Computational Mechanics
https://doi.org/10.1007/s00466-022-02212-8

ORIGINAL PAPER

An adaptive digital framework for energy management of complex multi-device systems

T. I. Zohdi[1]

Received: 16 April 2022 / Accepted: 29 June 2022

Abstract

Energy Management Systems (EMS) refer to frameworks that control the energy generation, transmission and storage for multiple devices which are coupled together. These can range from nationwide grids, utility-scale systems, microgrids, data-centers to electric vehicles and can consist of renewable energy sources, fossil-fuel energy, transmission lines, batteries, generators, ultracapacitors and transformers, to name a few. The goals of such systems are typically to balance the load, guarantee the power supply for each device, maximize overall efficiency and to minimize overall losses. Remarkable increases in desktop computing have opened up the possibility for researchers and practitioners to construct and tailor simulation paradigms for their own specific system's needs. Accordingly, the objective of this work is to develop a flexible and rapidly computable framework that researchers can easily alter and manipulate for their specific system. The approach taken in this work is to study a model problem, consisting of an energy supplier and a large number of strongly coupled devices with specific needs. The framework computes an energy balance for each device in the system and ascertains what the energy supplier must deliver or extract from the device to allow it to meet a specific target state while accounting for transmission losses. A digital-twin is created of such a system that is capable of running at extremely high speeds and which is coupled to a genetic-based machine-learning algorithm in order to optimize the operation of the supplier. Numerical examples are provided to illustrate the approach.

Keywords Energy management systems · Digital-twin · Machine-learning

1 Introduction

Energy Management Systems (EMS) refer to frameworks that control the energy generation, transmission and storage for multiple devices which are coupled together. These can range from nationwide grids, utility-scale systems, microgrids, data-centers and electric vehicles (Fig. 1) and can consist of renewable energy sources, fossil-fuel energy, transmission lines, batteries, generators, ultracapacitors and transformers. The goals of such systems are typically to balance the load, guarantee the power supply for each device, maximize overall efficiency and to minimize overall losses.

✉ T. I. Zohdi
zohdi@berkeley.edu

1 Department of Mechanical Engineering, University of California, Berkeley, CA 94720-1740, USA

1.1 Motivation

One motivation for the development of next generation Energy Management Systems that are flexible and easy to tailor to specific needs is the exponential rise of data-centers. Massive increases in internet users worldwide has led to significant demand for data-center services, and subsequent energy use. Following a review of Zohdi [59], we define data-centers as locations dedicated to housing computer systems comprised of data handling units, telecommunications, high-performance computing devices and associated equipment. Between 2010 and 2018, the global quantity of data traversing the internet increased more than ten-fold, while global data-center storage capacity increased by a factor of 25 in parallel (Masanet et al. [44]). At the largest industrial-scale, the energy usage of such systems is huge, requiring large-scale cooling and air conditioning. Such systems started in the 1940s with the advent of the first computers and have grown with the rise of industrial-scale computation at military installations, research labs, banks, to name a few. The

Published online: 30 July 2022

Fig. 1 Left: Modules in a renewable grid (Public domain: https://pixnio.com). Right: City scale management (Public domain: https://pixabay.com)

heat produced by such systems is immense, thus warranting sophisticated cooling systems. While the analysis of the energy trends are hotly debated, one point of agreement is that the volume of data-centers is consistently increasing, year by year. The reader is referred to [4,5,7,11,13,17,23,27–33,37,42,43,46,48,49,51,53,54,57] for a wide swath of the literature on this topic. All data indicate that the costs of such systems is huge and growing rapidly. The basic trends on energy consumption by data-centers can be found in the extensive report of Shehabi et al [50]. Therein, the authors have made accurate estimates of data-center energy consumption from 2000-2016, relying on previous studies, historical data and forecasted consumption. That report states that in 2014, data-centers in the U.S. consumed an estimated 70 billion kWh, representing about 1.8 % of total U.S. electricity consumption. Their analysis also indicates data-center electricity consumption increased by about 4 % from 2010-2014, a large shift from the 24 % percent increase estimated from 2005-2010 and a nearly 90 % increase estimated from 2000-2005. The trends of approximately 1 % increase each year have been consistent over the last decade. In 2017, US based data-centers alone used up more than 90 billion kilowatt-hours of electricity and consumed around 205 terawatt-hours (TWh) in 2018, or approximately 1 % of global electricity use (Masanet et al. [44]), and continues to grow, even through the era of pandemic. The massive growth in data-centers has led to increased interest and regulations for management of waste heat and its utilization. In Zohdi [59], an in-depth analysis was conducted to optimize a data-center's power management. That work sought to develop a combined digital-twin (a digital "replica") and machine-learning framework to optimize such systems by controlling both the surrounding ventilation and base foundation cooling of the data processors in the system. That framework ascertained optimal cooling strategies to deliver a target temperature in the system using a minimum amount of energy. A model problem was constructed for a data-center, where the design variables are the flow rates and air-cooling at multiple ventilation ports and ground-level, conduction-based, base-cooling of processors. A thermo-fluid model, based on the Navier-Stokes equations and the first law of thermodynamics, for the data-center was constructed and a rapid, voxel-based, iterative solution method was developed. This was then combined with a genetic-based machine-learning algorithm to develop a digital-twin of the system that could run in real-time or faster than the actual physical system, making it suitable as either a design tool or an adaptive controller. The present work extends such an analysis to entire networks of multiple data-centers or other energy-driven units referred to as "devices" (Fig. 2).

1.2 Restrictions

Much of the present work is motivated by recent governmental restrictions on energy waste for such systems. This has led to interest in developing EMS that power data-centers efficiently. A key aspect is the modularity of such frameworks and the ability to rapidly plan and configure multi-device systems. The power needed for a data-center can range from a few kilowatts for a small set of units to megawatts for a large-scale operation. There are a variety of performance metrics used, such as the power usage effectiveness ratio (PUE), which is the ratio of the total operational power used by the data-center divided by the power used by purely data processing equipment. It is an indicator of the overhead power consumption, such as cooling and lighting. Typical data-centers have a PUE of approximately 2, while the state of the art systems have a PUE $\approx$ 1.2. In 2014, the California Code of Regulations mandated energy efficiency regulations, in particular on airflow. In 2015, the United States enacted the Energy Efficiency Improvement Act, which requires efficient operation of federal facilities, including data-centers. Worldwide, in particular throughout the EU, there have been a series of similar legislation. However, even if one puts legislation aside, the sheer cost of running a data-center approaches the

