Particle Swarm Optimization for Load Balancing in Green Smart Homes

Hector M. Lugo-Cordero*, Abigail Fuentes-Rivera*, Ratan K. Guha *, and Eduardo I. Ortiz-Rivera †
*Department of Electrical Engineering and Computer Science
University of Central Florida, Orlando, Florida 32816
Email: {hlugo, guha}@eecs.ucf.edu, abigail.fuentes@ece.uprm.edu
†Department of Electrical and Computer Engineering
University of Puerto Rico-Mayagüez Campus, Mayagüez, Puerto Rico 00681
Email: eortiz@ece.uprm.edu

***Abstract*—Particle Swarm Optimization (PSO) is a promising evolutionary algorithm, which has been used in a wide range of applications, due to its simple implementation, fast convergence, parallel behavior, and versatility in working with continuous and discrete domains. In this paper, we consider its application to the load balancing problem, in green smart homes. Specifically, an adapted version of the Binary PSO has been used to determine the optimal distribution of energy resources, across different green energy sources in a green smart home. The case study of interest considers the usage of solar and wind energy, as green energy sources for the green smart home. Results demonstrate the effectiveness of the algorithm, in terms of the optimal outcome (efficient distribution of energy resources), finding installation material surplus, and the execution speed of the algorithm.**

I. Introduction

Evolutionary approaches have arisen in the field of evolutionary computation, for a wide variety of optimization problems and applications. Such methods are characterized by their ability to handle non-differentiable, non-linear multimodal functions, their inherent parallelizability, their ease of use, and their good convergence properties. A prominent evolutionary approach used to solve global optimization problems is the Particle Swarm Optimization (PSO) algorithm [1]. Such algorithm has been employed efficiently in different applications, such as speaker identification based on audio and facial characteristics, image segmentation for detecting objects of interest, and in the areas of wireless network and power engineering [2], [3], [4]. The usage of this evolutionary algorithm in different applications is due to the following exceptionally good characteristics that PSO possesses: non-complex implementation, fast execution, parallel behavior, and does not require the optimization problem to be differentiable.

This paper focuses on the application of PSO for load-balancing in smart homes powered by green energy. The last years, energy conservation has become of great interest; with the demand of energy and oil price rising [5]. To meet the energy demand, by causing the least changes to consumers economics, many green energy sources like solar energy and wind energy, have been proposed and studied.

For solving the problem of efficient energy resources distribution, a Binary PSO (BPSO) can be adapted for this application [6]. The original BPSO algorithm is used for problems that are binary in nature, in which each possible solution of a combination of different factors, is encoded as a binary string of 1s and 0s; hence, for the particular application of smart homes, such algorithm can be adapted in order to determine the optimal distribution of energy resources, as the factors.

The modified BPSO algorithm is employed to balance the load distribution in a house, powered by different energy types (e.g. solar, wind, traditional). The load balancing should be done in such a way that the electric bill of the house would be lowered, such a goal is achieved by maximizing the energy consumed at the green energy source.

To support the presence of simultaneous green energy sources, the algorithm is enhanced with the following: 1) multi-objective fitness functions are defined as the addition of individual fitness functions, and 2) two dynamic populations are used to keep mutual exclusion of appliances and their connected sources. Results show, that PSO can have an effective and fast convergence on the load balancing problem, independently of the location of the home. Furthermore, the PSO can help detect surplus of installation materials, such that the consumer can decide if removing the surplus is in need.

The rest of the paper is organized as follows: section II provides background information on the PSO. Section III defines the optimization problem of interest, and how the BPSO can be adapted to solve such a problem. Later in section IV, we elaborate on the daily events formulation and generation, considering the location and time of the house under study. Section V provides the simulation results and discussion. Finally section VI closes the paper with some remarks and future work.

II. Background

This section presents some general background on the PSO algorithm, supporting its use on several applications. Then some current approaches for smart grids are presented.

