Original Article

Institution of MECHANICAL ENGINEERS

Journal of MECHANICAL ENGINEERING SCIENCE

Proc IMechE Part C:
J Mechanical Engineering Science
2022, Vol. 236(4) 1943–1959
© IMechE 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0954406220933652
journals.sagepub.com/home/pic
SAGE

Flexible energy load identification in intelligent manufacturing for demand response using a neural network integrated particle swarm optimization

Md Monirul Islam[1], Zeyi Sun[2], Ruwen Qin[3], Wenqing Hu[4], Haoyi Xiong[5] and Kaibo Xu[2]

Abstract

Various demand response programs have been widely established by many utility companies as a critical load management tool to balance the demand and supply for the enhancement of power system stability in smart grid. While participating in these demand response programs, manufacturers need to develop their optimal demand response strategies so that their energy loads can be shifted successfully according to the request of the grid to achieve the lowest energy cost without any loss of production. In this paper, the flexibility of the electricity load from manufacturing systems is introduced. A binary integer mathematical model is developed to identify the flexible loads, their degree of flexibility, and corresponding optimal production schedule as well as the power consumption profiles to ensure the optimal participation of the manufacturers in the demand response programs. A neural network integrated particle swarm optimization algorithm, in which the learning rates of the particle swarm optimization algorithm are predicted by a trained neural network based on the improvement of the fitness values between two successive iterations, is proposed to find the near optimal solution of the formulated model. A numerical case study on a typical manufacturing system is conducted to illustrate the effectiveness of the proposed model as well as the solution approach.

Keywords

Flexible load, manufacturing system, demand response, neural network, particle swarm optimization

Date received: 27 August 2019; accepted: 16 May 2020

Introduction

Due to the growth in the world's population as well as the economy, coupled with rapid urbanization, the demand for electricity will increase substantially over the years. According to the estimation of the United Nations (UN), the world's population will grow from 7.6 billion in 2017 to 9.8 billion by 2050 and two-thirds of the world's people will be settled in the urban areas by 2050 (up from 54% in 2014) because of the process of urbanization.[1] Therefore, energy-consuming activities, such as manufacturing, provision of services, and transportation will increase in scale and also in importance. According to New Energy Outlook 2018, global electricity demand will reach around 38,700 terawatt-hours by 2050 from 25,000 terawatt-hours in 2017.[2]

Based on the information of the U.S. Energy Administration, most of the nation's electricity comes from burning fossil fuels, mostly coal and natural gas. Therefore, the exponential growth of electricity demand is exhausting the fossil fuel supply at an alarming rate. These sources are nonrenewable, thus will be depleted eventually. Another major concern due to the increasing electricity generation from fossil fuels is the impact on environment. Climate change and global warming resulted from greenhouse gas emissions due to the burning of

[1]Department of Industrial Management and Technology, Texas A&M University-Kingsville, Kingsville, TX, USA
[2]Mininglamp Academy of Sciences, Mininglamp Technology, Shanghai, China
[3]Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO, USA
[4]Department of Mathematics and Statics, Missouri University of Science and Technology, Rolla, MO, USA
[5]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA

Corresponding author:
Zeyi Sun, Missouri University of Science and Technology, Fl 29–32, 701 Yunjin Rd, Shanghai 200232, China.
Email: sunzeyi@mininglamp.com

excessive fossil fuel have been identified as the most critical issues of the 21st century. Electricity production generates the second largest share of greenhouse gas emissions. According to the Intergovernmental Panel on Climate Change (IPCC), the world emits approximately 27 gigatons of CO_2 from multiple sources, with electrical production emitting 10 gigatons or approximately 37% of global emissions.[3] Therefore, the challenge of meeting rapidly growing electricity demand, while reducing harmful emissions of greenhouse gases, is significant and therefore, these issues force the electricity sector to seek environmentally clean alternative energy resources.

Renewable energy sources are considered the clean sources of energy. Optimal use of these resources can minimize the environmental impacts and build a sustainable energy infrastructure for current and future economic development as well as societal needs. The most efficient forms of renewable energy sources are solar, wind, geothermal, biomass, and hydroelectricity. Due to the continuous technological innovation and the improvement of the cost-competitiveness of solar photovoltaic and wind power, the renewable sources accounted for an estimated 70% of net additions to global power capacity in 2017. Solar photovoltaic led the way, accounting for nearly 55% of newly installed renewable power capacity in 2017 which is more than the net additions of combined fossil fuels and nuclear power.[4] The remaining addition is from the wind (29%) and hydropower (11%).[5]

The renewable sources are intermittent because they are influenced by natural and meteorological conditions.[6] The variability of the sources introduces uncertainty in generation on the scale of seconds, hours, and days. Therefore, high penetration of the renewable resources is crucial due to the technical challenges including grid interconnection, reliability, power quality, generation dispatch, control, etc.[7] These factors motivate the system operators to develop more flexible and robust architecture in existing grid technology and therefore transform the conventional grid structure into a "smart grid" setting. According to the definition of European Technology Platform, a smart grid is "an electricity network that can intelligently integrate the actions of all users connected to it—generators, consumers, and prosumers in order to efficiently deliver sustainable, economic, and secure electricity supplies."[8] In the last couple of years, a significant number of technological advances on various aspects of smart grid technology and renewable energy integration have been achieved, such as system modeling, simulation, protection, control, stability, and planning for successful and reliable operation.[9–11]

In the market of electricity, where the product cannot be stored and where generators have limits on how quickly they can start and ramp up or down, both supply and demand become more inelastic as the time approaches to real time. Therefore, the markets closer to real time will be more volatile as well as vulnerable to manipulation. If the system runs out of balance, the power stability and quality will be deteriorated, which may trigger the disconnection of system components, and ultimately, lead to power blackouts.

To address the problem and enhance grid reliability through mitigating peak demand and load variability, electric storage resources and demand response programs are two widely adopted methods by utility companies. There have been many previous works focusing on optimally utilizing the energy storage resources to improve the reliability of the smart grid. For example, Rodriguez illustrated the issues and benefits of integrating the energy storage into the smart grid.[12] Bahramirad et al. presented a model for calculating the optimal size of an energy storage system in a microgrid considering the reliability criterion.[13] It has been summarized that the energy storage device is considered the most reliable option for small scale applications; however, it is not economically viable for a large-scale application and is still under investigations for the application on the grid level.[14]

Due to the scalability and less investment requirement, demand response programs have obtained more attention and become a widely adopted demand-side management strategy by the utility companies. Demand-side management is defined as the planning, implementation, and monitoring of utility activities that are designed to influence customer use of electricity.[14–17] This strategy will change the magnitude of the power consumption as well as the time pattern of the electricity usage of the customers. Generally, the objective is to encourage the customers to consume less power during the peak period or to shift energy use to off-peak period to flatten the demand curve. Due to the high penetration of the renewable sources in the smart grid, nowadays, it is more desirable to follow the generation pattern rather than just flattening the demand curve. To maximize the utilization of the generation from the renewable sources and maintain grid reliability, the operators have adopted different types of demand response programs such as time of use (TOU) demand response program, critical peak price (CPP)-based demand response program, and overgeneration-oriented demand response program, considering the requirement of the situation.

While participating in the demand response programs, the energy consumers can shape their loads by adopting the following techniques: peak clipping (reduction in the peak demand),[18] valley filling (increased demand at off-peak),[19] and load shifting (demand shifting to non-peak period).[20] To successfully implement any of the techniques, it is required to have the complete knowledge and control over their use of electricity. The first and foremost criteria to establish the optimal control is to identify the load characteristics existing in their system in terms of

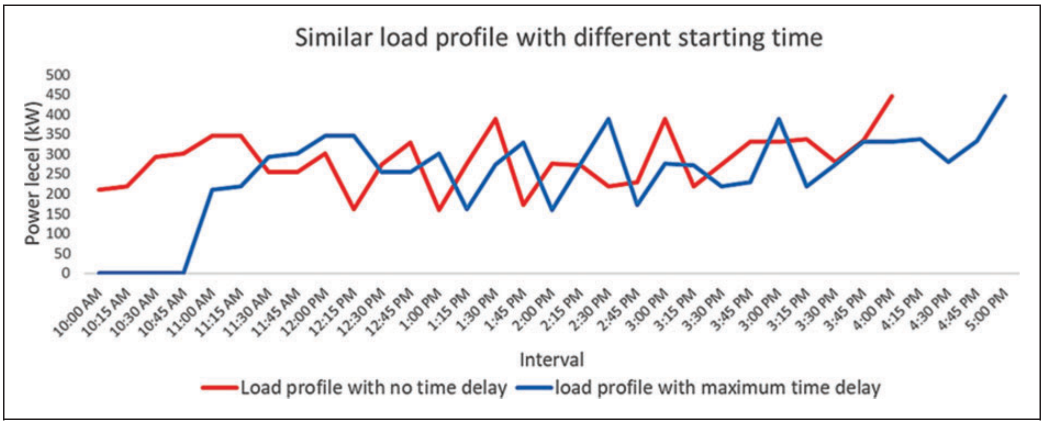


Figure 1. Load profile of flexible loads with earliest and latest starting time.

the flexible and non-flexible loads, their degree of flexibility, and corresponding consumption profile.