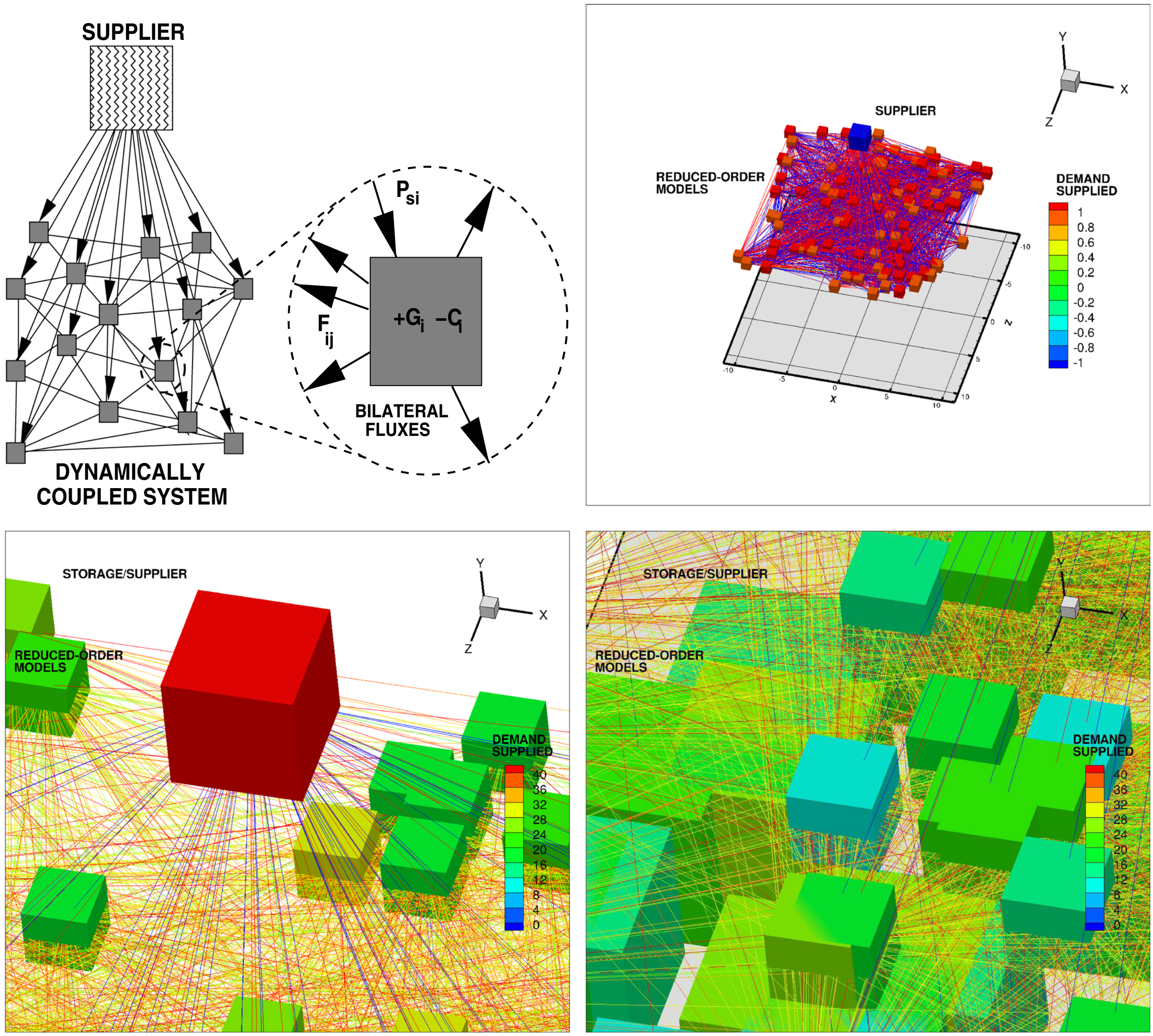

Fig. 2 Network model and an energy balance and a close-up for a device

construction costs.[1] This is one example of many modern, ultra-complex, energy management system problems that are now facing our society. *It has now become critical to develop Energy Management Systems that employ simulation based models to rapidly guide operations.*

1.3 Energy Management Systems

Remarkable increases in desktop computing have opened up the possibility for researchers and practitioners to construct and tailor simulation paradigms for their own specific system's needs. Accordingly, the objective of this work is to develop a flexible and rapidly computable framework that researchers can easily alter and manipulate for their specific system. The approach taken in this work is to study a model problem, consisting of an energy supplier and a large number of strongly coupled devices with specific needs. *The model problem (Fig. 2) has as the minimization of the overall energy that the supplier must deliver to the system as its main objective.* The key points in the model problem are:

- The design variables are the positions of the devices in the system.
- The devices generate and consume energy locally and may extract or return power to supplier.
- The supplier must make up deficits to the devices.

[1] For very energy intensive data-centers, electricity can account for over 10 % of the cost of ownership.

The main assumptions are:

- There is a specified, fixed, nonnegotiable connectivity between system devices.
- There is a unique specified energy level needed at any given time for each device.
- There are losses in power transmission per length and load magnitude.

The framework computes an energy balance for each device in the system and ascertains what the energy supplier must deliver or extract from the device to allow it to meet a specific target state while accounting for transmission losses. A digital-twin is created of such a system that is capable of running at extremely high speeds and which is coupled to a genetic-based machine-learning algorithm in order to optimize the operation of the supplier. Numerical examples are provided to illustrate the approach.

2 Model problem: a coupled multi-device system

There are two main entities in the system:

- The supplier, which delivers energy to and from each device in the system.
- The devices, whose energy states are denoted D_i, $i = 1, 2, \ldots, N$, which can consume and generate energy, as well as transfer energy to and from other units and the system supplier.

For each device in the system, $i = 1, 2, \ldots N$ (Fig. 2), consider a control volume tracking its energy state. Each device's energy state (Joules) at time $t + \Delta t$, denoted $D_i(t + \Delta t)$, is equal to the device state at time t, denoted $D_i(t)$, plus the total input/output of external energy from time t to time $t + \Delta t$, denoted $\Delta E_i^{tot}(t \rightarrow t + \Delta t)$

$$D_i(t + \Delta t) = D_i(t) + \Delta E_i^{tot}(t \rightarrow t + \Delta t). \tag{2.1}$$

By subtracting $D_i(t)$ from both sides and dividing by Δt yields

$$\frac{D_i(t + \Delta t) - D_i(t)}{\Delta t} = \frac{\Delta E_i^{tot}}{\Delta t}. \tag{2.2}$$

Taking the limit as $\Delta t \rightarrow 0$ yields the "power equation" for each device:

$$\frac{dD_i}{dt} = \frac{dE_i^{tot}}{dt} = P_i^{tot} = Total\ Power. \tag{2.3}$$

The breakdown of the components that comprise P_i^{tot}, $i = 1, 2, \ldots, N$ is discussed next.

2.1 Components of power

For each device, $i = 1, \ldots, N$ (Fig. 2), we consider the following breakdown of the total power:

$$\frac{dD_i}{dt} = P_i^{tot} = G_i - C_i - \Sigma_{j=1}^{N} F_{i \leftrightarrow j}\alpha_{ij} + P_{s \leftrightarrow i}, \tag{2.4}$$

leading to

$$D_i(t + \Delta t) = D_i(t) + \Delta t(G_i - C_i - \Sigma_{j=1}^{N} F_{i \leftrightarrow j}\alpha_{ij} + P_{s \leftrightarrow i}), \tag{2.5}$$

where

- G_i is the power generated,
- C_i is the power consumed,
- $F_{i \leftrightarrow j}$ is the net flux between devices i and j, with the following unilateral conditions:
- If $F_{i \leftrightarrow j} > 0$, then power is sent to device j and $\alpha_{ij} = 1$, with a loss at j
- If $F_{i \leftrightarrow j} \leq 0$, then power is sent from device j and with a loss at receiver i, $\alpha_{ij} = e^{-\kappa d_{ij}}$, where κ is the power loss per unit length and d_{ij} is the distance between device i and device j.