A. Particle Swarm Optimization

The original particle swarm optimization technique (PSO) was introduced by Kennedy and Eberhart in [1] . Such method updates each particle belonging to a population S, defined as

978-1-4244-7835-4/11/$26.00 ©2011 IEEE

$X_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iD} \end{bmatrix}$ for dimension D, by adjusting its velocity v_{id} in each dimension d, for each new iteration $t+1$:

$$v_{id} = v_{id} + c_1 \cdot rand_1 \cdot (p_{id} - x_{id}) + c_2 \cdot rand_2 \cdot (pgbest_{1d} - x_{id});$$

where p_{id} represents the best local position recorded, for each dimension, and $pgbest_{1d}$ corresponds to the best global position obtained by the whole population, in dimension d. The two positive constants c_1 and c_2 correspond to the cognitive and social learning rates, respectively. The new particle position for the next generation in dimension d, is thus obtained by adding the adjusted velocity to its current position:

$$x_{id} = x_{id} + v_{id}$$

A random inertia weight ω was then added to the velocity update equation, in order to control the impact on previous velocities to current velocities:

$$v_{id} \leftarrow \omega \cdot v_{id} + c_1 \cdot rand_1 \cdot (p_{id} - x_{id}) + c_2 \cdot rand_2 \cdot (pgbest_{1d} - x_{id})$$

In this way, the balance between global exploration and local exploration is affected. Global exploration helps for searching new areas in the provided space; while local exploration fine-tunes the search in the current area. A larger inertia weight ω provides greater global search abilities, while a smaller inertia weight leads to greater local search abilities.

The pseudo-code for this algorithm, using inertia weights (Original PSO), is shown in Algorithm 1.

Algorithm 1 Original PSO

```
Init population S of N particles X_i , in search space dimension D
Init particles velocities V_i to 0
Init individual best P_i to current population
Init Pgbest ← min_{X_i}{F(X_i)}
while Generations Remain less than G_max or Value to Reach Not Met do
  for Each X_i in S do
    Access each dimension of X_i
    for Each d = 1 to D do
      Obtain random inertia weight w_id in the interval (0,1)
      Adapt particle velocity and position in dimension d
      v_id ← ω · v_id + c_1 · rand_1 · (p_id − x_id) + c_2 · rand_2 · (pgbest_1d − x_id)
      Check Velocity Limit [Vmin, Vmax]
      x_id ← x_id + v_id
      Check Particle Limit [Xmin, Xmax]
    end for
    Update best local position vector P_i
    if f (X_i) ≤ f (P_i) then
      P_i ← X_i
    end if
    Update local best position vector P_i
    if f (X_i) ≤ f (P_i) then
      P_i ← X_i
      Update best global position vector Xgbest
      if f (P_i) ≤ f (Xgbest) then
        Pgbest ← P_i
      end if
    end if
  end for
end while
```

B. PSO Applications

The PSO algorithm has been used successfully in training Hidden Markov Models, for speech recognition [7]. The PSO algorithm has also been used to determine the optimal subset of audio and image features, for speaker identification based on voice and face recognition, in [3], [8], respectively. Facial characteristics and human voice, have been combined in a sigle classifier, for multimodal speaker identication in [2], where PSO is used to optimize the combined subset of audio and image features. PSO has also been utilized for other image applications, such as image segmentation, for the purpose of extracting meaningful objects from an image [9]. This evolutionary algorithm has also been successfully employed in wireless networks, with the purpose of finding the optimal path that an attacker may use [4]. In the area of power engineering, PSO has been used for designing optimal power system stabilizers, which are used to damp out local and inter-area oscillations [10].

In this paper, we are interested on employing the PSO for the load balancing in Green Smart Homes problem. We believe that the PSO can provide an effective solution, to ease up the demmand of energy consumption [5]. The following subsection describes existing smart grid approaches.

C. Existing Smart Grid Approaches

There are several characteristics that a smart grid in a Green Smart Home should have: self-healing, high reliability, energy management, and real-time pricing [11]. This paper concentrates mainly on the energy management issue. Other works in literature however, have worked on defining how smart appliances need to be [12].

Green energy, particulary solar and wind, have been studied previously [13], [14], [15], as alternatives for reducing energy costs. To incorporate such sources into traditional homes, concepts from networking have been common options, including TCP/IP [16] and peer-to-peer networks [17], due to their high scalibility and low cost solutions. These approaches have been suggested by the smart grid as a network of computers and the power infrastucture [18]; furthermore the telephone may can be integrated to such an architecture to provide control options for the grid [19].

III. Load Balancing Algorithm with PSO

This section presents how PSO can solve the load balancing problem, using a multi-objective approach. First, we define the optimization problem; following, the modified implementation of the PSO algorithm, which in combination with the smart home cognition, can provide promising performance.