The flexible load is defined as the load which can be manipulated, as there is a slackness in its consumption time without affecting the outcomes of the system.[21] Figure 1 shows an illustrative example where the product is required to be delivered by 5 p.m., and the management has a certain level of flexibility in terms of the start time (10:00 a.m. or 10:15 a.m. or 10:30 a.m. or 10:45 a.m. or 11:00 a.m.) of the production. The start time earlier than 11:00 a.m. will be permitted to meet the target production by 5:00 p.m. with a similar power consumption profile. Besides, the non-flexible load has no slackness and the loads are considered the base loads. The successful identification and management of the flexible loads at each sector can reduce the electricity cost substantially through participating in the demand response programs and facilitate the integration of the renewable sources into the smart grid to accelerate the transition toward a clean energy future.

Residential, commercial, and industrial sectors each account for roughly one-third of the nation's electricity use.[22] The industrial sector is considered the main contributor to this increasing trend of electricity demand. Electricity is consumed in this sector by a diverse group of industries including manufacturing, construction, and agriculture, and for a wide range of activities, such as processing and assembly, lighting, and space conditioning. In particular, the manufacturing activities dominate the industrial energy consumption.[23] Therefore, manufacturing industries are the focus of the implementation of the demand response programs.

Generally, the manufacturers have multiple production lines dedicated to the processing different parts of a product. The machines are different with respect to their processing time, rated power, etc., from production line to production line, or sometimes even within the same line. Therefore, considering the utilization of the machines to meet the target production, some production lines can be considered the critical ones where all the machines need to be run throughout the planning horizon while some others are not. The lines that are not critical can be considered the flexible lines and their loads can be utilized to participate in the demand response programs and take the advantage of a substantial reduction of overall energy related cost. While the literature on scheduling the machines in manufacturing systems to minimize the cost during demand response program is extensive,[24–27] very few attempts have been made to shift the considerable amount of load through the identification of the flexible load. Along with the identification of the flexible loads, it is also required to determine the level of the flexibility of the corresponding loads with the load profile.

In this paper, to participate in demand response programs successfully, a mathematical model is proposed for the manufacturers to identify the load characteristic such as non-flexible load (baseload) and flexible load, corresponding profile of each type of load, and the latest start time as a measure of flexibility. Besides, the model will help the manufacturing customers to identify the production schedule of each machine of the flexible line to minimize the maximum as well as the total power consumption level without any loss of production. A neural network-integrated particle swarm optimization (NN-PSO) is implemented to find the near-optimal solution. The learning rates used in particle swarm optimization (PSO) are predicted by the trained neural network according to the improvement of the fitness values between two successive iterations. A numerical case study is conducted to illustrate the effectiveness of the model as well as the proposed algorithm.

The rest of the paper is organized as follows. The upcoming section describes the proposed mathematical model to identify the load type with corresponding load profile and the latest start time. A further section introduces the proposed NN-PSO algorithm. The next section implements a case study to illustrate the effectiveness of the model as well as the proposed algorithm. The penultimate section discusses the results, and the final section draws the conclusion and discusses the future work.

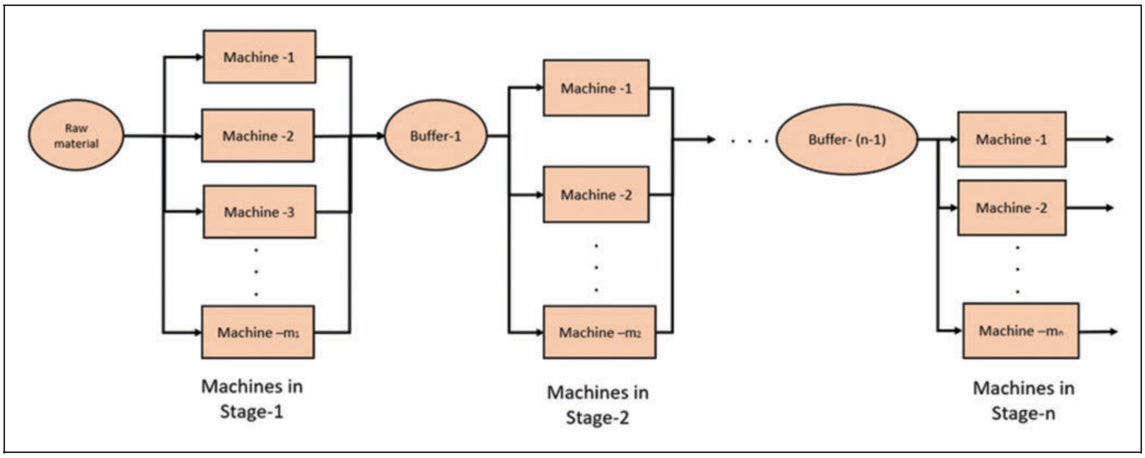


Figure 2. A typical production line.

Proposed model

A typical manufacturing system is modeled in this paper as shown in Figure 2. The system has S manufacturing stages (indexed by s, $s=1, 2\ldots S$). At each stage s, m_s parallel machines (indexed by j, $j=1, 2\ldots m_s$) are installed. The power consumption and processing time for each machine at the same stage are assumed to be identical. Between two consecutive stages, there is a buffer location (indexed by s, $s=1, 2 \ldots S-1$) with finite capacity to store the work-in-process parts.

The length of the planning horizon is assumed to be H hours and is discretized into T intervals (indexed by t, $t=1, 2 \ldots T$) with a constant duration of Δt. In this study, the flexibility is defined as the slackness of starting time to start the production without sacrificing the target production while minimizing the power consumption throughout the planning horizon. Therefore, the objective can be formulated as

$$\max_{x_{sjt}}(IST - PC^{\max}) \tag{1}$$

where IST is the time when the production starts (specified by the index of the discretized interval when production starts), and PC^{max} is the maximum power consumption over the planning horizon. x_{sjt} is the state of machine j in stage s at interval t (one denotes "on" state and zero denotes "off" state).

All the machines at a particular stage have the same processing time and the use of the parallel machines at any stage will start from the machine with the index of one. Therefore, the index of the start time interval for the production system can be determined by selecting the minimum index of the start time interval among all stages, which can be represented by

$$IST = \min_s ST_s \tag{2}$$

where ST_s is the index of the discretized interval at which the production of stage s starts. It can be calculated by

$$ST_s = \left[1 + \sum_{t=1}^{T}(1 - x_{sj1})\cdot(1 - x_{sj2})\ldots(1 - x_{sjt}))\right], \\ j = 1 \tag{3}$$

The power consumption at interval t, PC_t, can be determined by

$$PC_t = \sum_{s=1}^{S}\sum_{j=1}^{m_s} x_{sjt}\cdot PC_{sj} \tag{4}$$

where PC_{sj} is the rated power of machine j in stage s. Therefore, the maximum power over the planning horizon, PC^{max}, can be determined by

$$PC^{\max} = \max_t PC_t \tag{5}$$

It is assumed that the manufacturing system produces only a single type of product. Let p_{sjt} be a binary indicator denoting if the machine j in stage s starts to process a new part at the beginning of interval t or not. It takes the value of one if yes and zero otherwise. The relationship between x_{sjt} and p_{sjt} can be described by

$$x_{sjt} = \sum_{k=1}^{t} p_{sjk}, \quad \forall t \in \{1, 2\ldots PT_{sj}\} \tag{6}$$

$$x_{sjt} = \sum_{k=t-PT_{sj}+1}^{t} p_{sjk}, \quad \forall t \in \{PT_{sj}+1, PT_{sj}+2\ldots T\} \tag{7}$$

where PT_{sj} is the processing time of machine j in stage s.

Total number of parts processed by machine j in stage s up to interval t can be calculated by

$$TP_{sjt} = 0, \quad \forall t \in \{1, 2\ldots PT_{sj}\} \tag{8}$$

$$TP_{sjt} = \sum_{k=1}^{t-PT_{sj}} p_{sjk}, \quad \forall t \in \{PT_{sj}+1, PT_{sj}+2\ldots T\} \tag{9}$$

Let PP be the planned production throughput throughout the planning horizon. It can be calculated using the output of the machines at the last stage of the production line (i.e., $s = S$) at interval T.

$$PP = \sum_{j=1}^{m_S} TP_{SjT} \tag{10}$$

Total work-in-process parts at buffer location s at the beginning of interval t can be calculated by

$$BA_{st} = BA_{s(t-1)} + \sum_{j=1}^{m_s} p_{sjt'} - \sum_{j=1}^{m_s} p_{(s+1)j(t-1)}, \quad t \in \{2,\ 3, \ldots,\ T\},\ s \in \{1,\ 2, \ldots,\ S-1\} \tag{11}$$

where $t' = \max(0, (t - PT_{sj}))$

The constraints can be formulated as follows:

1. As the buffer has limited capacity, the amount of work-in-process parts at buffer location s at any interval t cannot exceed the maximum capacity, which can be represented by

$$0 \leqslant BA_{st} \leqslant BA_s^{\max} \tag{12}$$

where $BA_s^{\max}$ is the maximum buffer capacity of buffer location s.

