An energy-aware multiobjective ant colony algorithm to minimize total completion time and energy cost on a single-machine preemptive scheduling

Saeed S. Rubaiee [a], Mehmet B. Yildirim [b]

[a] *Department of Industrial Engineering, University of Jeddah, P.O.Box 80327, Jeddah 21589, Saudi Arabia*

[b] *Department of Industrial and Manufacturing Engineering, Wichita State University, 1845 Fairmount, Wichita, KS 67260, USA*

e-mail: [a] salrubaiee@uj.edu.sa, [b] bayram.yildirim@wichita.edu

*Corresponding author.

Saeed S. Rubaiee, Ph. D.
Assistant Professor
Department of Industrial and Manufacturing Engineering
University of Jeddah
(+966) 503970707 (Mobile)
salrubaiee@uj.edu.sa
http://salrubaiee.uj.edu.sa/Default.aspx?Site_ID=0013864&lng=EN

An energy-aware multiobjective ant colony algorithm to minimize total completion time and energy cost on a single-machine preemptive scheduling

ARTICLE INFO

Article history:

Keywords:
Single-machine preemptive scheduling
Energy-aware scheduling
Time-of-use electricity tariffs
Multiobjective optimization
Internet of Things (IoT)
Ant Colony Optimization algorithm
Green and sustainable manufacturing

ABSTRACT

Energy-aware scheduling in manufacturing operations with time-of-use electricity tariffs is a challenging problem. Sustainable manufacturing is gaining significant momentum: companies are not only improving their product quality, but also optimizing the production processes to improve energy consumption (i.e., minimizing energy cost) in order to manage environmental challenges which contribute to global climate change. The purpose of this paper is to study a preemptive scheduling problem on a single-machine to minimize the total completion time and total energy cost under time-of-use electricity tariffs, which is a mixed-integer multiobjective mathematical programming model. To solve these objectives, we develop several new holistic ant colony optimization algorithms. The proposed model is solved via several methods including weighted sum method (WSM) using CPLEX, and multiobjective ant colony optimization based on a dominance ranking (ACO-DR) or based on a dominance ranking procedure and crowding distance comparison (ACO-DRC) or based on a heuristic approach to obtain an approximate Pareto-front and also provide information on when to start and resume each job for any solution on the Pareto-front. We provide detailed experimental results evaluating the performance of the proposed algorithms. In a case study, we demonstrate how the results of the multiobjective model could be utilized in decision making using the multiobjective optimization on the basis of ratio analysis (MOORA) method. This proposed model and heuristics allows decision makers to operate in challenging-data enabled environments in industrial internet of things ecosystem, and assists in optimizing production planning to improve energy cost.

1. Introduction

Manufacturers put significant effort to investment in sustainability to minimize their environmental impact and corporate social responsibility which is simply a way for manufacturers to take responsibility of their operations management. These efforts may enforce a manufacturing environment to think differently on how to manage the operations, identify opportunities, and perform better. In an environment where utility companies apply dynamic electricity pricing (i.e., time-of-use tariffs) to manage generation capacity more efficiently by shifting demand during high price periods to avoid additional investments, a manufacturing environment could utilize smart planning and scheduling practices that takes into account the price of electricity and scheduling goals at the same time. For a manufacturing, the changes in capacity utilization over different periods or machines generate differences in the opportunity cost (i.e., energy cost) of the operations over the planning horizon.

Integrating the electricity pricing data into operations plan has become feasible to many manufacturers with the advances done in industrial internet of things. Performing a smart production plan that minimizes operations during high electricity pricing periods may not only help the manufacturing cost but also be an opportunity to corporate social responsibility by helping utility company to potentially reduce their environmental impact by not requiring additional infrastructure investment to satisfy the demand. In fact, committing to socially responsible initiatives is truly a win-win situation. Not only will manufacturers appeal to socially conscious consumers and employees, but they will also make a real difference in the sustainable and green businesses. This paper proposes a sustainable manufacturing practice to achieve energy cost savings while considering a preemptive scheduling problem (i.e., total completion time) on a single-machine. The proposed model is solved via several methods. The proposed framework can be applied not only in manufacturing environment but also in any other environment where energy savings can be recognized by simply incorporating time-of-use tariffs via scheduling problem.

