ones. Sensor readings for both experiments are recorded. We repeat this experiment at other time points like 10am, 12pm, 2pm and 4pm.

The results are shown in Figure 2. From the results we can see compared to the brute force, the proposed approach has a similar light intensity performance and a very short increase on standard deviation.

C. Adaptivity and Scalability

Our previous paper [14] has shown the artificial lighting control part satisfies the adaptivity and scalability features, that is when either pattern changes or amount of lights is increased, the system is still able to compute settings in a reasonable time. Now we need to study these two features for the blind control part of the lighting control system. To this end, we perform simulations on a more complex, synthetic setup, with randomly generated impact values.

Suppose there are n' blinds, n lights, m sensors, each blind has k settings. To match the scenario in the previous real experiment, we set $n = 9$, $m = 9$ and $k = 7$. Historic calibration dataset A will be given in the following way: to each blind, every setting's impact on a specific sensor m is randomly given an integer value from 1 to 100. The scale factor γ is set to be a random decimal value from 1 to 2, which matches the factor used in our real experiments. The real data is also derived from the dataset A, with a scale value in the range of 1 to 2 as well. Existing illumination level on each sensor is randomly set from 1 to 50. If blind i's total impact on sensors is $blind_i$, then the average impact of a blind is $\sum_{i=1}^{n'} \frac{blind_i}{k}/n'$. $lowerbound$ is selected as $n'/2$ × average impact, $upperbound$ is selected as $(n'/2 + 3)$ × average impact. Each artificial light's impact on sensor is randomly set between 1 to 100, same to the configurations described in [14].

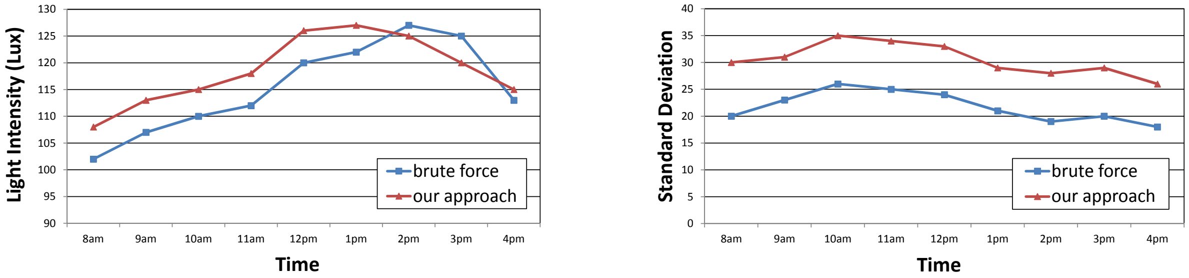
TABLE I
Lighting Level Comparison between Brute Force Method and Our Proposed Method(LUX)

# of Blinds	Min	Max	Brute Force	Proposed Method
8	175	306	181	294
9	191	298	260	269
10	219	350	292	236
11	229	333	309	318
12	246	369	296	340
13	277	383	298	364
14	298	426	365	403

We did 7 groups of simulation experiments with an increased number of blinds. We compared the results getting from brute force with our approach. TABLE I describes the average lighting levels generated by brute force method and our proposed method for each simulation experiment. It indicates settings computed by our proposed lighting control algorithm can generate the impact values that fit into the acceptable range. Figure 3 shows the standard deviation and computational time results for brute force method and our proposed method respectively. From the results, we can see the standard deviation of the proposed method is nearly 1.5 - 2 times as compared with the optimal solutions while computational time is far less than the latter one. Compared to the brute force, the proposed approach has an extremely better time performance with a similar light intensity performance and a reasonable increase on standard deviation. In other words, the blind control part of the lighting control system satisfies the adaptivity and scalability features.