2. The machine on/off state is constrained by the availability of the work-in-process parts at the immediate upstream buffer location. If there is no work-in-process part, all the downstream machines will be in the ‘‘off’’ state. The machines will be activated (‘‘on’’ state) based on their index (the first machine (indexed by 1) in any stage will be activated if there is only one unit of work-in-process available in the upstream buffer location; the second machine (indexed by 2) will be activated next if there is more than one unit of work-in-process in the buffer, and so on). The constraint can be represented by

$$x_{(s+1)jt} \leqslant BA_{st},\ t \in \{1, 2, \ldots, T\},\ j = 1,\ s \in \{1, 2, \ldots, S-1\} \tag{13}$$

$$x_{(s+1)jt} \leqslant \max(0, (BA_{st} - \sum_{k=1}^{j-1} p_{(s+1)kt})),\ t \in \{1, 2, \ldots, T\},\ j \in \{2, 3, \ldots, m_s\},\ s \in \{1, 2, \ldots, S-1\} \tag{14}$$

3. The index of the start time cannot exceed the index of the last interval of the planning horizon, which can be satisfied by

$$IST \leqslant T \tag{15}$$

4. The production throughput should satisfy the target production (denoted by TA), which can be described by

$$PP \geqslant TA \tag{16}$$

After solving the proposed model, the manufacturer can identify whether the existing load is flexible or not based on the value of the index of the starting time, IST. If the value of the IST is larger than one, the load will be considered as flexible and otherwise non-flexible. In addition to the identification of the status of the load, the manufacturer can also realize the degree of flexibility of the load based on the value of the index of the starting time interval. Apart from this, the load profile obtained from equation (4) will help the manufacturer optimally schedule the machines in each stage of the production line to reduce the maximum power as well as total energy consumption throughout the planning horizon.

Solution approach

It is quite challenging to solve the proposed formulation using the technique of convex optimization due to the non-convexity of the objective function, the involvement of non-linearity in constraints, and the binary/discrete variables. Thus, the use of the analytical optimization techniques is not quite popular to solve such scheduling problems in power system due to the requirement of high computational resources and longer execution time.[28–30] Rather, the metaheuristic computational intelligence algorithms such as PSO, Genetic Algorithm, Ant Colony Algorithm, and Artificial Bee Colony algorithm are used increasingly to obtain quick response in power system problems with many variables.[31–36] Among those techniques, PSO is the one that has been widely used to obtain the near-optimal solutions with a reasonable computational cost.[37–42]

PSO is a typical population-based meta-heuristic algorithm inspired and characterized by the foraging behaviors of animal swarms. It does not require continuity and differentiability on the search space of the optimization problem. The particles of the algorithm move around in the search-space based on the velocity defined by the random parameters w, r_1, and r_2 (shown in the next section), and the differences from the particles' positions to the best positions of corresponding particles as well as the best position of the entire swarm. The search process of each particle will be continued until a satisfactory solution is discovered. As the process follows a heuristic approach, it cannot guarantee global optimality. A detailed description of the PSO algorithm is presented in references.[43–45]

The performance of searching for the best solution in PSO depends on the appropriate exploration and exploitation capability of the particles in the search space. For better exploration, the learning rates r_1 and r_2 used to update the velocity of the particle are usually drawn from the uniform random distribution or based on some empirical values.

Intuitively, the fusion of random and guided selection for the parameters can help to explore and exploit the search space more effectively. The integration of an effective prediction tool (e.g., artificial neural network (ANN)) to predict those parameters based on their historical performance might be helpful to improve the efficiency of traditional PSO since ANN is considered a popular, powerful, and flexible tool for forecasting, providing a better solution to model the complex nonlinear relationships.[46–49] Therefore, in this paper, considering the characteristics of the formulated problem and the motivation to further improve traditional PSO, a neural network integrated particle swarm optimization (NN-PSO), is proposed to solve the problem.

In the remaining part of Solution approach section, a brief description of the continuous and Binary PSO is illustrated in PSO algorithm for continuous variables section and PSO algorithm for binary variables section, respectively, while the detailed steps of the proposed NN-PSO are described in NN-PSO algorithm section.

PSO algorithm for continuous variables

In the basic version of the PSO algorithm, swarm particles keep moving around in a search-space and the particles cluster or converge together around an optimum, or several optima through the combination of their exploration and exploitation. Suppose, the number of particles in the swarm is n in a search space of D-dimension and, then the particle i of the swarm can be represented by a D-dimensional vector: $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ $(i = 1, 2, \ldots, n)$ and the speed of particle i is $V_i = (V_{i1}, V_{i2}, \ldots, V_{iD})$ $(i = 1, 2, \ldots, n)$. The algorithm steps can be described as follows:

Step 1. Randomly initialize a group of n particles with positions and velocities.
Step 2. Calculate the fitness value of each particle.
Step 3. Calculate the fitness value of each particle.
Step 4. Calculate the position of best fitness value of all particles from the global historical movement, which called p_g.
Step 5. Update particles' speed and position using the next two formulas.

$$\begin{aligned} v(t+1) = w \cdot V(t) + c_1 \cdot r_1 \cdot (P_i - x_i(t)) \\ + c_2 \cdot r_2 \cdot (P_g - x_i(t)) \end{aligned} \tag{17}$$

$$x_i(t+1) = x_i(t) + V(t+1) \tag{18}$$

where c_1 and c_2 are positive constants, and r_1 and r_2 are two random variables drawn from the uniform distribution on the interval (0,1). In equation (17), w is the inertia weight which shows the effect of the previous probability vector on the new vector.
Step 6. Go to step 2 and repeat until convergence.

PSO algorithm for binary variables

Khanesar et al. proposed a discrete binary version of PSO for binary problems.[50] In their model, a particle will decide on "yes" or "no," "true" or "false," "include" or "not to include," etc. The P_{ibest} and P_{gbest} of the swarm are updated as a binary version. Two vectors for each particle are introduced as $\vec{V}_i^0$ and $\vec{V}_i^1$. $\vec{V}_i^0$ is the probability of the bits of the particle to change to zero while $\vec{V}_i^1$ is the probability that bits of particle change to one. Therefore, for D-dimensional space, $\vec{V}_i^0$ and $\vec{V}_i^1$ can be represented by the vectors $\vec{V}_i^1 = [V_{i1}^1, V_{i2}^1, V_{i3}^1, \ldots V_{iD}^1]$, and $\vec{V}_i^0 = [V_{i1}^0, V_{i2}^0, V_{i3}^0, \ldots V_{iD}^0]$, respectively. The probability of change in jth bit of ith particle is simply defined as follows

$$V_{ij}^c = \begin{cases} V_{ij}^1, \text{ if } x_{ij} = 0 \\ V_{ij}^0, \text{ if } x_{ij} = 1 \end{cases} \tag{19}$$

Assume that the best position visited so far for a particle is P_{ibest} and the global best position for the particle is P_{gbest}. Also, consider that the jth bit of ith particle is one. So, to guide the jth bit of ith particle to its best position, the probability of change to one ($\dot{V}_i^1$) for that particle increases and the probability of change to zero ($\dot{V}_i^0$) decreases if the particle's best position P_{gbest} is one. Based on the concept, the following rules are derived.

$$\begin{aligned} &\text{if } P_{ipbest}^j = 1 \text{ then } d_{ij,1}^1 = c_1 \cdot r_1 \text{ and } d_{ij,1}^0 = -c_1 \cdot r_1 \\ &\text{if } P_{ipbest}^j = 0 \text{ then } d_{ij,1}^0 = c_1 \cdot r_1 \text{ and } d_{ij,1}^1 = -c_1 \cdot r_1 \\ &\text{if } P_{igbest}^j = 1 \text{ then } d_{ij,2}^1 = c_2 \cdot r_2 \text{ and } d_{ij,2}^0 = -c_2 \cdot r_2 \\ &\text{if } P_{igbest}^j = 0 \text{ then } d_{ij,2}^0 = c_2 \cdot r_2 \text{ and } d_{ij,2}^{\prime} = -c_2 \cdot r_2 \end{aligned} \tag{20}$$

where $d_{ij,1,}^1$ and $d_{ij,1}^0$ are two temporary values. r_1 and r_2 are two random learning rates in the range of (0,1) which are updated at each iteration. c_1 and c_2 are two fixed variables which are determined by the user. Each element of the vector V_{ij}^0 and V_{ij}^1 can be calculated by

$$V_{ij}^0 = w \cdot V_{ij}^0 + d_{ij,1}^0 + d_{ij,2}^0 \tag{21}$$

$$V_{ij}^{1} = w \cdot V_{ij}^{1} + d_{ij,1}^{1} + d_{ij,2}^{1} \tag{22}$$

where w is the inertia term. In fact, in this algorithm if the jth bit in the global best variable is zero or if the jth bit in the corresponding personal best variable is zero, the probability (V_{ij}^{0}) will be increased. And the probability of changing to one will also decrease at the same rate. Besides, if the jth bit in the global best variable is one, V_{ij}^{1} increases and V_{ij}^{0} decreases. The probability of change is obtained using equation (19).

The normalization process is done by using the sigmoid function which is presented by

$$c(t) = sig(V_{ij}^{c}(t)) = \frac{1}{1 + e^{V_{ij}^{c}(t)}} \tag{23}$$

The next particles state is computed by

$$x_{ij}(t+1) = \begin{cases} \overline{x}_{ij}(t), & \text{if } l_{ij} < V_{ij}'' \\ x_{ij}(t), & \text{if } l_{ij} > V_{ij}' \end{cases} \tag{24}$$

where $\bar{x}_{ij}(t)$ is the complement of the $x_{ij}(t)$. l_{ij} is random value between (0,1).