Energy is the lifeblood of the manufacturing economy and modern society. Energy efficiency, advanced energy technologies, internet of things (IoT) and their associated products and services have been among the most rapidly growing projects for investment in recent years, with many manufacturers becoming ownership investing rather than simply technology transfer followers. Stable, reasonably priced energy supplies are central to maintaining and improving the financial of many manufacturing sectors.

IoT is a system of connected computing devices, mechanical and digital machines, etc., and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. For example, integration of smart metering at machine level, sensors and computer network systems serving as the foundation of IoT has enabled operation scheduler to schedule in real time with respect to dynamic parameters such as pricing of electricity (i.e., time-of-use tariffs), deadlines, priorities, etc. Interested readers can also refer to Da et al. (2014) on IoT.

Manufacturing has been historically one of the biggest energy consumers of fossil and renewable energy including coal, oil, nuclear, wind and hydropower and currently accounts for about one-half of the world's total energy consumption (Annual Energy Outlook, 2013). In addition, manufacturing is responsible for 90% of industry energy consumption and 84% of greenhouse gas emissions (Schipper, 2006). Moreover, manufacturing energy consumption, which was at 175 quadrillion Btu in 2006, is projected to increase 40% by 2030 (Fang et al., 2011). Indeed, the consumption of energy by the manufacturing sector has virtually doubled over the last 60 years (Fang et al., 2011). In the United States, manufacturing sector is the largest energy consumer and consumes about one-half of the world's total energy consumption. Moreover, the manufacturing sector contributes 27% of the U.S. greenhouse gas emissions. Thus, the manufacturing sector becomes the second after the transportation sector in terms of greenhouse gas emissions.

Manufacturing activities play a major role in industrial energy consumption (Lei and Feng, 2012). Despite the fact that the machining performs a small portion of the whole product life cycle, reducing the energy consumed during this state is recently determined as the significant tasks to improve sustainability in manufacturing. For example, Chen et al. (2013) note that the painting processes at automotive assembly plants consumed about 16% of the $700 million energy expenditures in 37 U.S. automobile manufacturing plants. While all the energy consumed and waste produced by manufacturing affect both economic and environmental impacts, it is important for decision makers to work on utilizing energy efficiently and effectively because it is less expensive to save energy than to produce that same amount of energy (Annual Energy Outlook, 2013). As a result, energy efficient scheduling via intelligent operational methods may provide an immediate opportunity to decrease energy intensity of a manufacturing process. For example, utilizing the time-of-use electricity tariffs during production scheduling will make detailed manufacturing scheduling on the energy consumption and associate cost.

Giving increasing energy prices over the last ten years and efforts for sustainability, the energy savings would be more significant nowadays especially if we are focusing efforts on operations instead of focusing solely on process efficiency by developing and designing more energy efficient machines to reduce the power and energy demands of machines and tools (Salido et al., 2013). One of the strategies, in this paper, is we focus our attention on a single-machine preemptive scheduling problem, where a machine can respond to insert machine "off/on" times over planning horizon and in a time-of-use electricity tariffs, where energy prices vary hourly over planning horizon, which is typically announced a day ahead. Therefore, this strategy integrates different control policies: the first control request switches the machine "off" or "idle" (i.e., low energy level) wherein inserting machine off time is occurring. The second request switches the machine "on" wherein the job (or part of it) is resuming.

Limiting energy resources and increasing number of manufacturing plants lead to higher energy prices. In the long term, this trend is expected to continue as a result of growing demand around the world. Therefore, it is very important to utilize energy resources in such a way that the total energy cost is minimized. Rubaiee et al. (2018) develop a mixed integer multiobjective mathematical model to minimize the total tardiness and total energy cost under time-of-use electricity tariffs. Zhang et al. (2018) address bi-objective single-machine batch scheduling under time-of-use electricity prices to minimize the total energy cost and the makespan. Che et al. (2016) addresses an energy-conscious single machine scheduling problem under time-of-use or time-dependent electricity tariffs, in which electricity prices may vary from hour to hour throughout a day. In another research, Che et al. (2017) investigates an energy-conscious unrelated parallel machine scheduling problem under time-of-use electricity pricing scheme. Lee et al. (2017) propose a dynamic control algorithm to enable an energy-aware single machine scheduling under the time-varying electricity pricing policy, in which price rates remain fixed day-to-day over the season. Ding et al. (2016) develop a mathematical model under a time-of-use electricity prices scheme to minimize the total electricity cost by appropriately scheduling the jobs such that the overall completion time does not exceed a predetermined production deadline. Sharma et al. (2015) present a new 'econological scheduling' model combining the economic and ecological aspects of a multi-part multi-machine setup operating under a time-of-use tariff. Fang et al. (2014) consider the problem of processing jobs under time-of-use electricity tariffs. They assume that with time-of-use tariffs, energy prices vary hourly, which is typically announced a day ahead or an hour ahead of their production processes (i.e., price structures are used to shift electricity use from on-peak hours to off-peak hours). Zhang et al. (2014) develop a mathematical model to minimize the electricity cost and the carbon footprint under time-of-use tariffs without compromising production throughput. In further research, Zhang et al. (2015) use a distributed optimization approach to minimize the total electricity cost as a function of the manufacturing schedule, where energy efficient scheduling is subject to real-time electricity pricing. In results of the aforementioned studies, they provide a model which could be used in a time-varying tariff energy costs to decrease electricity bills. But, none of them considers the preemptive job scheduling with multiobjective functions such as optimizing energy cost and scheduling objectives.