A. The Optimization Problem

It is of great importance to distribute loads properly, such that one can obtain the highest profit out of the smart home system. Thus, we define the optimization problem as follows:

Given a set of appliances $A = \{a_1, a_2, a_3, ..., a_i\}$, where each appliance consumes a total of w_i watts. Such appliances may be connected to the traditional power grid, or a green energy source, but not both simultaneously. The capacity of each energy source its assumed to be W_k for the green energies, and ∞ for the traditional grid. Simiraly, we neglect the normal usage cost for the green energy source, only the installation costs are considered. The cost of the traditional grid is based on the number of kilowatts per hour consumed, and although the price varies per region, it is common that such price is relatively high. Hence, our objective is to distribute the load across the green energy sources evently, such that the price of using the traditional grid is minimized.

To achieve such minimization, it is sufficient to select the optimal combination of appliances for each green source, such that

$$F_k = \sum_{i\epsilon|a_i|} a_i \cdot w_i < W_k, a_i\epsilon\{0, 1\} \quad (1)$$

is maximized. Each green energy source its constrained to a total power provision of W_k, where W_k depends on the location of the smart home. Given the constrains, previously defined, the appliances in each source are mutually exclusive, that is, no appliance may be connected to more than one green energy source.

Equation 1 resembles the the well-known combinatorial $0/1$ Knapsack problem [20], with $k = 1$. However as k increases, the the complexity of the problem increases fast enough, because it transforms into a Multidimensional Knapsack problem, with the additional constrain that the capacity of the knapsacks are not equal. This causes problems, because the Knapsack solution has to be found for each dimension, and their solutions are not mutually independent.

The Knapsack problem has a solution of the form

$$m(i) = max\{w + w_i|a - a_i, w|a\} \quad (2)$$

Using the concept of dynamic programing, this problem can be expressed as a matrix on which the equation 2 is evaluated for $1 \leq W \leq W_k$. Given that the range of total watts in practical green energy sources is on the order of kilowatts, this yields problems because most of the entries on such a matrix are computed unnecessarily, increasing the complexity of the problem.

Hence a new solution is needed, for which we have proposed to use Evolutionary Computation, particulary the Particle Swarm Optimization. The next subsection covers the advantages of such an algorithm, and its requirements.

B. Two Dimensional Binary Particle Swarm Optimization

The PSO [1] has been chosen for load balancing optimization in green smart homes. In particular, the BPSO variation can avoid unncessary computations, having fast convergence rate [6]. BPSO is an extension of the original PSO, on which the parameters to be optimized are of binary nature, with non-continuous fitness functions. The BPSO is a generational evolutionary algorithm (EA) which utilizes simple arithmetic to minimize a given fitness function. The fitness function does not need to be differentiable, as for traditional gradient descent EAs. Each particle in the generation, is represented by a binary string $(b_1b_2b_3...b_n)$, which for our problem, are the appliances in set A. These particles are updated (rather than removed), using its individual velocity v, and evaluated according to the fitness function. After evaluation they are recorded if they surpass the currently known results. The constants c_1 and c_2 (see algorithm 2, line 7) represent the importance of the individual knowledge (i.e. exploitation) and the global knowledge (i.e. exploration) respectively. The BPSO monitors both velocity and position, as opposed to canonicals EAs, which use only the position of the particles. Lines 8-11 are to prevent sigmoid saturation at line 13.

Hence, the items in set A are selected to be powered by a green energy source or not (i.e. 1 or 0 respectively). When an appliance is selected on a green energy source, it stops demanding energy from the traditional power grid, and works using only the green energy source for as long as possible. To avoid high maintenance costs, our design does not consider the use of batteries; hence, once the green energy source can't supply the demand, the appliance switches back to the traditional power grid, until the green energy source becomes available again.