The Binary PSO steps are as follows:

Step 1 Initialize the swarm X_i, the position of particles is randomly selected from binary values 0 and 1.
Step 2 Evaluate the performance F of each particle, using its current position $X_i(t)$.
Step 3 Compare the performance of each individual to its best performance so far:

$$\text{if } F(X_i(t)) < F(P_{ibest}(t)) \colon F(P_{ibest}(t)) = F(X_i(t))$$
$$P_{ibest}(t) = X_i(t)$$

Step 4 Compare the performance of each particle to the global best particle:

$$\text{if } F(X_i(t)) < F\left(P_{gbest}\right) \colon F\left(P_{gbest}\right) = F(X_i(t))$$
$$P_{gbest} = X_i(t)$$

Step 5 Change the probability of the particle, $\vec{V}_i^{0}$ and $\vec{V}_i^{1}$ according to equations (21) and (22).
Step 6 Calculate the probability of change of the bits, $\vec{V}_{ij}^{c}$ as in equation (19).
Step 7 Generate the random variable (l_{ij}) in the range: (0, 1). Move each particle to a new position using equation (24).
Step 8 Go to step 2 and repeat until convergence.

NN-PSO algorithm

In the Binary PSO algorithm, the movements of the particles are guided by the best position of each particle in the search-space and the best position of the entire swarm. To explore and exploit the search space successfully, the learning parameters w, r_1 and r_2 are chosen randomly. Interestingly, the degree of randomness for the parameters becomes the same throughout the entire process of the optimization. Therefore, the particles keep moving irrespective of the proximity of the optima. In many cases, it may get the algorithm stuck in a bad local optimum due to the inappropriate direction generated by the random parameters. As a result, the number of iterations typically increases to reach the global optima and most of the cases, poor solutions are generated.

To overcome the shortcoming of the Binary PSO, the NN-PSO algorithm is proposed in this study to solve the problem. In this proposed algorithm, the Binary PSO is run for a small number of iterations at the beginning and the corresponding features related to the learning parameters (r_1 and r_2) are stored to train the neural network. Later, the trained neural network will be used to predict the r_1 and r_2 values to further guide the PSO algorithm. Therefore, this fusion of random and guided search will help the PSO algorithm to converge faster as well as generate a better solution than the Binary PSO algorithm. The flow diagram and detailed algorithm steps are illustrated in Steps of the NN-PSO algorithm section. The specific structure of NN implemented to guide the PSO algorithm is exhibited in Specific structure of NN for guiding the PSO algorithm section.

Steps of the NN-PSO algorithm. To visually organize the step-by-step process of the NN-PSO algorithm, a flowchart of the algorithm is illustrated in Figure 3.

From Figure 3, the detailed steps of the NN-PSO approach for the problem can be described as follows:

Step 1 Initialize the swarm X_i, the position of particles is randomly selected from binary values 0 and 1.
Step 2 Evaluate the performance F of each particle, using its current position $X_i(t)$.
Step 3 Compare the performance of each individual to its best performance so far:

$$\text{if } F(X_i(t)) < F(P_{ibest}(t)) \colon F(P_{ibest}(t)) = F(X_i(t))$$
$$P_{ibest}(t) = X_i(t)$$

Step 4 Compare the performance of each particle to the global best particle:

$$\text{if } F(X_i(t)) < F\left(P_{gbest}\right) \colon F\left(P_{gbest}\right) = F(X_i(t))$$
$$P_{gbest} = X_i(t)$$

Step 5 Change the probability of the particle, $\vec{V}_i^{0}$ and $\vec{V}_i^{1}$ according to equations (20) to (22) where r_1 and r_2 are selected randomly between (0, 1)
Step 6 Calculate the probability of change of the bits, $\vec{V}_{ij}^{c}$ as in equation (19).

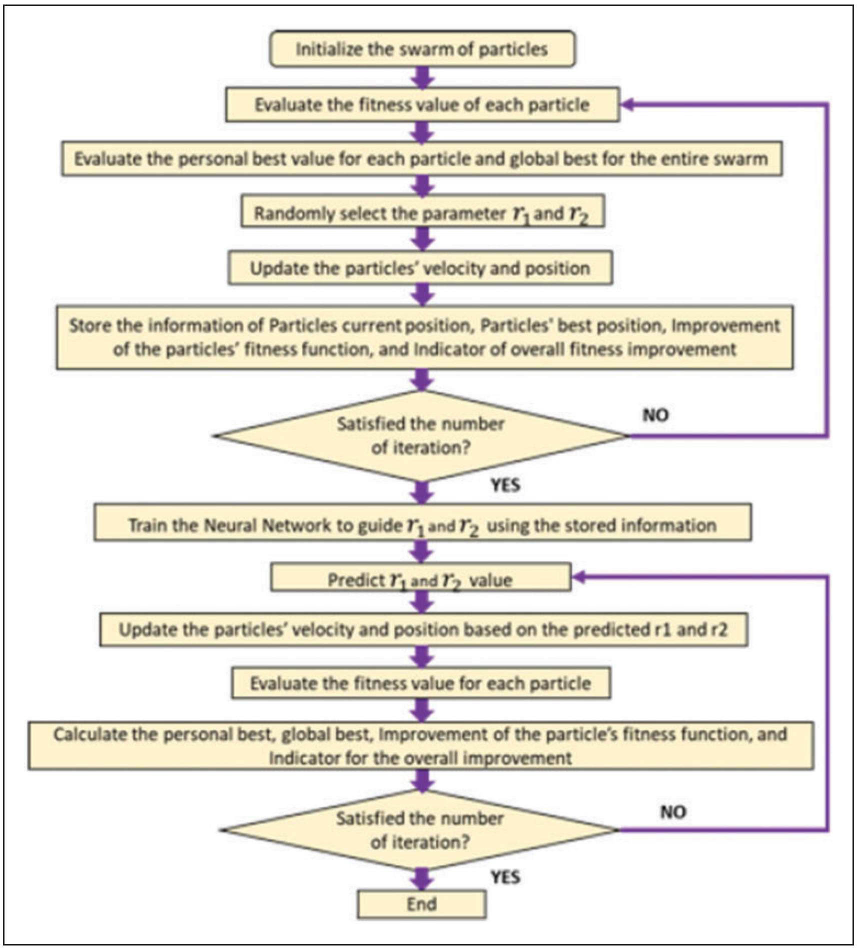

Initialize the swarm of particles
Evaluate the fitness value of each particle
Evaluate the personal best value for each particle and global best for the entire swarm
Randomly select the parameter r_1 and r_2
Update the particles' velocity and position
Store the information of Particles current position, Particles' best position, Improvement of the particles' fitness function, and Indicator of overall fitness improvement
Satisfied the number of iteration?
NO
YES
Train the Neural Network to guide r_1 and r_2 using the stored information
Predict r_1 and r_2 value
Update the particles' velocity and position based on the predicted r1 and r2
Evaluate the fitness value for each particle
Calculate the personal best, global best, Improvement of the particle's fitness function, and Indicator for the overall improvement
Satisfied the number of iteration?
NO
YES
End

Figure 3. Flow diagram illustrating the NN-PSO algorithm.

Step 7 Generate the random variable (l_{ij}) in the range: (0, 1). Move each particle to a new position using equation (24).

Step 8 Go to step 2 and repeat for a small number of iterations.

Step 9 Train the neural network model for the dataset obtained after step 8. $X_i(t)$, P_{ibest}, and δh (improvement of the objective function at each iteration) are used as input for predicting r_1 while $X_i(t)$, P_{gbest}, and δh for r_2.

Step 10 Import the $P_{ibest}(t)$, P_{gbest}, and particles' position from the last iteration at step 8.

Step 11 Predict r_1 and r_2 using the trained network obtain at step 10.

Step 12 Change the probability of the particle, $\vec{V}_i^0$ and $\vec{V}_i^1$ according to equations (20) to (22).

Step 13 Calculate the probability of change of the bits, $\vec{V}_{ij}^c$ as in equation (19).

Step 14 Generate the random variable (l_{ij}) in the range: (0, 1). Move each particle to a new position using equation (24).

Step 15 Compare the performance of each individual to its best performance so far:

$$\text{if } F(X_i(t)) < F(P_{ibest}(t)) : F(P_{ibest}(t)) = F(X_i(t))$$
$$P_{ibest}(t) = X_i(t)$$

Step 16 Compare the performance of each particle to the global best particle:

$$\text{if } F(X_i(t)) < F\left(P_{gbest}\right) : F\left(P_{gbest}\right) = F(X_i(t))$$
$$P_{gbest} = X_i(t)$$

Step 17 Go to step 11 and repeat until converges.