Key features of this study are: (1) the opportunity for sectors to have access to time-of-use electricity tariffs, where energy prices vary hourly, (2) sectors that see and respond to energy prices can reduce the demand when prices are high and increase the demand when prices are low, and (3) equipment that processes jobs with allowing preemptions such as industrial battery charging stations or computer data processing jobs. The first feature can be applied via smart metering technology whereas the second feature can be applied via production scheduling. Consequently, manufacturing sectors would pay less for electricity consumption over the long run.

In this paper, we aim on a framework for sustainable manufacturing practice achieving energy consumption savings without any major equipment investment. The proposed framework focuses on manufacturing operations management which could provide significant savings in addition to reductions which may be gained through designing more energy efficient machines with less power demand and energy consumption. In this paper, the goal is minimization of a scheduling objective, total completion time, and the total energy cost in a single-machine preemptive scheduling setting under time-of-use tariffs, i.e., real-time electricity pricing over planning horizon. This problem is modeled as a multiobjective mixed-integer optimization problem (MMIP) by which when solved one can obtain a set of non-dominated solutions (Pareto front). The methods utilized to solve this problem are 1) weighted sum method (WSM) to obtain an exact Pareto front using CPLEX solver and 2), multiobjective Ant Colony Optimization (ACO) based on a dominance ranking (ACO-DR), 3) based on a dominance ranking procedure and crowding distance comparison (ACO-DRC) 3) and based on a heuristic approach to obtain an approximate Pareto-front. Moreover, the exact method Pareto front is used as a benchmark to evaluate the performance of proposed ACO algorithms which generate the approximate Pareto front. Finally, we use the multiobjective optimization on the basis of ratio analysis (MOORA) method to assist decision makers choose the most appropriate solution based on criteria for implementation.

Note that there are several multi-criteria decision making (MCDM) or multi-criteria decision analysis (MCDA) methods that can be used on selecting the best or preferred alternative(s) from a finite of alternatives. For example, decision maker (i.e., production planner) can use the Analytic Hierarchy Process (AHP) method or Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) method. Moreover, interested reader can refer to Belton et al. (2012) on MCDA or MCDM methods.

Some research efforts focused on the problem of scheduling start-up and shutdown of machines to minimize total energy consumption. Drake et al. (2006) show that machine start-up and machine idling consume a significant amount of energy. In a mass production environment, more than 85% of the energy is consumed for activities that are not directly related to the production processes (Gutowski et al., 2005). For example, compressors in an industrial setting consume about 50% of the maximum power when they are idle (Yildirim and Mouzon, 2012). In a research study, Prabhu et al. (2012) propose an analytical model based on queuing theory to include energy control for waste reduction and calculate the time interval for switching the machine off during idle period with respect to energy savings. Chang et al. (2013) provide a systematic method to search for energy saving opportunities and strategies in a machining line under random failures.

In this paper, the main contribution is to consider the total energy cost objective and scheduling objective on a single-machine preemptive scheduling. We assume that, the single-machine preemptive scheduling is able to respond to inserting machine on/off time while scheduling is designed. Furthermore, when the mathematical model is developed, methods to solve the model need to be introduced in order to find solutions belonging to the global optimal Pareto front. Therefore, we propose a mixed-integer multiobjective mathematical model to minimize the total completion time and total energy cost by inserting machine on/off time on a single-machine setting, where preemptions are allowed, (i.e., the processing of a job can be interrupted at any time and restarted at a later time in favor of another job with no cost).