Regardless of whether the device i sends or receives from j, the loss between them is

$$\mathcal{L}_{ij} = ||F_{i \leftrightarrow j}||(1 - e^{-\kappa d_{ij}}). \tag{2.6}$$

The term $P_{s \leftrightarrow i}$ is determined from the needs of the device, dictated by γ_{si}

$$\gamma_{si} \stackrel{\text{def}}{=} \frac{dD_i}{dt} - (G_i - C_i - \Sigma_{j=1}^{N} F_{i \leftrightarrow j}\alpha_{ij}), \tag{2.7}$$

where

- If $\gamma_{si} > 0$, then power needs to be sent to device i from the supplier,
- If $\gamma_{si} \leq 0$, then power will be sent from device i to the supplier.

2.2 Targeted energy supply

If we set the target value at any time to be $D_i^*(t + \Delta t)$, then

- If there is an energy deficit to be supplied to unit i from S, then we may write that the externally needed is (factoring in the losses)

$$P^{*}_{s\leftrightarrow i} \stackrel{\text{def}}{=} P^{*,+}_{s\leftrightarrow i} e^{-\kappa d_{si}} = \frac{D^{*}_{i}(t+\Delta t) - D_i(t)}{\Delta t} - (G_i - C_i - \Sigma_{j=1}^{N} F_{i\leftrightarrow j}\alpha_{ij}), \tag{2.8}$$

with connectivity losses of

$$\mathcal{L}_{ij} = ||P^{*,+}_{s\leftrightarrow i}||(1 - e^{-\kappa d_{si}}). \tag{2.9}$$

- If there is excess energy is to be returned from unit i to S, then we may write

$$P^{*}_{s\leftrightarrow i} \stackrel{\text{def}}{=} P^{*,-}_{s\leftrightarrow i} = \frac{D^{*}_{i}(t+\Delta t) - D_i(t)}{\Delta t} - (G_i - C_i - \Sigma_{j=1}^{N} F_{i\leftrightarrow j}\alpha_{ij}) \tag{2.10}$$

with connectivity losses of

$$\mathcal{L}_{ij} = ||P^{*,-}_{s\leftrightarrow i}||(1 - e^{-\kappa d_{si}}), \tag{2.11}$$

where $P^{*,-}_{s\leftrightarrow i}$ leaves unit i, which then arrives at S as $P^{*,-}_{s\leftrightarrow i} e^{-\kappa d_{si}}$.

This is computed for each device at every time step and changes continuously, based on the individual needs and its energy exchange with the surroundings.

3 Genetic-based machine-learning optimization

The objective now is to minimize the energy needed to be delivered by the supplier to the devices in the system:

$$\Pi^{o}(\Lambda_1, \ldots, \Lambda_N) \stackrel{\text{def}}{=} \Sigma_{i=1}^{N} \int_0^T P^{*}_{s\leftrightarrow i}(\boldsymbol{\Lambda}, t)\, dt, \tag{3.1}$$

where we consider a design vector $\boldsymbol{\Lambda} \stackrel{\text{def}}{=} \{\Lambda_1, \Lambda_2, \Lambda_3, \ldots, \Lambda_N\}$, which represents the positions of the devices and where the cost is the power integrated over the course of time. In order to avoid clustering of the devices, we enforce a spacing penalty for a configuration consisting of a proximity penalization function summing all individual pairwise device separation distances

$$\zeta \stackrel{\text{def}}{=} \frac{\sqrt{\Sigma_{i=1}^{N}\Sigma_{j=1}^{N}\phi_{ij}(||\boldsymbol{r}_i - \boldsymbol{r}_j|| - d^*)^2}}{N d^*}, \tag{3.2}$$

where $\boldsymbol{r}_i$ is the position vector of device i, $\boldsymbol{r}_j$ is the position vector of device j, d^* is a critical separation distance and where:

- If $||\boldsymbol{r}_i - \boldsymbol{r}_j|| \le d^*$ then $\phi_{ij} = 1$,
- If $||\boldsymbol{r}_i - \boldsymbol{r}_j|| > d^*$ then $\phi_{ij} = 0$.

The function is then included to penalize clustering via

$$\Pi(\Lambda_1, \ldots, \Lambda_N) = (1 + w\zeta)\Pi^{o}(\Lambda_1, \ldots, \Lambda_N), \tag{3.3}$$

where $w \ge 0$ is a penalty weight. The rapid rate at which these simulations can be completed allows the exploration of inverse problems seeking to determine what parameter combinations can deliver a desired result (Fig. 3). In order to cast the objective mathematically, we set the problem up as a machine-learning algorithm (MLA), specifically a genetic algorithm (GA) variant, which is well-suited for nonconvex optimization. Following Zohdi [59–65], we formulate the objective as a cost function minimization problem that seeks system parameters that match a desired response, in this case a minimum of $\Pi(\Lambda_1, \ldots \Lambda_N)$. We systematically minimize Eq. (3.1), $min_{\Lambda}\Pi$, by varying the design parameters: $\boldsymbol{\Lambda} \stackrel{\text{def}}{=} \{\Lambda_1, \Lambda_2, \Lambda_3, \ldots, \Lambda_N\}$. The system parameter search is conducted within the constrained ranges of $\Lambda_1^{(-)} \le \Lambda_1 \le \Lambda_1^{(+)}$, $\Lambda_2^{(-)} \le \Lambda_2 \le \Lambda_2^{(+)}$, $\Lambda_3^{(-)} \le \Lambda_3 \le \Lambda_3^{(+)}$, etc. These upper and lower limits are dictated by what is physically feasible.

3.1 Machine-learning algorithm (MLA)

Cost functions such as Π are nonconvex in design parameter space and often nonsmooth. Their minimization is usually difficult with direct application of gradient-based methods. This motivates nonderivative search methods, for example those found in machine-learning algorithms (MLAs). One of the most basic subsets of MLAs are so-called Genetic Algorithms (GAs). For a review of GAs, see the pioneering work of John Holland ([24,25]]), as well as Goldberg [20], Davis [10], Onwubiko [45] and Goldberg and Deb [21]. A description of the algorithm will be described next, following Zohdi [59–65].

3.2 Algorithmic structure

The MLA/GA approach is extremely well-suited for nonconvex, nonsmooth, multicomponent, multistage systems and, broadly speaking, involves the following essential concepts (Fig. 3):

1. **POPULATION GENERATION:** Generate a parameter population of genetic strings: $\boldsymbol{\Lambda}^i$
2. **PERFORMANCE EVALUATION:** Compute performance of each genetic string: $\Pi(\boldsymbol{\Lambda}^i)$
3. **RANK STRINGS:** Rank them $\boldsymbol{\Lambda}^i$, $i = 1, \ldots, S$ from best to worst
4. **MATING PROCESS:** Mate pairs/produce offspring

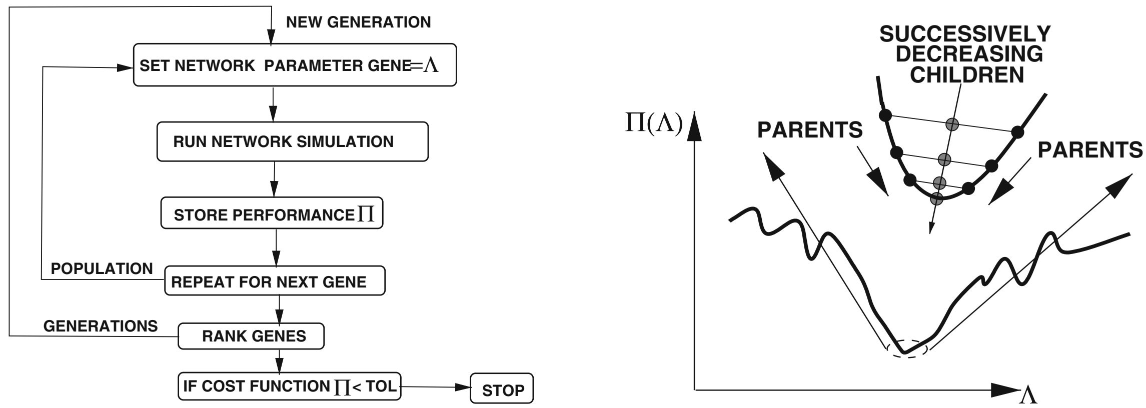

Fig. 3 The basic action of a machine-learning/genetic algorithm (Zohdi [59–65])