Specific structure of NN for guiding the PSO algorithm. The NN architecture developed to solve the problem is shown in Figure 4. The network has 50 fully connected hidden layers with 50 neurons at each layer. The dimension of the input vector is $(S \times m_s \times T + S \times m_s \times T + 2)$. The first $S \times m_s \times T$ elements represent the particles' current positions. The second $S \times m_s \times T$ elements represent the particles' best positions while training for r_1. When training r_2, the second $S \times m_s \times T$ elements represent the particles' global best positions. The last two elements of the vector represent the improvement of the fitness function compared to the last iteration and status of the generated solution (good or bad). Rectified linear unit (ReLU) ($f(x) = \max(0, x)$) is used as an activation function as it allows faster convergence.[51] As it is a

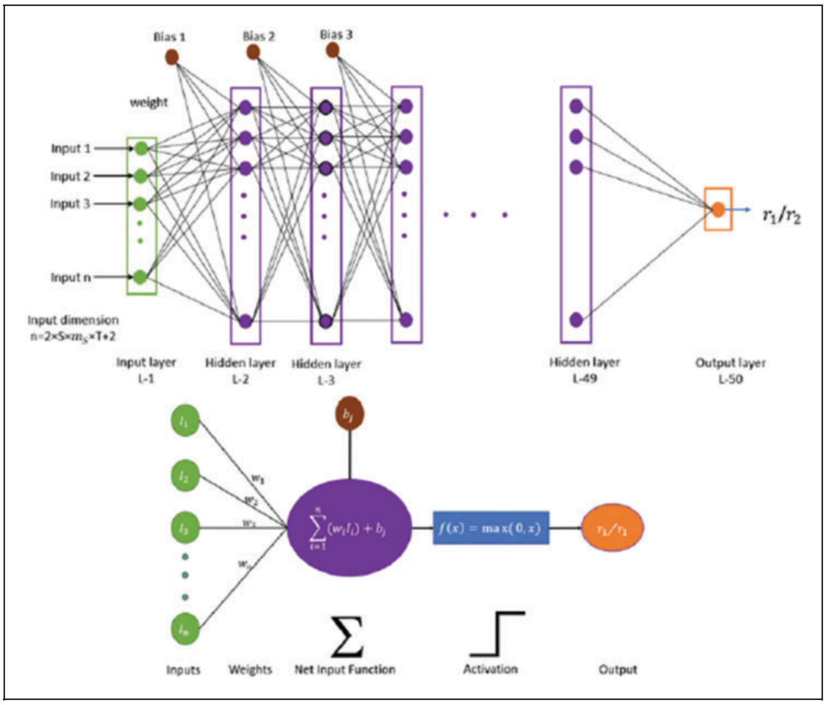


Figure 4. Neural network architecture.

regression problem, no activation function is used for the output layer. The efficient ADAM optimization algorithm is used as an optimizer and mean squared error is used as a cost function.

Case study

Consider a three-stage manufacturing system as shown in Figure 5. At each stage of the production line, there are three machines with the same processing time and rated power as illustrated in Table 1. The initial buffer amount and corresponding maximum capacity of each buffer location are shown in Table 2.

The planning horizon is considered 8 h (from 9:00 a.m. to 5:00 p.m.) and it is discretized into a set of 15-min intervals. The target production for the planning horizon is 60 units.

The Binary PSO and NN-PSO described in the previous section are implemented in this section to solve the case study and compared their performance in the next section. The algorithms are encoded in Python. The hardware used to implement the algorithms is a desktop with an Intel(R) Xeon(R) CPU W3505@ 2.53 GHz processor and a 16 GB memory.

Results and discussion

Implications of the results to demand response programs

The latest start time and corresponding power consumption level at each interval obtained from the NN-PSO algorithm are shown in Figure 6.

From Figure 6, it can be seen that the latest start time to begin the production is the fifth interval of the planning horizon which indicates that there exists a slackness of time to begin the production. The production system can be considered the flexible one and the degree of flexibility is up to the fifth interval of the planning horizon. If the system starts their production earlier than the fifth interval (10:15 a.m. as the planning horizon starts from 9:00 a.m.), the production will be completed before 5:00 p.m. (ending time of the planning horizon), which may not create any extra value for the manufacturers. However, the manufacturers can utilize the slackness to minimize their energy-related cost by participating in different demand response programs such as TOU demand response program, CPP-based demand response program, and overgeneration-oriented demand response program.

These demand response programs are different in their natures and efficacies. To maintain the grid reliability and enhance the power quality, the grid operators introduce the appropriate demand response program considering the characteristics of the situation. For instance, the TOU-based demand response program is usually designed to incentivize the customers to use more energy at off-peak times and less energy at on-peak times to balance the demand and supply. The CPP-based demand response program is introduced by the grid when the generation falls unexpectedly and substantially due to specific triggering events, e.g., extreme high temperature in a summer afternoon and hardware equipment breakdowns.[52] Manufacturers need to ensure their optimal participation to reduce the stress on the grid and high electricity prices.

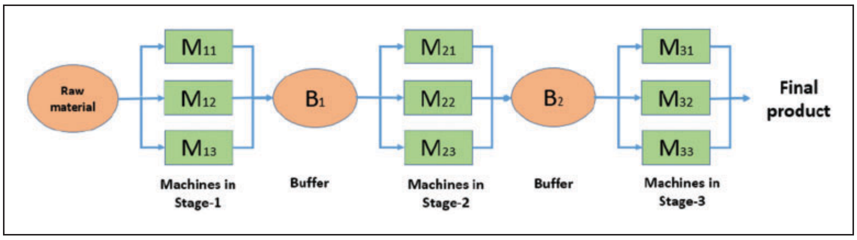


Figure 5. Three-stage production line for case study.

Table 1. Processing time and power consumption level of each machine at different stage.

Stage	S-1			S-2			S-3		
Machine index	M-1	M-2	M-3	M-1	M-2	M-3	M-1	M-2	M-3
Processing time (min)	15	15	15	15	15	15	15	15	15
Power consumption rate (kW)	37	37	37	43	43	43	33	33	33

Table 2. Initial buffer amount and corresponding maximum capacity at each buffer location.

Stage	B-1	B-2
Initial buffer amount	4	6
Buffer capacity	15	9

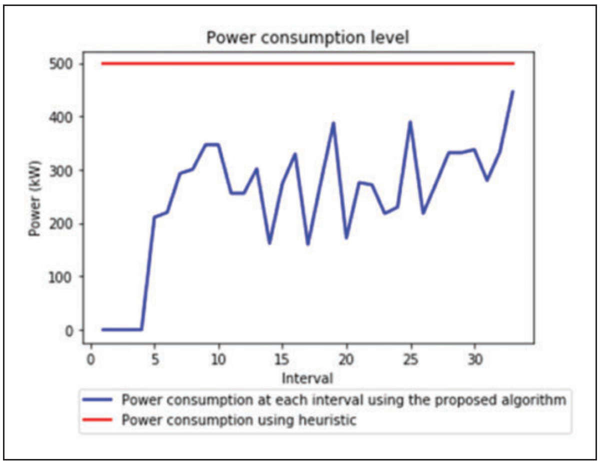


Figure 6. Power consumption level at each interval and latest starting time.

In addition, another demand response program that has recently become more popular in the smart grid environment is the overgeneration-oriented demand response program. This type of demand response program can be observed when the conventional power system is not able to accommodate the ramp rate and range needed to fully utilize renewable energy.[53] In such demand response programs, the grid operators design the incentive and penalty mechanisms to attract more energy consumption during the over-generation periods. Based on the scheme of the program and maximum consumption level of the manufacturing, the manufacturer decides on the participation in the program. If the manufacturer is committed to participating, it will be required to adjust their energy consumption patterns to reach such committed consumption levels so that the incentive can be obtained, and the penalty can be avoided. Through successful participation in the demand response program, the users can reduce their total energy consumption substantially.[38,54–56]

Furthermore, the identification of flexible load and successful participation in the demand response programs will eventually facilitate the integration of the renewable sources into power system which will help to meet the carbon challenges of 2050.[57] The reference[58] shows that the generation of 1 kW of energy from burning the coal produces 2.21 pounds of CO_2. To capture CO_2 in power stations, the cost is estimated to be approximately \$30–50 per ton CO_2 of emissions (\$110–180/ t C), being equivalent to an increase of about 50% in the cost of electricity.[59] From the cost perspective, it can be devised that about \$0.525 economic benefit can be achieved by consuming 1 kWh of renewable energy instead of the fossil fuel energy. For the case study, it is assumed that due to the identification of the flexible load and their corresponding flexibility, the manufacturer can shift their load which facilitates them to meet the energy demand by utilizing the renewable sources rather than fossil fuel-based sources. Therefore, the utilization of the renewable energy for flexible time (four intervals) reduces the amount of CO_2 emission for the corresponding intervals and produces economic benefits. As the average power consumption for the intervals is 283 kW, the total estimated economic benefit due to CO_2 reduction is around \$125.

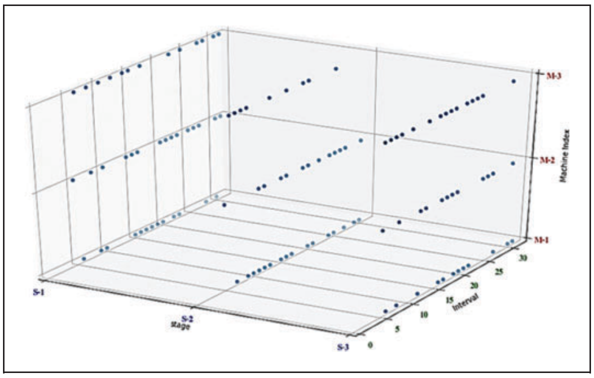


Figure 7. Machine status throughout the planning horizon.

Table 3. Comparison of the maximum power and total energy consumption between proposed model and simple heuristic.