Extensive research has been devoted to the single-machine preemptive scheduling problem. However, little attention has been paid to the problem of inserting machine on/off time on a single-machine preemptive scheduling setting where a multiobjective mathematical model is used. Such problems arise in network routing and battery-operated system such as laptops or mobile phone. For example, Swaminathan and Chakrabarty (2003) present a control system to reduce energy consumption and extend battery liftime and show that by changing the state of the machines (e.g., on/off, etc.), there can be a significant reduction in energy consumption. Inserting machine on/off time occurs when a machine is kept waiting (i.e. the machine is turned off for a predetermined amount of time) for the arrival of a job (or a part of it) that is whether or not being processed on a machine. This may be desirable in many situations such as during the delay of a job (or a part of it) when there are significant amounts of energy costs for early completion (i.e. there can be a significant amount of energy savings when idle machines are turned off for a certain amount of time). Kanet and Sridharan (2000) define inserting idle time as a feasible schedule in which a machine is kept idle at a time when it could begin processing another operation. In the context of production scheduling, they provide a taxonomy of environments in which inserting idle time scheduling is relevant.

A scheduling problem that is widely studied in the literature is the minimization of total completion time. Completion time of a job is the time when a machine has finished processing a job. When all release dates are equal (i.e. all the jobs are available at the beginning, $r_j = 0$) and the processing time is not interrupted (i.e. preemptions are not allowed), the shortest processing time (SPT) rule minimizes the total completion time (Pinedo, 2005). When one of these assumptions is omitted (i.e. processing time is interrupted due to inserting machine on/off time), the total completion time problem is NP-hard (Akturk et al., 2003), i.e. no known algorithm can solve this problem in polynomial time.

To solve difficult optimization problems, metaheuristic approaches are widely utilized. Ant Colony Optimization (ACO) algorithms are used to solve combinatorial optimization problems including multiobjective optimization problems. The basic mechanism of the ACO algorithm is that a colony of artificial ants cooperates in finding good solutions to combinatorial optimization problems. Moreover, interested reader can also refer to Dorigo and Stützle (2003) for more knowledge in ACO algorithm.

While optimizing the production processes to improve energy consumption has been studied by many researchers under various situations and conditions, there is still need for further research that focuses on assigning jobs with priorities (i.e., at times it is necessary to process job that has higher priority before another job although it is processing). However, to our knowledge, none of the previous research simultaneously considers both minimizing total completion time and total energy cost by inserting machine on/off time on a single-machine preemptive scheduling setting. In summary, our contributions are as follows: (1) proposing smart planning and scheduling practices that takes into account the price of electricity (i.e., time-of-use tariffs) and scheduling goals (i.e., minimizing the total completion time) at the same time, (2) presenting an energy-aware scheduling model, which is a mixed-integer multiobjective mathematical model to minimize the total completion time and total energy cost by inserting machine on/off time on a single-machine setting, where preemptions are allowed, (3) developing three multiobjective ACO algorithms to obtain a near-optimal Pareto front in a timely fashion, (4) considering sustainability practices to balance the energy demands and generating capacity (i.e., reducing the demand when energy prices are high and increase the demand when energy prices are low), and (5) enabling the decision makers to operate in challenging-data enabled environments in industrial internet of things ecosystem, and assists in optimizing production planning to improve energy cost.

The organization of this paper is as follows: In Sections 2 and 3, the multiobjective model is defined, and a weighted sum method is utilized to solve the model and evaluate the performance of the heuristic. In Section 4, two ACO algorithms are proposed to solve the multiobjective problem. Section 5 discusses experimental design and evaluation. Section 6 presents experimental design, generation of the experimental data and some performance measures for evaluating the algorithm. Section 7 compares the results of ACO algorithm with and without incorporating a shortest remaining process time (SRPT). Section 8 discusses the results of the applied algorithms and weighted sum method. In Section 9, a case study is developed to illustrate the optimized ACO-DRC algorithm. Finally, in section 10, a conclusion and future work of the research are presented.

2. A multiobjective mixed-integer programming model to minimize energy cost and total completion time (MMIP-MEC-TCT)

The MMIP-MEC-TCT problem is a single-machine scheduling problem with known deterministic processing times (p_j) and all jobs are available at the same time (i.e. $r_j = 0$). The electricity pricing, which is typically announced a day ahead, varies hourly. A machine can process one job at a time and once the process of a job is started, it can be interrupted and restarted at a later time, i.e., jobs are preemptive. The objectives are to minimize the total completion time and total energy cost. The MMIP-MEC-TCT is formulated as a mixed-integer programming (MIP) problem.