5. **GENE ELIMINATION:** Eliminate poorly performing genetic strings
6. **POPULATION REGENERATION:** Repeat process with updated gene pool and new *random* genetic strings
7. **SOLUTION POST-PROCESSING:** Employ gradient-based methods afterwards in local "valleys"-*if smooth enough*

3.3 Specifics

Following Zohdi [59–65]. the algorithm is as follows:

- **STEP 1:** Randomly generate a population of S starting genetic strings, $\mathbf{\Lambda}^i$, $(i = 1, 2, 3, \ldots, S)$:

$$\mathbf{\Lambda}^i \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \Lambda^i_1 \\ \Lambda^i_2 \\ \Lambda^i_3 \\ \cdots \\ \Lambda^i_N \end{array} \right\} \tag{3.4}$$

- **STEP 2:** Compute fitness of each string $\Pi(\mathbf{\Lambda}^i)$, (i=1, …, S)
- **STEP 3:** Rank genetic strings: $\mathbf{\Lambda}^i$, (i=1, …, S) from best to worst
- **STEP 4:** Mate nearest pairs and produce two offspring, (i=1, …, S):

$$\lambda^i \stackrel{\text{def}}{=} \mathbf{\Phi} \circ \mathbf{\Lambda}^i + (\mathbf{1} - \mathbf{\Phi}) \circ \mathbf{\Lambda}^{\mathbf{i+1}} \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \phi_1 \Lambda^i_1 \\ \phi_2 \Lambda^i_2 \\ \phi_3 \Lambda^i_3 \\ \cdots \\ \phi_N \Lambda^i_N \end{array} \right\} + \left\{ \begin{array}{c} (1-\phi_1)\Lambda^{i+1}_1 \\ (1-\phi_2)\Lambda^{i+1}_2 \\ (1-\phi_3)\Lambda^{i+1}_3 \\ \cdots \\ (1-\phi_N)\Lambda^{i+1}_N \end{array} \right\} \tag{3.5}$$

and

$$\lambda^{i+1} \stackrel{\text{def}}{=} \mathbf{\Psi} \circ \mathbf{\Lambda}^i + (\mathbf{1} - \mathbf{\Psi}) \circ \mathbf{\Lambda}^{\mathbf{i+1}} \stackrel{\text{def}}{=} \left\{ \begin{array}{c} \psi_1 \Lambda^i_1 \\ \psi_2 \Lambda^i_2 \\ \psi_3 \Lambda^i_3 \\ \cdots \\ \psi_N \Lambda^i_N \end{array} \right\} + \left\{ \begin{array}{c} (1-\psi_1)\Lambda^{i+1}_1 \\ (1-\psi_2)\Lambda^{i+1}_2 \\ (1-\psi_3)\Lambda^{i+1}_3 \\ \cdots \\ (1-\psi_N)\Lambda^{i+1}_N \end{array} \right\} \tag{3.6}$$

where for this operation, the ϕ_i and ψ_i are random numbers, such that $0 \leq \phi_i \leq 1$, $0 \leq \psi_i \leq 1$, which are different for each component of each genetic string

- **STEP 5:** Eliminate the bottom M strings and keep top K parents and their K offspring (K offspring+K parents+M=S)
- **STEP 6:** Repeat STEPS 1-5 with top gene pool (K offspring and K parents), plus M new, randomly generated, strings
- **REFOCUS OPTION:** One can refocus search around best performing parameter set every few generations, thus concentrating the computational effort around the most promising (optimal) areas of design space.

Remark 1 If one selects the mating parameters $\phi's$ and $\psi's$ to be greater than one and/or less than zero, one can induce "mutations", i.e. characteristics that neither parent possesses. However, this is somewhat redundant with introduction of

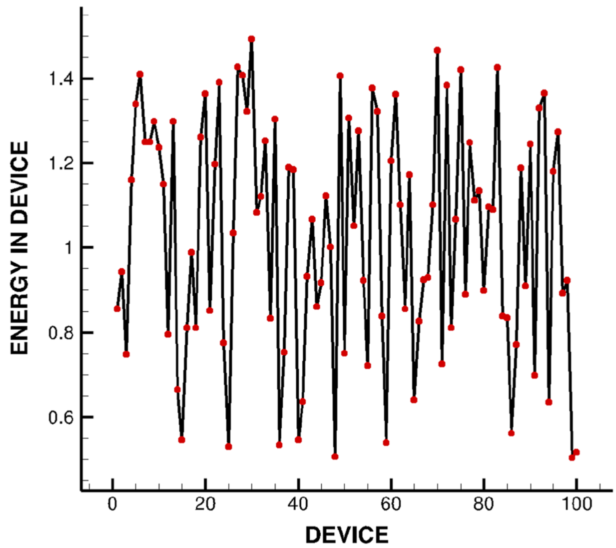

Fig. 4 An example of one-hundred specified device energies that must be met locally by the supplier. In this test case, they fluctuated between 0.5 and 1.5

new random members of the population in the current algorithm. If one does not retain the parents in the algorithm above, it is possible that inferior performing offspring may replace superior parents. Thus, top parents should be kept for the next generation. Retained parents do not need to be reevaluated, making the algorithm less computationally expensive. In the absence of refocussing, numerous studies of the author (Zohdi [59–65]) have shown that the advantages of parent retention outweighs inbreeding, for sufficiently large population sizes. Finally, we observe that this algorithm is easy to parallelize.

Remark 2 After application of such a global search algorithm, one can apply a gradient-based method around the best performing parameter set, if the objective function is sufficiently smooth in that region of the parameter space. In other words, if one has located a convex portion of the parameter space with a global genetic search, one can employ gradient-based procedures locally to minimize the objective function further, since they are generally much more efficient for convex optimization of smooth functions. An exhaustive review of these methods can be found in the texts of Luenberger [35] and Gill, Murray and Wright [18]. *However, refocussing usually makes this extra step unnecessary, since the search eventually concentrates the computational effort locally around the best parameter set beforehand.*

3.4 Algorithmic settings

In the upcoming example, search parameter ranges were used $\Lambda_i^- = -10 \leq \Lambda_i \leq \Lambda_i^+ = 10$ for 200 planar coordinate variables, $\mathbf{\Lambda}_i = (x_i, 0, z_i), i = 1, \ldots, 100$ (200 variables). Specifically, we used the following MLA settings:

- Number of design variables: 200,
- Population size per generation: 24,
- Number of parents to keep in each generation: 6,
- Number of children created in each generation: 6,
- Number of completely new genes created in each generation: 12,
- Number of generations for re-adaptation around a new search interval: 20 and
- Number of generations: 4000.