	Maximum power (kW)	Total energy consumption (kWh)
Proposed model	446	2058.75
Simple heuristic	498	2988
Reduction	10.44%	31.09%

Comparison to baseline scenarios

The corresponding machine scheduling for the production system throughout the planning horizon is shown in Figure 7.

The total power consumption while maintaining the proposed scheduling is further compared with a simple heuristic where the manufacturer will keep all their machines in the "ON" state to meet the production target without considering the maximum power level and total energy consumption. The comparison of the maximum power and total energy consumption is shown in Table 3. It can be observed that there are reduction opportunities of maximum power by 10.44 % and total energy consumption by 31.09%, which will significantly reduce the overall energy cost for the production.

Additionally, the performance of the proposed NN-PSO is investigated and compared with the Binary PSO proposed by Khanesar et al.[50] The comparison of the performance in terms of fitness improvement and corresponding convergence time are shown in Table 4.

From Table 4, it can be concluded that the exploration and exploitation capability of the NN-PSO algorithm is better than the Binary PSO algorithm. This is due to the fact that the integration of the neural network helps the algorithm to learn and generalize the features of the parameters r_1 and r_2 responsible for the improvement of the fitness function. Furthermore, the trained network is utilized to predict r_1 and r_2 for future steps of the PSO algorithm rather than a simple random/empirical selection. Integration of this guided search along with the random exploration at the beginning steps helps the algorithm find better optima rather than fluctuating near the local optima. Therefore, the proposed NN-PSO outperforms the Binary PSO in terms of the fitness improvement in all the conditions shown in Table 4. However, the convergence time varies based on the population size of the PSO. For the smaller number of population size, the Binary PSO is faster than the NN-PSO algorithm due to the additional time required for training the neural network. However, for a larger population size, the Binary PSO algorithm requires more time to converge as the convergence steps for the NN-PSO algorithm is significantly less than the additional time required for training the network.

Examine a large size formulation

The case study is further extended to a larger size case where there are six different stages in the production line and at each stage, six machines are installed. The target production of the line is still 60 units for the specific planning horizon. The processing time and power consumption level are similar to the values in Table 1. The parameters related to the buffer location are shown in Table 5. The NN-PSO and Binary PSO algorithms are implemented to find the latest start time and corresponding machine schedule for the planning horizon. To find the effectiveness of the algorithm, the performance metrics are compared in Table 6.

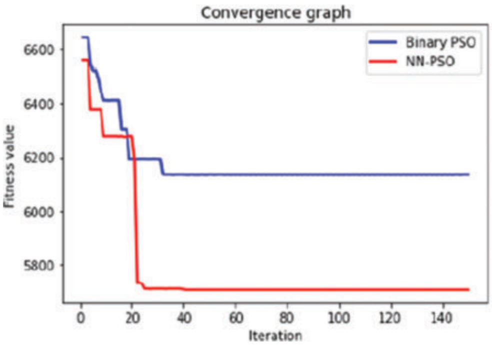
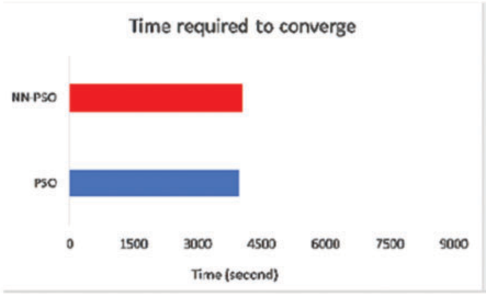
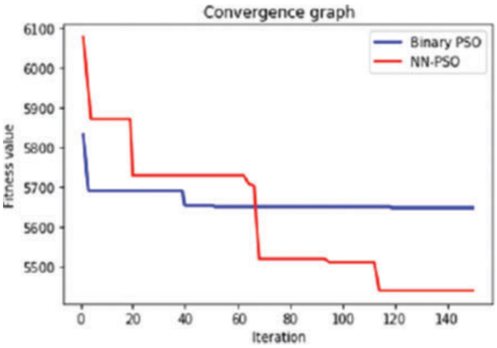
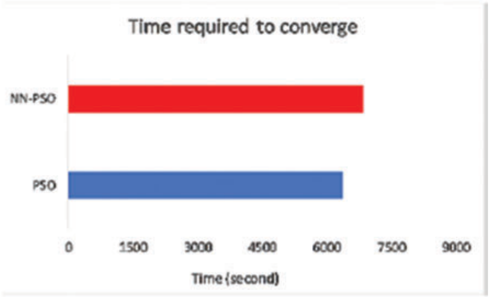
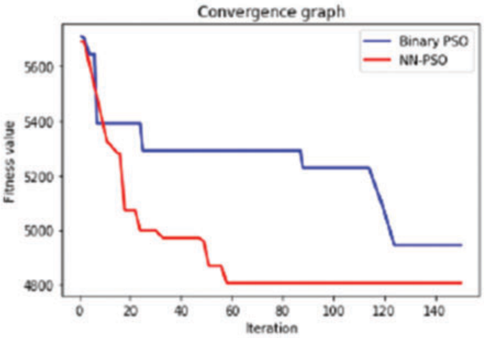
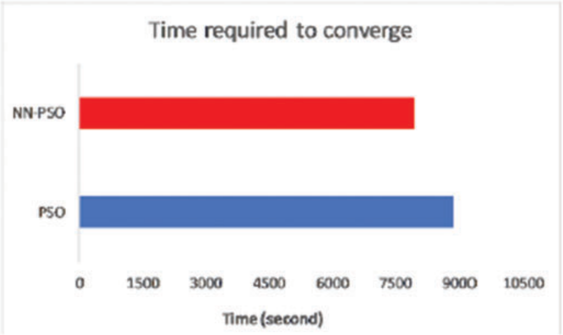
From Table 6, it is clearly visible that the NN-PSO algorithm performs better than the Binary PSO algorithm regarding the fitness improvement and convergence time.

Comparison to LSTM-PSO

Finally, a particular category of recurrent neural networks model called the long short-term memory (LSTM) network model is implemented to predict the parameters (r_1 and r_2) and compared with the performance of the fully connected neural network as shown in Table 7. The LSTM network is quite popular for the prediction problem due to the incorporation of the temporal dynamics in its structure. A detailed description of the LSTM network can be found in.[60–63] The architecture of the LSTM network implemented to solve the problem is shown in Figure 8. Each LSTM cell is illustrated in Figure 9.

The Keras framework is used to develop the network. The optimal choice of the network is based on several trials and errors. The proposed network architecture includes eight stacked LSTM layers followed by two dense layers and an output layer. Each LSTM layer consists of 128 LSTM cells. The dense layers have 64 and 32 neurons, respectively, and they are

Table 4. Comparison of performance between Binary PSO and NN-PSO.

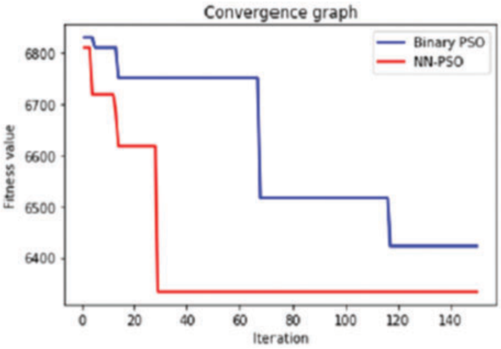
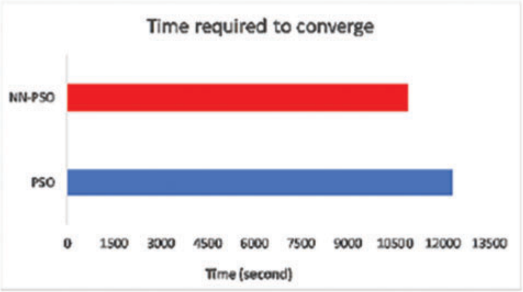
Condition	Convergence graph	Convergence time
Condition-I (popsize: 50)		
	Performance gap Fitness improvement: 7.01%	Time reduction: −11.34%
Condition-II (popsize: 100)		
	Performance gap Fitness improvement: 7.01%	Time reduction: −7.14%
Condition-III (popsize: 200)		
	Performance gap Fitness improvement: 2.8%	Time reduction: 10.83%

NN-PSO: neural network integrated particle swarm optimization.

Table 5. Initial buffer amount and corresponding maximum capacity at each buffer location in a large size case.

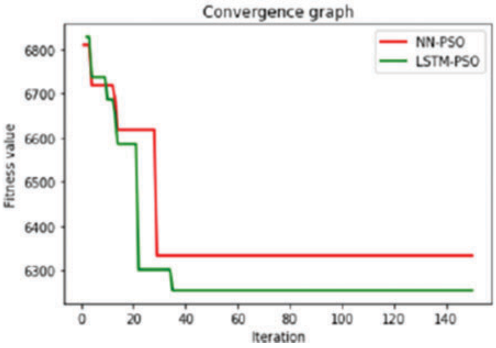
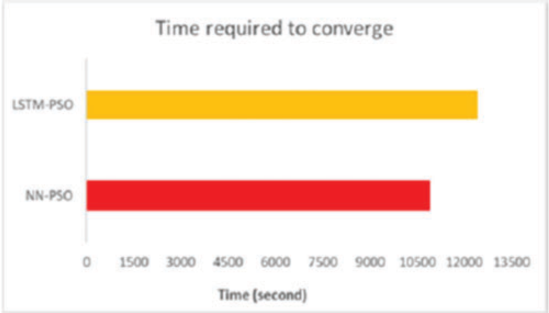
Stage	B-1	B-2	B-3	B-4	B-5
Initial buffer amount	4	12	8	7	9
Buffer capacity	25	19	25	17	18

Table 6. Comparison of performance between binary PSO and NN-PSO for a large size case.