We consider an environment where an appliance may be connected to at most one green energy source, plus the traditional grid. Morever, two fundamental modifications have been made to the original BPSO (see algorithm 2) to adapt it for this situation:

1) **N-Population:** To maintain mutual exclusion of appliances among green energy sources, we consider a population of possible solutions, for each source. Thus, when optimizing particles, each population has the same size; one particle may be selected from each population, such that each particle is evaluated at a different fitness function. Furthermore, in the event that an appliance is selected on more than one particle, it can be easily detected and corrected, using simple AND/OR (lines 17-22) logic functions, such that the appliance may be selected on one of the two sources or none (i.e. both are connected to traditional grid).
2) **Additive Fitness Functions:** Multiple fitness functions are considered, one for each green energy source. To compare solutions the total fitness is calculated as the summation of all fitness functions, evaluated at their respective particles. Fitness functions that pass over the maximum energy supplied, by the source which they are mapped to, are set to 0. For maximizing with the original BPSO, which is a minimization algorithm, the total fitness is multiplied by -1.

Hence, the load balancing in green smart homes has the form $B^N \rightarrow R$, where the dimension N of the binary set B comes from the number of green energy sources available, and the selection of different particles are mapped to a real number in R, using the fitness function approach described above. Particulary, we consider only solar energy and wind energy, having a total fitness function $F = -(F_{solar} + F_{wind})$. Each fitness F_{source} is defined as

$$F_{source} = \begin{cases} \sum_{i \epsilon M} b_i \cdot w_i & \text{if result} \leqslant \max(source) \\ 0 & \text{if result} > \max(source) \end{cases}$$

where M is the number of appliances considered.

Algorithm 2 2D-BPSO Algorithm

Init population of size P randomly
Init particles velocities to 0
Init individual best to current population
Init global best to $min_{particle}\{F(particle)\}$
while Generations Remain *AND* Value not reached **do**
  for Each Particle P **do**
    $v \leftarrow \alpha \cdot v + c_1 \cdot rand_1 \cdot (individual\_best - particle) + c_2 \cdot rand_2 \cdot (global\_best - particle)$
    if $v > 4$ **then**
      $v \leftarrow 4$
    end if
    if $v < -4$ **then**
      $v \leftarrow -4$
    end if
    for Each bit **do**
      if $rand < \frac{1}{1+e^{-v}}$ **then**
        $bit \leftarrow 1$
      else
        $bit \leftarrow 0$
      end if
    end for
    if $P_1 \& P_2$ *NOT 0* **then**
      Generate random integer $index \epsilon Z_3$
      if $index = 0$ **then**
        $P_2 \leftarrow P_2 \wedge 1$
      else if $index = 1$ **then**
        $P_1 \leftarrow P_1 \wedge 1$
      else
        $P_1 \leftarrow P_1 \wedge 1$
        $P_2 \leftarrow P_2 \wedge 1$
      end if
    end if
    Evaluate particle
    if $F(particle) < F(individual\_best)$ **then**
      $individual\_best \leftarrow particle$
    end if
    if $F(particle) < F(global\_best)$ **then**
      $global\_best \leftarrow particle$
    end if
  end for
end while

IV. Experimental Setup

A grammar has been created to compose different senarios, based on a series of actions A with a duration D and units U. An example of such a sequence can be seen in listing 1, for a small period of time (i.e. 1 day). Such a format is repeated for the 365 days in the year, using a stochastic process. In general scenarios are composed of tuples of the form $(A, D, U, timestamp)$ (see equation 3).

$$\begin{aligned} A &= \{WORK, COOK, EAT, READ, \\ &\quad WATCH, SLEEP, TRAVEL, \\ &\quad VACATION\} \\ D &\ \epsilon\ R \\ U &= \{DAYS, HRS, MINS\} \end{aligned} \quad (3)$$

Listing 1. Fragment of Scenario

```
WORK 12 HRS 08/26/2010 21:00
COOK 32 MINS 08/26/2010 21:32
EAT 25 MINS 08/26/2010 21:57
READ 107 MINS 08/26/2010 23:44
WATCH 201 MINS 08/27/2010 03:05
SLEEP 5.92 HRS 08/27/2010 09:00
```

The scenarios also take into consideration the details of house location and day of year. For example, a day at 9:00am at summer will be hotter than one at winter; depending on the location the length of days also varies as the year progresses. For completeness, we present the equations used by the simulation to model the energy availability, which have been extracted from literature [21].