3.5 Parameter search ranges and results

The system parameter setting for were set as follows:

- Transmission losses: the loss per unit length: $L = 0.05$,
- Connectivity: for any given device, i, 50 % of the other system units were connected to it,
- Target desired device energy values (Fig. 4): fixed over the time interval with a random variation of $D_i = D_o(1 + A \times RAND)$, where $-1 \leq RAND \leq 1$, $D_o = 1$, $A = 0.01$,
- Inter-device fluxes: during the time interval, the fluxes were were set to vary randomly between: $-100 \leq F_{i \leftrightarrow j} \leq 100$,
- Energy generated: during the time interval (Fig. 4), the energies generated locally were set to vary randomly between: $-100 \leq G_i \leq 100$,
- Energy consumed: during the time interval, the energies consumed locally were set to vary randomly between: $-100 \leq C_i \leq 100$,
- The following search parameter ranges were used $\Lambda_i^- = (-10, 0, -10)m \leq \Lambda_i \leq \Lambda_i^+ = (10, 0, 10)m$ for 200 planar coordinate variables, $\mathbf{\Lambda}_i = (x_i, 0, z_i), i = 1, \ldots, 100$ (200 variables).

Figure 5 illustrates the results for the cost function for the best performing gene (*red*) as a function of successive generations, as well as the average performance cost function of the entire population of genes (*green*). Starting at the top left and moving to the right, we allowed the MLA/GA to readapt every 20 generations. Often, this action is more efficient than allowing the algorithm not to readapt, since it probes around the current optimum for better local alternatives. The total cost function was initially $\Pi^{total} \approx 10.7$ Megawatts and was reduced to $\Pi^{tot} \approx 4.7$ Megawatts, a reduction of $\frac{10.7-4.7}{10.7} \rightarrow 56\%$. The entire 4000 generation simulation, with 24 genes per evaluation (96,000 total designs) took a few minutes on a laptop, *making it ideal as a design tool.* We note that, for a given set of parameters, a complete simulation takes a fraction of a second, thus hundreds of thousands

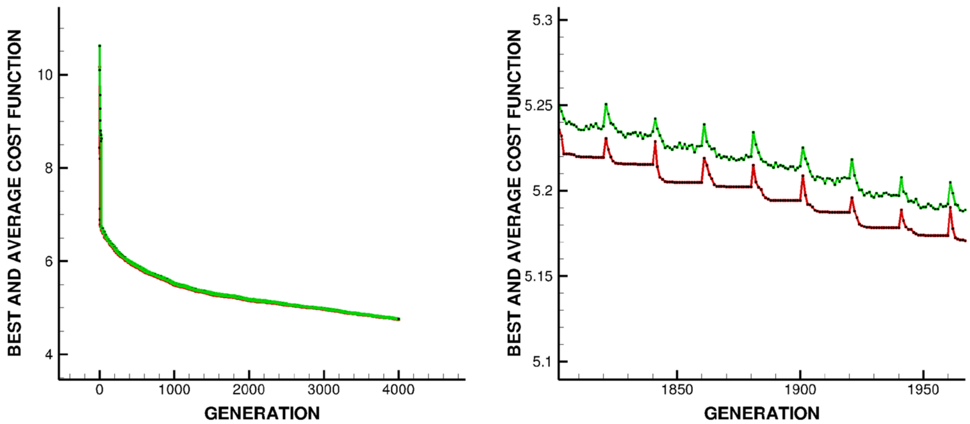

Fig. 5 Successive generations and a zoom: The following search parameter ranges were used $\Lambda_i^- = (-10, 0, -10) \leq \Lambda_i \leq \Lambda_i^+ = (10, 0, 10)$ for 200 planar coordinate variables, $\mathbf{\Lambda}_i = (x_i, 0, z_i), i = 1, \ldots, 100$ (200 variables). This figure illustrates the results for the cost function for the best performing gene (*red*) as a function of successive generations, as well as the average performance cost function of the entire population of genes (*green*). Starting at the top left and moving to the right, we allowed the MLA/GA to readapt every 20 generations. Often, this action is more efficient than allowing the algorithm not to readapt, since it probes around the current optimum for better local alternatives. The total cost function was initially $\Pi^{total} \approx 10.7$ Megawatts and was reduced to $\Pi^{tot} \approx 4.7$ Megawatts, a reduction of $\frac{10.7-4.7}{10.7} \rightarrow 56\%$

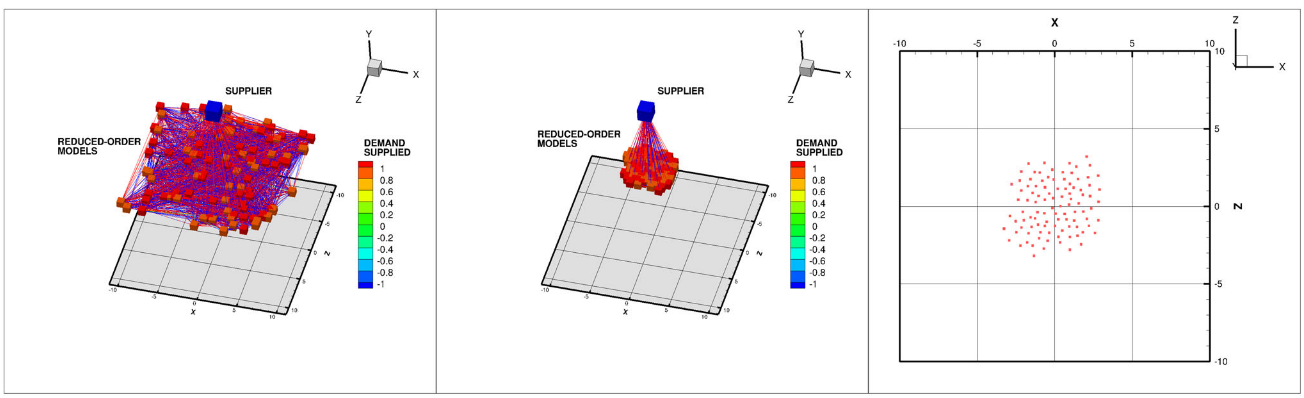

Fig. 6 The starting configuration (before MLA optimization) and the final (tightest) configuration associated with the global optimum

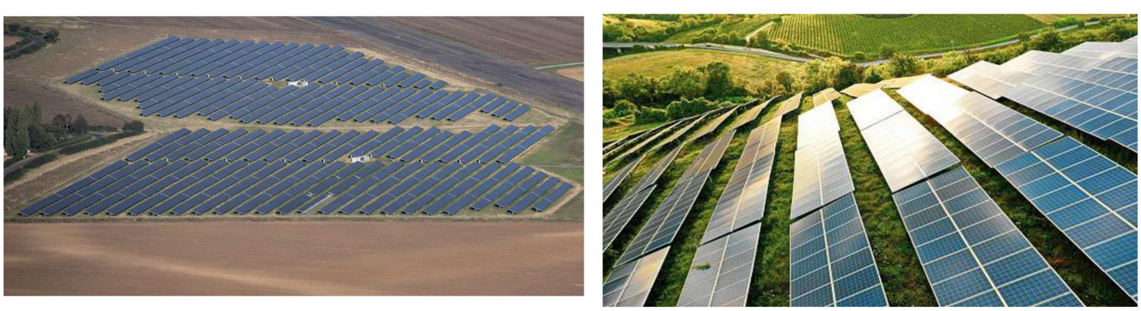

Fig. 7 Modules in agrophotovoltaic facilities (Public domain: https://pixnio.com and https://pixabay.com)

of parameter sets can be evaluated in an hour, *without even exploiting the inherent parallelism of the MLA/GA*. The speed at which the overall process can be completed makes it a suitable digital-twin of the system that can run in real-time or faster than the actual physical system, making it suitable as either a design tool or an adaptive controller. Figure 6 illustrates the final tightest configuration.