Condition	Convergence graph	Convergence time
Popsize: 200		
	Performance gap Fitness improvement: 7.01%	Time reduction: 11.65%

NN-PSO: neural network integrated particle swarm optimization.

Table 7. Comparison of performance between NN-PSO and LSTM-PSO.

Condition	Convergence graph	Convergence time
Popsize: 200		
	Performance gap Fitness Improvement: 1.25%	Time reducton: −12.41%

NN-PSO: neural network integrated particle swarm optimization; LSTM-PSO: long short-term memory integrated particle swarm optimization.

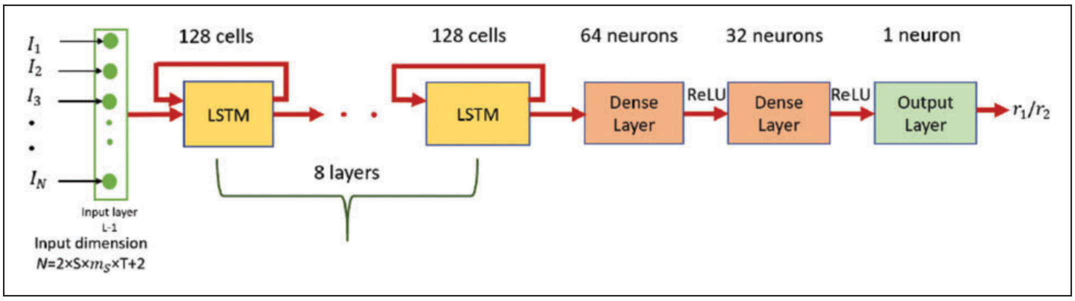


Figure 8. LSTM architecture.
LSTM-PSO: long short-term memory; ReLU: rectified linear unit.

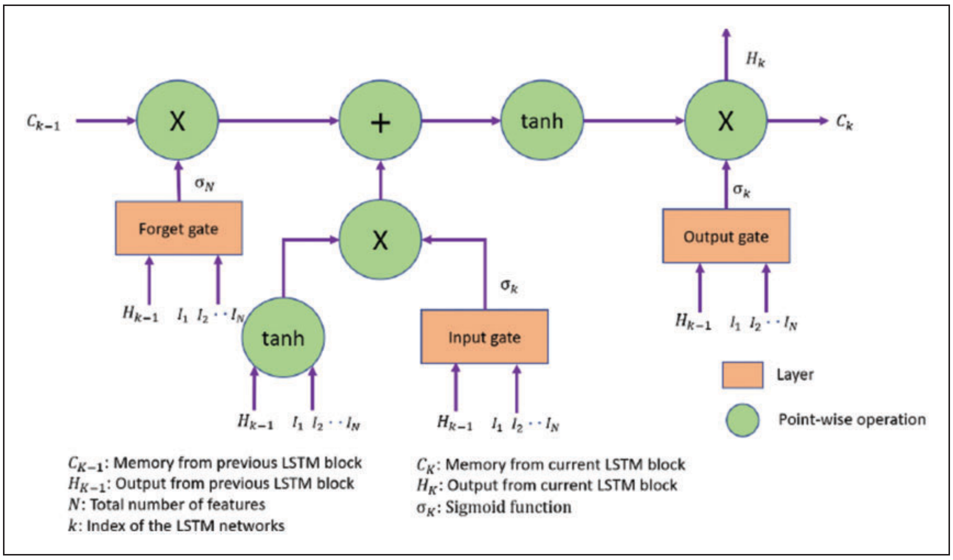

C_{k-1}
X
+
tanh
X
H_k
C_k
σ_N
Forget gate
X
σ_k
Output gate
H_{k-1} I_1 $I_2 \cdot\cdot I_N$
tanh
σ_k
Input gate
H_{k-1} I_1 $I_2 \cdot\cdot I_N$
H_{k-1} I_1 $I_2 \cdot\cdot I_N$
H_{k-1} I_1 $I_2 \cdot\cdot I_N$
Layer
Point-wise operation
C_{K-1}: Memory from previous LSTM block
H_{K-1}: Output from previous LSTM block
N: Total number of features
k: Index of the LSTM networks
C_K: Memory from current LSTM block
H_K: Output from current LSTM block
σ_K: Sigmoid function

Figure 9. LSTM cell.

fully connected. The final dense output layer contains a single neuron to predict r_1 or r_2. The dropout technique is used for regularization.

From Table 7, it can be seen that the LSTM-PSO performs marginally better than the NN-PSO algorithm regarding the fitness improvement. However, the NN-PSO is much faster than the LSTM-PSO due to its simple structure and a smaller number of weights carried over from previous steps.

Conclusions

In this paper, a mathematical model is developed to identify the load characteristic of the manufacturing system in terms of the non-flexible and flexible loads, corresponding power consumption profile, and the latest start time. The identification of the flexible loads and latest start time will help manufacturers measure their load shifting capability during the demand response period. Moreover, the proposed model will create an opportunity for the manufacturers to shift a considerable amount of load as the identification of the flexible load is at the level of a production line or a subsystem including all machines in the system. As a result, significant cost reduction is possible if there exists a considerable amount of the flexible loads in the system. Besides, the proposed neural network integrated PSO algorithm will help to find better optima compared to the existing PSO algorithm for binary problems.

In future, the smart grid can build a new market structure where customers like manufacturing system can directly participate in the day-ahead market by providing the information of the non-flexible load (baseload) and flexible load, corresponding power consumption profile, and the degree of flexibility. It may help the grid authority to develop an efficient day ahead purchase plan as well as flexible dispatch planning and therefore, the customers can get cheaper electricity price compared to the present structure.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Zeyi Sun https://orcid.org/0000-0003-0704-2708
Ruwen Qin https://orcid.org/0000-0003-2656-8705

References

1. World Nuclear Association. World energy needs and nuclear power, www.world-nuclear.org/information-library/current-and-future-generation/world-energy-needs-and-nuclear-power.aspx (2019, accessed 27 May 2020).
2. Bloomberg. Global electricity demand to increase 57% by 2050, https://about.bnef.com/blog/global-electricity-demand-increase-57-2050/ (2018, accessed 27 May 2020).
3. Center for Climate and Energy Solution. Global emission, www.c2es.org/content/international-emissions/ (2018, accessed 27 May 2020).
4. Kabir E, Kumar P, Kumar S, et al. Solar energy: potential and future prospects. *Renew Sustain Energy Rev* 2018; 82: 894–900.
5. Renewables. Global status report, www.ren21.net/wp-content/uploads/2018/06/17-8652_GSR2018_Full Report_web_-1.pdf (2018, accessed 27 May 2020).
6. Wang XY, Vilathgamuwa DM and Choi SS. Determination of battery storage capacity in energy

buffer for wind farm. *IEEE Trans Energy Convers* 2008; 23: 868–878.