Solar energy depends on the location of the house on the globe, in latitude lat and longitude lng coordinates. This location, combined with the time of the day AST, can help to determine if an event is taking place at daytime. The apparent solar time AST, which is the local time LST with an additional correction due to the location, can be computed using

$$\begin{aligned} \lambda &= \frac{2}{15} arccos(-tan(lat)tan(\delta)) \\ \delta &= 23.45 sin(\frac{360}{365}(284 + N_{day})) \\ AST &= LST + ET \pm 4(SL - LL) - DST \\ ET &= 9.87 sin(2B) - 7.53 cos(B) - 1.5 sin(B) \\ B &= \frac{360}{364}(N_{day} - 81) \end{aligned}$$

where ET is the equation of time, λ is the length of the day in hours, and $delta$ and B are correction factors that depend on the number of the day in the year N_{day}. LL and SL denote the local and standard longitude at the location. When solar panels are designed, the engineers test and rate them to a maximum energy value. On average, a good solar panel may have a 15% of environmental loss. Hence, the amount of energy supplied by the solar system to the house, at any time, can be known as

$$E_{solar} = \begin{cases} 0.85 N_s P_s & \text{if } 12 - \frac{\lambda}{2} \leqslant AST \leqslant 12 + \frac{\lambda}{2} \\ 0 & \text{otherwise} \end{cases}$$

Similarly, we consider the use of wind energy, which can complement the solar system at nightime, and be modeled by

$$E_{wind} = 0.5 \cdot \rho \cdot A \cdot C_p \cdot V^3 \cdot \eta_g \cdot \eta_b$$

TABLE I
EVENTS AND APPLIANCES MAXIMUM CONSUMPTION (WATTS)

Event	AC	Fridge	Range	TV	Lights
WORK	9200	800	0	0	0
TRAVEL	9200	800	0	0	0
VACATION	9200	800	0	0	0
COOK	9200	800	12200	0	240
EAT	9200	800	0	0	240
READ	9200	800	0	0	240
WATCH	9200	800	0	125	240
SLEEP	9200	800	0	0	0

where A is the rotor swept area, and ρ is the air density (i.e. $1.255\ \frac{kg}{m^3}$). For good designs, the coefficient of perfomance C_p can be assumed as 0.35, and the efficiencies for the generator η_g, and bearings η_b can be assumed as 0.5 and 0.95 respectively.

Furthermore, the scenarios also simulate vacation and work time. WORK is done daily, unless users are in vacation time. A typical VACATION has a probability in a year of around $\frac{2}{52}$ weeks. The probability of continuing on the same state (i.e. WORK or VACATION) decreases over time.

V. RESULTS AND DISCUSSION

In this section we show the results obtained by using the 2D-BPSO, presented in section III, for balancing loads in different types of homes: traditional, green, and smart. Ten one-year scenarios experiments at Orlando, Florida, and Mayagüez, Puerto Rico, were conducted, in Java. Different random seeds were generated, with the purpose of providing distinct lifestyle patterns. The virtual smart home simulated was a house of 1640 square feet. The television, refrigerator, air conditioner, lighting system were the devices considered for these experiments. Their corresponding ranges with their average normal power consumptions listed in table I. During the events specified by table I, the devices are active. I. A maximum consumption of 0W indicates that the device is inactive. The fridge and TV were connected to the solar energy, for the green energy cases. In the case of wind energy, the lights, AC, and range were connected.

Figure 1 shows how the 2D-BPSO can help to obtain a faster return of investment, by using it on a green smart home. The algorithm had two possible effects that caused the smaller slope for the green smart home line: 1) load was effectively balanced according to the real-time demand of the home, and 2) a surplus in source panels, and windmills was detected, such that the installation costs may be lowered. For the case of green homes, it is possible to detect excess of materials, but the performance of the 2D-BPSO is almost static (i.e. it is run only when drastic load changes occurr).

Figure 2 presents how 2D-BPSO affects the kilowatts per hour rate. The traditional house only depends on the power grid and achieves the total kilowatts. The 2D-BPSO can help to reduce the dependence on the power grid amost by a half, in the case of the green home. Furthermore, with the smart home, the efficiency of the system can be further improved, by adding