Remark 3 There are other machine-learning type paradigms that complement genetic-based approaches, such as Artificial Neural Networks (ANN). ANN have received huge attention in the scientific community over the last decade and are based on layered input-output type frameworks that are essentially adaptive nonlinear regressions of the form $\mathcal{O} = \mathcal{B}(\boldsymbol{I}, \boldsymbol{w})$, where $\mathcal{O}$ is a desired output and $\mathcal{B}$ is the ANN comprised of (1) **synapses**, which multiply inputs (I_i, $i = 1, 2, \ldots, M$) by weights (w_i, $i = 1, 2, \ldots, N$) that represent the input relevance to the desired output, (2) **neurons**, which aggregate outputs from all incoming synapses and apply activation functions to process the data and (3) **training**, which calibrates the weights to match a desired overall output.

4 Summary

In summary, the objective of this work was to develop a flexible and rapidly computable framework that researchers can easily adapt and manipulate for their specific system. The approach taken in this work was to study a model problem, consisting of an energy supplier and a large number of strongly coupled devices with specific energy needs. The framework computes an energy balance for each device in the system and ascertains what the energy supplier must deliver or extract from the device to allow it to meet a specific target state while accounting for transmission losses, as well as local energy consumption and generation. A digital-twin was created of such a system that is capable of running at extremely high speeds and which is coupled to a genetic-based machine-learning algorithm in order to optimize the operation of the supplier. Numerical examples were provided to illustrate the approach.

At the outset of the work, one motivation provided was the massive growth of data-centers and the corresponding need for precise energy management. In Zohdi [59], a more detailed system at the unit level was developed combining a digital-twin and machine-learning framework to optimize such systems by controlling both the ventilation system and the cooling of the base supports of the data processing units in the system. That framework ascertained the optimal cooling strategies needed to deliver a target temperature in the system using a minimum amount of energy. A model problem was studied, where the design variables were the flow rates and air-cooling of the multiple ventilation ports and ground-level conduction-based base-cooling of processors. A fast solution method, based on a CFD representation of the data-center was developed using a voxel-based discretization of the Navier-Stokes equations and the first law of thermodynamics, which was combined with a genetic-based machine-learning algorithm to develop a digital-twin of the system, which is suitable as either a design tool or a controller, *when coupled to an overall Energy Management System, such as the one presented in this work*. However, there are many more societal areas that strongly motivate simulation based Energy Management Systems. For example, on the large infrastructural scale, it is important to highlight that more heterogeneous energy systems are being introduced into the market, driven largely by renewable energy being blended into societal systems. For example, agrophotovoltaic (APV) systems (Fig. 7) attempt to co-develop the same area of land for both solar photovoltaic power and agriculture. APV systems were pioneered in the 1980s (Goetzberger and Zastrow [19]) and have steadily grown as photovoltaic systems have become more robust and inexpensive. We refer the reader to [1–3,6,8,9,12,14–16,19,22,26,34,36,38–41,47,52,55,56,58] for a broad survey of such systems.[2] Regardless of the exact type of blended system, there is a necessity to optimize these complex heterogeneous systems so that they operate smoothly. If configurations are properly optimized, for example in the context of APVs, the approach can yield the best of both worlds, yielding energy and abundant agriculture. For example, in Zohdi [65], the focus was on developing a digital-Twin framework to track and optimize the flow of optical power through complex APV facilities. The optical power flow was rapidly computed with a reduced order model of Maxwell's equations, based on a high-frequency decomposition of optical power into multiple rays, which were propagated forward in time to ascertain multiple reflections and absorption for various system configurations, varying multi-panel inclination, tracking, refractive indices, sizes, shapes and ground refractive properties. The method allowed for a solar installation to be tested from multiple source directions quickly and uses a genetic-based machine-learning algorithm to optimize the system. This is particularly useful for planning of complex next-generation solar farm systems involving bifacial (double-sided) panelling, which are capable of capturing ground albedo reflection, exemplified by APV systems.

The presented Energy Management System work augments the previous applications (such as data-centers and APV's), with the key goal being to develop an easy simulation tool that is computationally inexpensive and accessible to a wide range of researchers. A central component of this framework was the digital-twin paradigm of physical reality, i.e. a digital replica of a complex system that can then be

[2] APV systems can involve a variety of aspects, even utilizing pollinating insects, such as bees, to "solar grazing" systems.

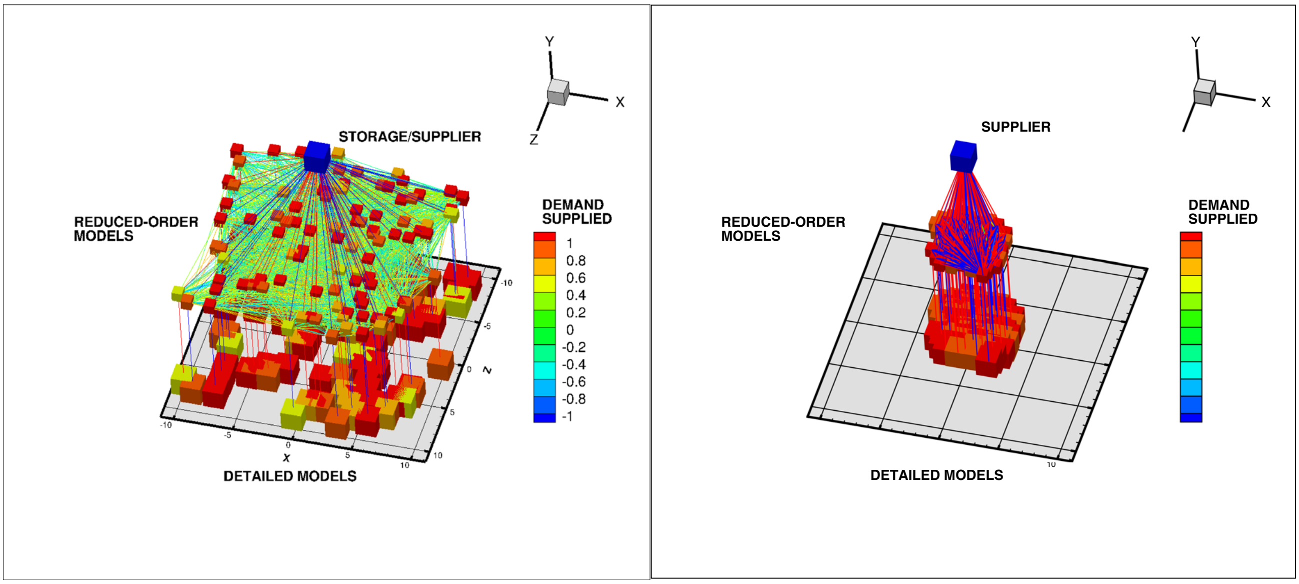

Fig. 8 With subgrid models underneath: The unoptimized (left) and the final (right, tightest) configuration associated with the global optimum

inexpensively and safely manipulated, improved and optimized in a virtual setting. The computationally designed system can then be deployed in the physical world afterwards, reducing the potential costs of experiments associated with bringing new technologies to the market. One important issue how to seamlessly couple complex subsystems to Energy Management Systems, for example using "subgrid" simulation, whereby a more complex simulation is activated either before the network model starts, in order to pre-calibrate ("train") the reduced-order model, or, during the simulation, at select time intervals, in order to recalibrate the system. A corresponding pictorial framework can be found in Fig. 8, where the subgrid models (such as data-centers or agrophotovoltaic subsystems) are shown underneath the reduced-order blocks. The hybrid use of digital-twins, genetic-based machine-learning, Artificial Neural Networks and multiscale simulation is currently under investigation by the author.