To model MMIP-MEC-TCT, we discretize the planning horizon into the intervals of one-unit hour length.

2.1 Notation

The notation used in the problem statement and through the paper is as follows:

Sets

N	Jobs, $\{1, \dots, n\}$
T	Planning horizon, $\{1, \dots, t\}$
T'	Subset of planning horizon, $T' \subseteq T$

Indices

j, i	Jobs, $j, i \in N$
t, t'	Time interval, $t \in T$ and $t' \in T'$

Parameters

E_t	Electric price signal at t
p_j	Processing time of job j

Variables

C_j	Completion time of job j
x_{jt}	$= \begin{cases} 1, & \text{if the job (or part of job) is assigned to time unit } t \\ 0, & otherwise \end{cases}$

Note that the jobs are labeled in such a way that if $p_i \le p_j$ then $i \le j$.

The proposed mixed-integer programming model MMIP-MEC-TCT is a mathematical program with multiple objectives and several constraints.

$$Minimize \sum_{j=1}^{n} C_j \tag{1}$$

$$Minimize \sum_{j=1}^{n} \sum_{t=1}^{|T|} E_t x_{jt} \tag{2}$$

$$\sum_{t=1}^{|T|} x_{jt} = p_j \qquad \forall j \in J \tag{3}$$

$$\sum_{j=1}^{n} x_{jt} \le 1 \qquad \forall t \in T \tag{4}$$

$$C_j \ge t x_{jt} + 1 \qquad \forall j \in N;\ t \in T \tag{5}$$

$$C_j \le |T| \qquad \forall j \in N \tag{6}$$

$$C_j \ge 0 \qquad \forall j \in N \tag{7}$$

$$x_{jt} \in \{0,1\} \qquad \forall j \in N;\ t \in T \tag{8}$$

The first objective (1) minimizes the total completion time and the second objective (2) minimizes energy cost. The satisfaction constraint set (3) states that the whole processing time of every job is satisfied; the capacity constraint (4) ensures that the machine can handle at most one job during each time interval. Constraint (5) provides the completion time of each job. Constraint (6) defines the boundary of completion time for each job over time horizon $|T|$. Further, Constraint (7) is the non-negativity and set (4.8) is the integrality constraint.

MMIP-MEC-TCT has $|N| + |N||T|$ variables and $2|N| + |T| + |N||T|$ constraints for a problem of size $|T|$ and $|N|$ where $|T| > \sum_j^n p_j$. Obviously, if either $|T|$ or the number of jobs N increases, the size of the problem also increases significantly. Note that these constraints could also be written in terms of the start-time variables.

This multiple objective scheduling and pricing problem can be solved by using multiple objective programming solution techniques such as weighted sum method, ε-constraints method, and multi-level programming (Caramia and Dell'Olmo, 2008) or by using metaheuristic methods such as genetic algorithm, and tabu search (Jaszkiewicz, 2001). In the next section of this paper, we will present the weighted sum method to obtain an exact Pareto front to the multiobjective mathematical model. Next we will propose an efficient multiobjective ACO algorithms to solve larger sized problems in a reasonable amount of time.

3. The weighted sum method to solve MMIP-MEC-TCT problem

The weighted sum method to solve multiobjective optimization problems converts the multiobjective problem into an aggregated scalar objective function (WS-MMIP-MEC-TCT) by multiplying each objective function by a weighting factor and summing up all terms as follows:

$$\min_{x \in X} \sum_{k=1}^{K} w_k f_k(x) \tag{9}$$

where k is the number of criteria functions, x is decision vector, X is the parameter space, and w_k is weights. Without loss of generality, the solution for the above optimization problem defines a solution in the set of the Pareto front. Moreover, interested reader can also refer to Kim and De Weck (2005) for more knowledge in WSM for multiobjective optimization.

To determine an approximate Pareto front for large-sized problems in reasonable amount of time, in next section, we propose multiobjective ant colony optimization, a multiobjective metaheuristic approach.