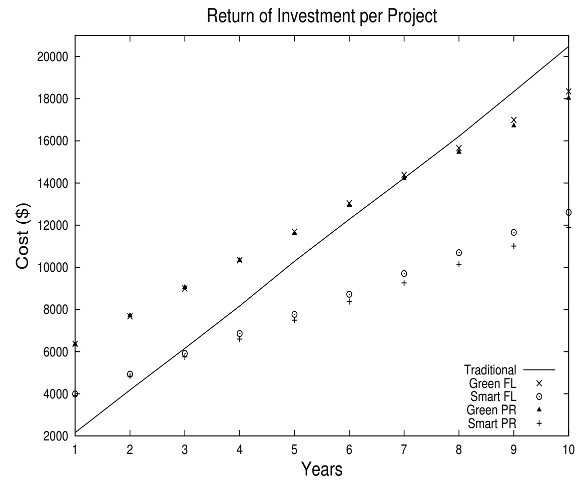

Fig. 1. **Return of Investment for Projects.** The traditional house is at the peak of cost, followed by the house powered by green energy. A smart home powered by green energy and managed by the 2D-BPSO is, in a long term basis, the most economical type of house. This occurrs when load balancing is optimized, and with some cognition the energy consumption can be adapted to the current context. Hence, the 2D-BPSO can adapt the load accordingly. This accomplishes two things: 1) lower installation costs, and 2) lower energy consumption per appliance.

cognition and adaptiveness with the 2D-BPSO, reducing the grid dependency by nearly 70%.

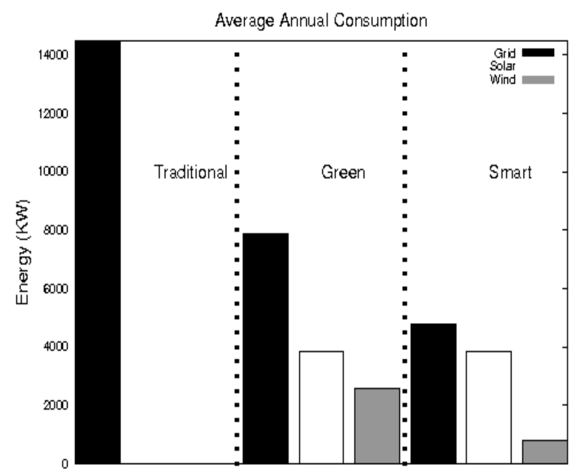

Fig. 2. **Consumption for Different House Types.** Load distribution per energy source per each type of home. For the green home, the 2D-BPSO cuts the dependency by distributing the load, across the green sources. On the smart home, the 2D-BPSO can be combined with the cognition, to obtain higher profit. Thus, conserving energy resources.

The performance of the 2D-BPSO was compputed on the average global best value. Figures 3 and 4 depict the results acquired. On average, a maximum number of 130 function evaluations (i.e. 5 generations) were needed in order to find the Pareto optimal configuration. Therefore, a faster solution is produced, for the problem, similar to the Multidimensional Knapsack. Moreover, the average curve demonstrates room for improvement, because many particles obtained low fitness.

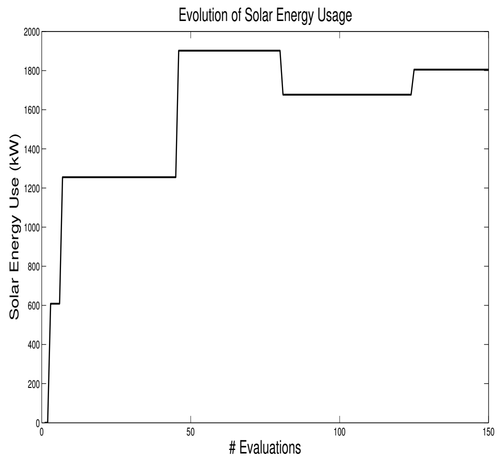

Fig. 3. **2D-BPSO Performance (Solar Energy).** The average best fitness is shown. The optimum value for the solar energy was obtained around 50 function evaluations. However, the system dropped the fitness at then around 130 function evaluations, because a Pareto point existed. In terms of generations the values were obtained in 2 (best) and 5 (Pareto best) generations.

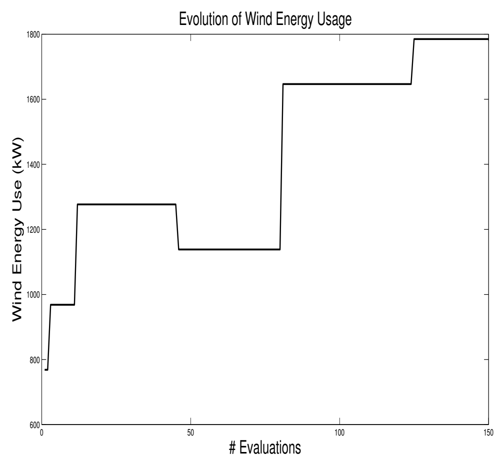

Fig. 4. **2D-BPSO Performance (Wind Energy).** The average best fitness is shown. The optimum value for the wind energy was obtained around 130 function evaluations (5 generations). This in time, is also the Pareto optimum of the system as seen in the solar energy performance.