Acknowledgements This work has been partially supported by the UC Berkeley College of Engineering and the USDA AI Institute for Next Generation Food Systems (AIFS), USDA award number 2020-67021-32855.

References

1. Amaducci S, Yin X, Colauzzi M (2018) Agrivoltaic systems to optimise land use for electric energy production. Appl Energy 220:545–561. https://doi.org/10.1016/j.apenergy.2018.03.081
2. Armstrong A, Ostle NJ, Whitaker J (2016) Solar park microclimate and vegetation management effects on grassland carbon cycling. Environ Res Lett 11:74016. https://doi.org/10.1088/1748-9326/11/7/074016
3. Barron-Gafford GA, Minor RL, Allen NA, Cronin AD, Brooks AE, Pavao-Zuckerman MA (2016) The photovoltaic Heat Island effect: larger solar power plants increase local temperatures. Sci Rep 6(35070):1–7. https://doi.org/10.1038/srep35070
4. Belkhir L, Elmeligi A (2018) Assessing ICT Global Emissions Footprint: Trends To 2040 & Recommendations. J Clean Prod 177:448–463
5. Brown R, Masanet E, Nordman B, Tschudi W, Shehabi A, Stanley J, Koomey J, Sartor D, Chan P, Loper J, Capana S, Hedman B, Duff R, Haines E, Sass D, Fanara A (2007) Report to Congress on Server and data-center Energy Efficiency: Public Law 109–431. Lawrence Berkeley National Laboratory, Berkeley, California (**LBNL-363E**)
6. Castellano S (2014) Photovoltaic greenhouses: evaluation of shading effect and its influence on agricultural performances. J Agric Eng 45(4):168–175. https://doi.org/10.4081/jae.2014.433 (**ISSN 2239-6268**)
7. Cheung IH, Greenberg S, Mahdavi R, Brown R, Tschudi W (2014, August). Energy Efficiency in Small Server Rooms: Field Surveys and Findings. Proceedings the 2014 ACEEE Summer Study on Energy Efficiency in Buildings. LBNL- 6952E
8. Cossu M, Murgia L, Ledda L, Deligios PA, Sirigu A, Chessa F, Pazzona A (2014) Solar radiation distribution inside a greenhouse with south-oriented photovoltaic roofs and effects on crop productivity. Appl Energy 133:89–100. https://doi.org/10.1016/j.apenergy.2014.07.070
9. Cossu M, Yano A, Li Z, Onoe M, Nakamura H, Matsumoto T, Nakata J (2016) Advances on the semi-transparent modules based on micro solar cells: first integration in a greenhouse system. Appl Energy 162:1042–1051. https://doi.org/10.1016/j.apenergy.2015.11.002
10. Davis L (1991) Handbook of Genetic Algorithms. Thompson Computer Press, Van Nostrand Reinhold, New York
11. U.S. Department Of Energy (2020). Annual Energy Outlook 2020. https://Www.eia.gov/Outlooks/Aeo/
12. Dinesh H, Pearce JM (2016) The potential of agrivoltaic systems. Renew Sustain Energy Rev 54:299–308. https://doi.org/10.1016/j.rser.2015.10.024
13. Dudkowski D, Hasselmeyer P (2015) Energy-Efficient Networking in Modern data-centers. In: Samdanis K, Rost P, Maeder A, Meo M, Verikoukis C (eds) Green Communications: Principles, Concepts