VI. Conclusion and Future Work

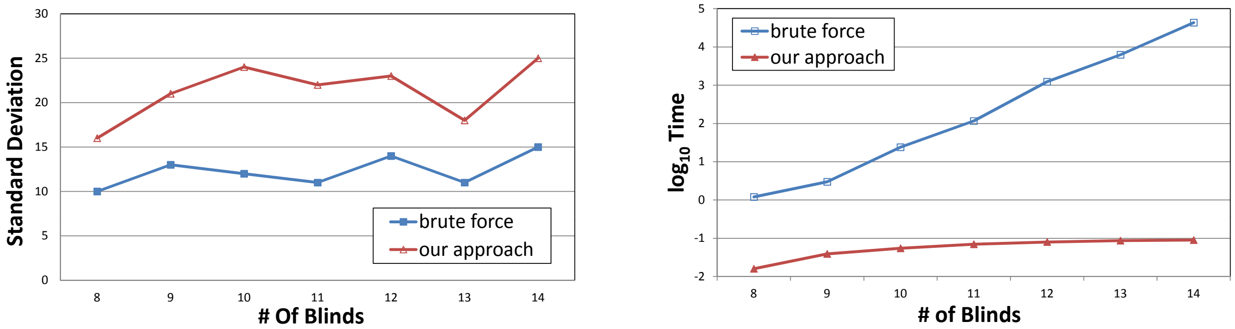
To enable automated lighting control, under varying conditions of occupancy, weather, seasons and other lighting influences it is essential to have a complete system that is effective and adaptive. Such system must be deployable in a simple, cost effective system without the need for customizations and reprogramming as conditions change. It also needs to create a comfortable environments with energy savings as a goal. This paper presents such a complete core system that contains both daylight harvesting module and artificial lighting control module. It also tests the feasibility and effectiveness in both experimental and simulated scenarios. The underlying system can be deployed in buildings and homes without



(a) Light Intensity

(b) Standard Deviation

Fig. 2. Evaluation of System Feasibility



(a) Standard Deviation

(b) Computational Time

Fig. 3. Evaluation of System Adaptivity and Scalability

excessive costs. The lighting control problem is built using non-linear integer programming model, which is NP-hard. A heuristic algorithm is proposed to solve the problem to compute approximate optimal solutions.

Future improvements to the lighting control system include the investigation of data classification, sensor placement and scale factor learning. Current data classification only consider seasons (such as summer or winter) or weather (such as sunny or cloudy) conditions. More conditions such as daylight illumination level, geographical information need to be collected for better classification. With the use of the system, more data will be collected in the database. These data can be used to better predict the scale factors. The results will be applied into the current system control plan. When dataset is made larger, the system will become more robust and precise.

References

[1] S. Matta and S. Mahmud, "An intelligent light control system for power saving," in *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2010, pp. 3316–3321.

[2] V. Singhvi, A. Krause, C. Guestrin, J. Garrett Jr, and H. Matthews, "Intelligent light control using sensor networks," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, 2005, pp. 218–229.

[3] P. Ihm, A. Nemri, and M. Krarti, "Estimation of lighting energy savings from daylighting," *Building and Environment*, vol. 44, no. 3, pp. 509–514, 2009.

[4] J. Lu and K. Whitehouse, "Suncast: fine-grained prediction of natural sunlight levels for improved daylight harvesting," in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM, 2012, pp. 245–256.

[5] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 88–97.

[6] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2009, pp. 1–14.

[7] C. Feng, L. Yang, J. W. Rozenblit, and P. Beudert, "Design of a wireless sensor network based automatic light controller in theater arts," in *Engineering of Computer-Based Systems, 2007. ECBS'07. 14th Annual IEEE International Conference and Workshops on the*. IEEE, 2007, pp. 161–170.

[8] H. Park, J. Burke, and M. B. Srivastava, "Design and implementation of a wireless sensor network for intelligent light control," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 370–379.

[9] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse, "The smart thermostat: using occupancy sensors to save energy in homes," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 211–224.

[10] A. Schaeper, C. Palazuelos, D. Denteneer, and O. Garcia-Morchon, "Intelligent lighting control using sensor networks," in *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*. IEEE, 2013, pp. 170–175.

[11] M. Miki, A. Amamiya, and T. Hiroyasu, "Distributed optimal control of lighting based on stochastic hill climbing method with variable neighborhood," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*. IEEE, 2007, pp. 1676–1680.

[12] R. Hemmecke, M. Köppe, J. Lee, and R. Weismantel, "Nonlinear integer programming," *arXiv preprint arXiv:0906.5171*, 2009.

[13] Wikipedia. Nonlinear programming.

[14] Y. Wang and P. Dasgupta, "Designing adaptive lighting control algorithms for smart buildings and homes," in *Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference on*. IEEE, 2014, pp. 279–284.