VI. Conclusion

The 2D-BPSO demostrates how the basic PSO evolutionary technque is practical for the load-balancing application, for a green smart home. The original BPSO was successfully adapted, in order to find an multi-objective optimal configuration for the solar and wind systems. This new adapted BPSO algorithm resulted to be cheap, and was able to find surplus of resources.

The optimal configuration of loads and their respective power sources (i.e. tradition and green energy sources), reflected a significant cost reduction on initial payment, and longterm usage of the system. Thus the green smart home demonstrated to be a good investion over time, with faster return of investment. This leaves open research for further study of how evolutionary computation can provide other features in green smart homes, such as fault tolerance circuits.

The next step in our research, involves adding fault tolerance methods to the 2D-BPSO, such that the algorithm can adapt in the event of a natural disaster. Furthermore, we shall consider employing the 2D-BPSO other energy management applications, such as industrial energy distribution.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," vol. 4, nov. 1995, pp. 1942–1948.
[2] M. Tariquzzaman, J. Y. Kim, and S. Y. Na, "Pso based optimized reliability for robust multimodal speaker identification," 2010.
[3] J. L. C. L. W. W. Wei Yang, Xiaoqing Yu, "Audio feature optimization based on the pso and attribute importance," 2010.
[4] N. Ghosh, S. Nanda, and S. K. Ghosh, "A quantitative approach towards detection of an optimal attack path in a wireless network using modified pso technique," 2009.
[5] US Energy Information Administration, "Independent statistics and analysis," http://www.eia.doe.gov/aer/consump.html.
[6] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm," vol. 5, oct. 1997, pp. 4104–4108 vol.5.
[7] L. Xue1, J. Yin, Z. Ji, and L. Jiang, "A particle swarm optimization for hidden markov model training," 2006.
[8] R. M. Ramadan and R. F. Abdel-Kader, "Face recognition using particle swarm optimization-based selected features," 2009.
[9] H. Gao, W. Xu, J. Sun, and Y. Tang, "Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm," 2010.
[10] T. K. Das and G. K. Venayagamoorthy, "Optimal design of power system stabilizers using a small population based pso," 2006.
[11] R. Brown, "Impact of smart grid on distribution system design," jul. 2008, pp. 1–4.
[12] C.-Y. Chen, Y.-P. Tsoul, S.-C. Liao, and C.-T. Lin, "Implementing the design of smart home and achieving energy conservation," jun. 2009, pp. 273–276.
[13] G. Bellarmine, "Load management techniques," 2000, pp. 139–145.
[14] T. Dung, "The first solar villa in vietnam," in *Photovoltaic Energy Conversion, 2003. Proceedings of 3rd World Conference on*, vol. 3, 2003, pp. 2318–2321 Vol.3.
[15] T. K. Barlas and G. A. M. van Kuik, "State of the art and prospectives of smart rotor control for wind turbines," *Journal of Physics: Conference Series*, vol. 75, no. 1, p. 012080, 2007. [Online]. Available: http://stacks.iop.org/1742-6596/75/i=1/a=012080
[16] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation improvements in home automation strategies for designing apparatus for efficient smart home," *Power Systems Conference and Exposition, IEEE Proceedings on*, mar. 2009.
[17] M. Jahn, M. Jentsch, C. Prause, F. Pramudianto, A. Al-Akkad, and R. Reiners, "The energy aware smart home," may. 2010, pp. 1–8.
[18] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *Security Privacy, IEEE*, vol. 7, no. 3, pp. 75–77, may. 2009.
[19] K. Balasubramanian and A. Cellatoglu, "Improvements in home automation strategies for designing apparatus for efficient smart home," *Consumer Electronics, IEEE Transactions on*, vol. 54, no. 4, pp. 1681–1687, nov. 2008.
[20] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990.
[21] S. Kalogirou, *Solar Energy Engineering: Processes and Systems*, 2009.