and Practice. John Wiley & Sons, Hoboken, New Jersey (**ISBN 978-1-118-75926-4**)
14. Dupraz C, Marrou H, Talbot G, Dufour L, Nogier A, Ferard Y (2011) Combining solar photovoltaic panels and food crops for optimizing land use: towards new agrivoltaic schemes. Renew Energy 36:2725–2732. https://doi.org/10.1016/j.renene.2011.03.005
15. Elamri Y, Cheviron B, Mange A, Dejean C, Liron F, Belaud G (2017) Rain concentration and sheltering effect of solar panels on cultivated plots. Hydrol Earth Syst Sci Discuss 2017:1–37. https://doi.org/10.5194/hess-2017-418
16. Elamri Y, Cheviron B, Lopez J-M, Dejean C, Belaud G (2018) Water budget and crop modelling for agrivoltaic systems: application to irrigated lettuces. Agric Water Manag 208:440–453. https://doi.org/10.1016/j.agwat.2018.07.001
17. U.S. Energy Information Administration (2020). How Much Carbon Dioxide Is Produced Per Kilowatthour Of U.S. Electricity Generation https://Www.eia.gov/Tools/Faqs/Faq.phpId=74&T=11
18. Gill P, Murray W, Wright M (1995) Practical optimization. Academic Press, Cambridge, Massachusett
19. Goetzberger A, Zastrow A (1982) On the Coexistence of Solar-Energy Conversion and Plant Cultivation. Int J Solar Energy 1(1):55–69. https://doi.org/10.1080/01425918208909875 (**ISSN 0142-5919**)
20. Goldberg DE (1989) Genetic algorithms in search, optimization & machine learning. Addison-Wesley, Boston
21. Goldberg DE, Deb K (2000) Special issue on Genetic Algorithms. Comput Methods Appl Mech Eng 186(2–4):121–124
22. Gorjian S, Calise F, Karunesh Kant Md, Ahamed S, Copertaro B, Najafi G, Zhang X, Aghaei M, Shamshiri RR (2021) A review on opportunities for implementation of solar energy technologies in agricultural greenhouses. J Clean Prod 285:124807. https://doi.org/10.1016/j.jclepro.2020.124807 (**ISSN 0959–6526**)
23. Greenberg S, Mills E, Tschudi B, Rumsey P, Myatt B (2006) Best Practices for data-centers: Lessons Learned from Benchmarking 22 data-centers. Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings in Asilomar, CA. ACEEE, August 3, 76-87. http://eetd.lbl.gov/emills/PUBS/PDF/ACEEE-datacenters.pdf
24. Holland JH, Miller JH (1991) Artificial Adaptive Agents in Economic Theory (PDF). Am Econ Rev 81(2):365–71. Archived from the original (PDF) on October 27, 2005
25. Holland JH (1975) Adaptation in natural & artificial systems. University of Michigan Press, Ann Arbor, Mich
26. Homma M, Doi T, Yoshida Y (2016) A field experiment and the simulation on agrivoltaic-systems regarding to rice in a paddy field. J Jpn Soc Energy Resour 37:23–31. https://doi.org/10.24778/jjser.37.6_23
27. Horner N, Azevedo I (2016) Power usage effectiveness in data-centers: overloaded and underachieving. Electr J 29(4):61–69
28. Jones N (2018) How To Stop Data Centres From Gobbling Up The Worlds Electricity. Nature 561(7722):163–166. https://doi.org/10.1038/D41586-018-06610-Y
29. Koomey J (2008) Worldwide electricity used in data-centers. Environ Res Lett 3(034008). September 23. [http://stacks.iop.org/1748-9326/3/034008]
30. Koomey JG (2007) Estimating Total Power Consumption by Servers in the U.S. and the World. February 15. http://www.mediafire.com/file/exywo1hf6ionskw/AMDserverpowerusecomplete-final.pdf
31. Koomey JG (2011) Growth in data-center Electricity Use 2005 to 2010. Analytics Press, Oakland, California. http://www.analyticspress.com/datacenters.html
32. Koomey JG, Berard S, Sanchez M, Wong H (2011) Implications of historical trends in the electrical efficiency of computing. Ann Hist Comput, IEEE 33(3):46–54
33. Lanzisera S, Nordman B, Brown RE (2012) Data network equipment energy use and savings potential in buildings. Energ Effi 5(2):149–162
34. Liu W, Liu L, Guan G, Zhang F, Li M, Lv H, Yao P, Ingenhoff J (2018) A novel agricultural photovoltaic system based on solar spectrum separation. Sol Energy 162:84–94
35. Luenberger D (1974) Introduction to Linear & Nonlinear Programming. Addison-Wesley, Menlo Park
36. Majumdar D, Pasqualetti MJ (2018) Dual use of agricultural land: introducing agrivoltaics in Phoenix metropolitan statistical area, USA. Landsc Urban Plan 170:150–168. https://doi.org/10.1016/j.landurbplan.2017.10.011
37. Malone C, Belady C (September 2006) Metrics to characterize datacenter & IT equipment energy use. In: Proceedings of the Digital Power Forum, Richardson, TX
38. Malu PR, Sharma US, Pearce JM (2017) Agrivoltaic potential on grape farms in India. Sustain Energy Technol Assess 23:104–110. https://doi.org/10.1016/j.seta.2017.08.004
39. Marrou H, Dufour L, Wery J (2013) How does a shelter of solar panels influence water flows in a soil-crop system? Eur J Agron 50:38–51. https://doi.org/10.1016/j.eja.2013.05.004
40. Marrou H, Guilioni L, Dufour L, Dupraz C, Wery J (2013) Microclimate under agrivoltaic systems: is crop growth rate affected in the partial shade of solar panels? Agric For Meteorol 177:117–132. https://doi.org/10.1016/j.agrformet.2013.04.012
41. Marrou H, Wery J, Dufour L, Dupraz C (2013) Productivity and radiation use efficiency of lettuces grown in the partial shade of photovoltaic panels. Eur J Agron 44:54–66. https://doi.org/10.1016/j.eja.2012.08.003
42. Masanet E, Brown RE, Shehabi A, Koomey JG, Nordman B (2011) Estimating the Energy Use and Efficiency Potential of U.S. data-centers. Proc IEEE 99(8):1440–1453
43. Masanet E, Shehabi A, Ramakrishnan L, Liang J, Ma X, Walker B, Mantha P (2013) The Energy Efficiency Potential of Cloud-Based Software: A US Case Study. Lawrence Berkeley National Laboratory, Berkeley, California
44. Masanet E, Shehabi A, Lei N, Smith S, Koomey J (2020) Recalibrating Global data-center Energy-Use Estimates. Science 367(6481):984–986
45. Onwubiko C (2000) Introduction to engineering design optimization. Prentice Hall, Hoboken, New Jersey
46. Reviriego P, Maestro JA, Larrabeiti D (2010) Burst transmission for energy-efficient ethernet. Internet Comput, IEEE 14(4):50–57
47. Santra P, Pande P, Kumar S, Mishra D, Singh R (2017) Agri-voltaics or solar farming: the concept of integrating solar PV based electricity generation and crop production in a single land use system. Int J Renew Energy Res 7:694–699
48. Shehabi A et al. (2016) United States data-center Energy Usage Report. No. LBNL-1005775. Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States)
49. Shehabi A, Masanet E, Price H, Traber K, Horvath A, Nazaroff WW (2011) Data Center Design and Location: Consequences for Electricity Use and Greenhouse-Gas Emissions. Build Environ 46(5):990–998
50. Shehabi A, Smith SJ, Horner N, Azevedo I, Brown R, Koomey J, Masanet E, Sartor D, Herrlin M, Lintner W (2016) United States data-center Energy Usage Report. Lawrence Berkeley National Laboratory, Berkeley, California (**LBNL-1005775**)
51. Sullivan A (2010) Energy Star for data-centers. Green Grid Forum. February 4, 2010
52. Trommsdorff M, Kang J, Reise C, Schindele S, Bopp G, Ehmann A, Weselek A, Hogy P, Obergfell T (2021) Combining food and energy production: Design of an agrivoltaic system applied in arable and vegetable farming in Germany. Renew Sustain Energy Rev 140:110694. https://www.sciencedirect.com/science/article/pii/S1364032120309783

53. Tschudi W, Xu T, Sartor D, Stein J (2003). High Performance Data Centers: A Research Roadmap. Lawrence Berkeley National Laboratory, Berkeley, CA. LBNL53483. http://hightech.lbl.gov/documents/DataCenters_Roadmap_Final.pdf
54. Upton F (2015) North American Energy Security and Infrastructure Act of 2015. H.R. 8, 114th Congress. https://www.congress.gov/bill/114th-congress/house-bill/8
55. US National Renewable Energy Laboratory Website (NREL): https://www.energy.gov/science-innovation/energy-sources/renewable-energy/solar
56. Valle B, Simonneau T, Sourd F, Pechier P, Hamard P, Frisson T, Ryckewaert M, Christophe A (2017) Increasing the total productivity of a land by combining mobile photovoltaic panels and food crops. Appl Energy 206:1495–1507. https://doi.org/10.1016/j.apenergy.2017.09.113
57. Van Heddeghem W, Lambert S, Lannoo B, Colle D, Pickavet M, Demeester P (2014) Trends in worldwide ICT electricity consumption from 2007 to 2012. Comput Commun 50:64–76
58. Weselek A, Ehmann A, Zikeli S, Lewandowski I, Schindele S, Hogy P (2019) Agrophotovoltaic systems: applications, challenges, and opportunities. A review Agron Sustain Dev 39:35
59. Zohdi TI (2022) A digital-twin and machine-learning framework for precise heat and energy management of data-centers. Comput Mech (2022) https://doi.org/10.1007/s00466-022-02152-3
60. Zohdi TI (2018) Dynamic thermomechanical modeling and simulation of the design of rapid free-form 3D printing processes with evolutionary machine learning. Comput Methods Appl Mech Eng 331:343–362
61. Zohdi TI (2019) Electrodynamic machine-learning-enhanced fault-tolerance of robotic free-form printing of complex mixtures. Comput Mech 63:913–929
62. Zohdi TI (2020) A machine-learning framework for rapid adaptive digital-twin based fire-propagation simulation in complex environments. Computer Methods Appl Mech Eng 363:112907
63. Zohdi TI (2021) A digital twin framework for machine learning optimization of aerial fire fighting and pilot safety. Comput Methods Appl Mech Eng 373:113446
64. Zohdi TI (2021) A Digital-Twin and Machine-learning Framework for Ventilation System Optimization for Capturing Infectious Disease Respiratory Emissions. Arch Comput Methods Eng 28:4317–4329. https://doi.org/10.1007/s11831-021-09609-3
65. Zohdi TI (2021) A Digital-Twin and Machine-learning Framework for the Design of Multiobjective Agrophotovoltaic Solar Farms. Comput Mech 68:357–370. https://doi.org/10.1007/s00466-021-02035-z