7. Teleke S, Baran ME, Huang AQ, et al. Control strategies for battery energy storage for wind farm dispatching. *IEEE Trans Energy Convers* 2009; 24: 725–732.
8. United Nations Economic Commission for Europe. Electricity system development: a focus on smart grids, www.unece.org/fileadmin/DAM/energy/se/pdfs/eneff/eneff_h.news/Smart.Grids.Overview.pdf (2015, accessed 27 May 2020).
9. Wei W, Shengwei M, Feng L, et al. Smart scheduling of power system under energy saving policy and renewable energy integration. In: *Proceedings of the 31st Chinese control conference*. Piscataway: IEEE, 2012, pp.6809–6814.
10. Joo JY, Raghavan S and Sun Z. Integration of sustainable manufacturing systems into smart grids with high penetration of renewable energy resources. In: *2016 IEEE green technologies conference (GreenTech)*. Piscataway: IEEE, 2016, pp.12–17.
11. Islam M, Sun Z and Dagli C. Simulation-based investigations on electricity demand response for manufacturing systems to mitigate overgeneration due to high penetration of renewable sources. *DEStech Trans Eng Technol Res*. DOI: 10.12783/dtetr/icpr2017/17613.
12. Rodriguez GD. A utility perspective of the role of energy storage in the smart grid. In: *IEEE PES general meeting*. Piscataway: IEEE, 2010, pp.1–2.
13. Bahramirad S, Reder W and Khodaei A. Reliability-constrained optimal sizing of energy storage system in a microgrid. *IEEE Trans Smart Grid* 2012; 3: 2056–2062.
14. Castillo A and Gayme DF. Grid-scale energy storage applications in renewable energy integration: a survey. *Energy Convers Manag* 2014; 87: 885–894.
15. Gelazanskas L and Gamage KA. Demand side management in smart grid: a review and proposals for future direction. *Sustain Cities Soc* 2014; 11: 22–30.
16. Sun Z and Li L. Potential capability estimation for real time electricity demand response of sustainable manufacturing systems using Markov decision process. *J Clean Prod* 2014; 65: 184–193.
17. Sun Z, Li L, Fernandez M, et al. Inventory control for peak electricity demand reduction of manufacturing systems considering the tradeoff between production loss and energy savings. *J Clean Prod* 2014; 82: 84–93.
18. Sinha A, Neogi S, Lahiri RN, et al. Role of demand side management for power distribution utility in India. In: *2011 IEEE power and energy society general meeting*. Piscataway: IEEE, 2011, pp.1–8.
19. Logenthiran T, Srinivasan D and Shun TZ. Demand side management in smart grid using heuristic optimization. *IEEE Trans Smart Grid* 2012; 3: 1244–1252.
20. Hamid E, Nallagownden P, Nor NB, et al. Intelligent demand side management technique for industrial consumer. In: *2014 5th international conference on intelligent and advanced systems (ICIAS)*. Piscataway: IEEE, 2014, pp.1–6.
21. Vieira GE, Herrmann JW and Lin E. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *J Schedul* 2003; 6: 39–62.
22. United States Environmental Protection Agency. Energy and the environment, www.epa.gov/energy/electricity-customers (2017, accessed 27 May 2020).
23. Duflou JR, Sutherland JW, Dornfeld D, et al. Towards energy and resource efficient manufacturing: a processes and systems approach. *CIRP Ann* 2012; 61: 587–609.
24. Fernandez M, Li L and Sun Z. "Just-for-Peak" buffer inventory for peak electricity demand reduction of manufacturing systems. *Int J Prod Econ* 2013; 146: 178–184.
25. Moon JY, Shin K and Park J. Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency. *Int J Adv Manuf Technol* 2013; 68: 523–535.
26. Ding JY, Song S, Zhang R, et al. Parallel machine scheduling under time-of-use electricity prices: new models and optimization approaches. *IEEE Trans Autom Sci Eng* 2015; 13: 1138–1154.
27. Che A, Zhang S and Wu X. Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs. *J Clean Prod* 2017; 156: 688–697.
28. Soares J, Silva M, Sousa T, et al. Distributed energy resource short-term scheduling using signaled particle swarm optimization. *Energy* 2012; 42: 466–476.
29. Kozyrev VP. Estimation of the execution time in real-time systems. *Progr Comput Softw* 2016; 42: 41–48.
30. Betts A. *Hybrid measurement-based WCET analysis using instrumentation point graphs*. Doctoral dissertation, University of York, UK, 2008.
31. Molavi H and Ardehali MM. Utility demand response operation considering day-of-use tariff and optimal operation of thermal energy storage system for an industrial building based on particle swarm optimization algorithm. *Energy Build* 2016; 127: 920–929.
32. Kinhekar N, Padhy NP and Gupta HO. Particle swarm optimization based demand response for residential consumers. In *2015 IEEE power & energy society general meeting*. Piscataway: IEEE, 2015, pp.1–5.
33. Zhu R. The approach of genetic algorithms application on reactive power optimization of electric power systems. In: *2017 4th international conference on machinery, materials and computer (MACMC 2017)*. Paris: Atlantis Press, 2018, pp.206–210.
34. Askarzadeh A. A memory-based genetic algorithm for optimization of power generation in a microgrid. *IEEE Trans Sustain Energy* 2017; 9: 1081–1089.
35. Suhane P, Rangnekar S, Mittal A, et al. Sizing and performance analysis of standalone wind-photovoltaic based hybrid energy system using ant colony optimisation. *IET Renew Power Gen* 2016; 10: 964–972.
36. Roy R and Jadhav HT. Optimal power flow solution of power system incorporating stochastic wind power using Gbest guided artificial bee colony algorithm. *Int J Electr Power Energy Syst* 2015; 64: 562–578.
37. Kesharwani R, Sun Z and Dagli C. Biofuel supply chain optimal design considering economic, environmental, and societal aspects towards sustainability. *Int J Energy Res* 2018; 42: 2169–2198.
38. Islam MM, Zhong X, Xiong H, et al. Optimal scheduling of manufacturing and onsite generation systems in over-generation mitigation oriented electricity demand response program. *Comput Ind Eng* 2018; 115: 381–388.
39. Islam MM and Sun Z. Onsite generation system sizing for manufacturing plant considering renewable sources

towards sustainability. *Sustain Energy Technol Assess* 2019; 32: 1–8.

40. Sun Z, Li L, Bego A, et al. Customer-side electricity load management for sustainable manufacturing systems utilizing combined heat and power generation system. *Int J Prod Econ* 2015; 165: 112–119.
41. Sun Z, Li L and Dababneh F. Plant-level electricity demand response for combined manufacturing system and heating, venting, and air-conditioning (HVAC) system. *J Clean Prod* 2016; 135: 1650–1657.
42. Sun Z, Dababneh F and Li L. Joint energy, maintenance, and throughput modeling for sustainable manufacturing systems. *IEEE Trans Syst Man Cybern Syst* 2018; 50: 2101–2112.
43. Kennedy J. Particle swarm optimization. In: *Encyclopedia of machine learning*. New York: Springer, 2010, pp.760–766.
44. Shi Y. Particle swarm optimization: developments, applications and resources. In: *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*. Piscataway: IEEE, 2001, Vol. 1, pp.81–86.
45. Coello CA, Pulido GT and Lechuga MS. Handling multiple objectives with particle swarm optimization. *IEEE Trans Evolut Comput* 2004; 8: 256–279.
46. Chae YT, Horesh R, Hwang Y, et al. Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings. *Energy Build* 2016; 111: 184–194.
47. Anbazhagan S and Kumarappan N. Day-ahead deregulated electricity market price forecasting using recurrent neural network. *IEEE Syst J* 2012; 7: 866–872.
48. Quan H, Srinivasan D and Khosravi A. Uncertainty handling using neural network-based prediction intervals for electrical load forecasting. *Energy* 2014; 73: 916–925.
49. Khosravi A, Nahavandi S, Creighton D, et al. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Trans Neural Netw* 2011; 22: 1341–1356.
50. Khanesar MA, Teshnehlab M and Shoorehdeli MA. A novel binary particle swarm optimization. In: *2007 Mediterranean conference on control & automation*. Piscataway: IEEE, 2007, pp.1–6.
51. Agarap AF. Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv*:1803.08375, 22 March 2018.
52. Zhang Y, Islam MM, Sun Z, et al. Optimal sizing and planning of onsite generation system for manufacturing in critical peaking pricing demand response program. *Int J Prod Econ* 2018; 206: 261–267.
53. Denholm P, O'Connell M, Brinkman G, et al. *Overgeneration from solar energy in California. A field guide to the duck chart*. Golden: National Renewable Energy Lab, 2015.
54. Md MI, Sun Z and Dagli C. Reward/penalty design in demand response for mitigating overgeneration considering the benefits from both manufacturers and utility company. *Procedia Comput Sci* 2017; 114: 425–432.
55. Fallahi Z and Smith AD. Economic and emission-saving benefits of utilizing demand response and distributed renewables in microgrids. *Electr J* 2017; 30: 42–49.
56. Jin M, Feng W, Marnay C, et al. Microgrid to enable optimal distributed energy retail and end-user demand response. *Appl Energy* 2018; 210: 1321–1335.
57. C2ES – The Center for Climate and Energy Solutions. *Pathways to 2050: scenarios for decarbonizing the U.S. economy*. Arlington: Center for Climate and Energy Solutions, 1 May 2019.
58. U.S. Energy Information Administration. *How much carbon dioxide is produced per kilowatthour of U.S. electricity generation?* Washington, DC: U.S. Energy Information Administration, 20 February 2020.
59. Sims RE, Rogner HH and Gregory K. Carbon emission and mitigation cost comparisons between fossil fuel, nuclear and renewable energy resources for electricity generation. *Energy Policy* 2003; 31: 1315–1326.
60. Gers FA, Schmidhuber J and Cummins F. Learning to forget: continual prediction with LSTM. *Proc 9th Int Conf Artif Neural Netw (ICANN)* 1999; 2: 850–855.
61. Kong W, Dong ZY, Jia Y, et al. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans Smart Grid* 2017; 10: 841–851.
62. Duan Y, Lv Y and Wang FY. Travel time prediction with LSTM neural network. In: *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*. Piscataway: IEEE, 2016, pp.1053–1058.
63. Altché F and de La Fortelle A. *An LSTM network for highway trajectory prediction. In: 2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*. Piscataway: IEEE, 2017, pp.353–359.

Appendix

Notation

Symbol indexes	Description
j	Index of the machines in each stage
s	Index of the stages in the production line
t	Index of discretized intervals in the planning horizon

Variables

BA_{st}	Work-in-process in buffer location s at the beginning of interval t
IST	Index of the interval denoted the starting time of the production
PP	Production throughput
p_{sjt}	Indicator denotes whether machine j in stage s starts to process a new part or not at interval t (it takes the value of one if yes, and zero otherwise)
TP_{sjt}	Total number of the parts produced by machine j in stage s up to interval t

x_{sjt} State of machine j in stage s at interval t (one denotes "on" state and zero denotes "off" state)

Parameters

BA_s^{max} Maximum capacity of buffer location s

H Total hours in the planning horizon

m_s Number of parallel machines in stage s

$PC^{\max}$ Maximum power consumed over the planning horizon

PC_{sj} Rated power of machine j in stage s

PT_{sj} Processing time of machine j in stage s (unit: number of intervals)

S Total number of stages

T Total number of intervals in the planning horizon

TA Target product