4. Solution to MMIP-MEC-TCT model using a multiobjective ACO-DRC algorithm

In this section, we propose two ACO algorithms, based on Swarm Intelligence methods, for solving multiobjective optimization problem to obtain an approximate Pareto front. Generally, ACO algorithm was inspired by some principles of the behavior simulation of social insects such as ants. In our implementation, the ACO-DR is based on a dominance ranking and the ACO-DRC is based on a dominance ranking procedure and crowding distance comparison. These types of algorithms are used for generating multiple solutions in a single run. In those ACO algorithms, they determine an approximate Pareto front for a large-size problem in a reasonable amount of time, and allow flexibility when rescheduling is needed due to disruptions on the shop floor or changes in the manufacturing environment. At each iteration, the ACO algorithms provide an approximate Pareto solution that has information on when to start each job (or part of it) to calculate the scheduling and energy costs for that particular solution. The structure of the proposed ACO algorithms are based on the ant colony optimization for the MMIP-MEC-TCT scheduling problem given in Fig. 1.

1. **Initialization:** the pheromone trails, the heuristic information and the parameters are initialized
2. **Iterative loop:**
   2.1 A Colony of ants determines starting (on) time and off time over time horizon.
   2.2 Construct a complete tour for each ant:
   Repeat
   Apply sate transition rule to select the next node
   Apply the local updating rule
   Obtain approximate Pareto front of a tour using reduced WS-MMIP-MEC-TCT
   Evaluate the fitness value of the tours
   Until a complete solution is constructed
   2.3 Apply the global update rule
   2.4 Perform selection operation
   2.5 Determining the a Pareto front
3. **Cycle:** if the maximum number of iterations is realized, then stop; else go to step 2.

Fig. 1. The structure of MMIP-MEC-TCT-MOACOA.

Note that ACO-DR and ACO-DRC are using the same solution components, but they differ in the evaluation of solution quality (i.e., fitness function and fitness value) and selection process. In the next subsections, we discuss different components of the algorithms in more details.

4.1 Construction of graph

The ACO algorithms for MMIP-MEC-TCT model can be represented by a complete graph $G = (N, A)$ with N being the set of nodes representing the times (e.g. $N =|T|$), and A being the set of arcs. Each arc $(\mathfrak{a}, \mathfrak{b}) \in A$ is assigned a value (i.e., it can be an actual time moment a price signal) $d_{\mathfrak{a},\mathfrak{b}}$, which represents the value at node $\mathfrak{b}$. The total number of arcs is $(N(N-1))$, (see Fig. 2). On this graph, the goal of an artificial ant is to visit $n = \sum_j^n p_j$ nodes of G exactly one, where each artificial ant is initially put on a random chosen start-time (node) and each step iteratively assigns zero to unvisited nodes and 1 to visited nodes of its partial tour. The unvisited nodes are considered to be inserting machine off time, where the machine is turned off for a certain amount of time. For instance, with $T = 5$ and $\sum_j^n p_j = 3$, the graph has five sets of nodes ($N = 5$) and twenty possibilities for the arcs. Each artificial ant can visit only three out of five nodes. When an artificial ant completes a tour, the visiting node yields the time moment that a job can be assigned to. Furthermore, the visited nodes take value 1 and unvisited nodes take value 0 (e.g. $[10110]$). Value 1 indicates where a job can be executed and value 0 indicates where the inserting machine off time can be allocated. One tour shows the assignment of execution times and inserted machine off time over the planning horizon.

4.2 Constraints

Walks on the construction graph G have to satisfy the constraints given by equations (3) and (4) to obtain a valid assignment. One way of generating such an assignment is by an artificial ant's walk which iteratively switches from one node to another node without repeating any node to find routes that correspond to better solutions.

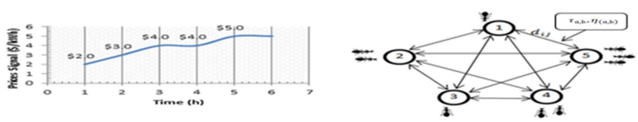

Fig. 2. Construction graph for ACO algorithms of size N = 5.

4.3 Pheromone trail

During the construction of a tour, ants repeatedly have to take the following two basic decisions: (1) choose the node to assign next job based on actual time moment (i.e., node with lowest time indices) in order to minimize the completion time and (2) choose the lower energy cost node moment (i.e., node with lowest electricity price indices) in order to minimize the energy cost. Pheromone trail information can be associated with any of the two decisions: an appropriate sequence for node assignments in order to minimize the total completion time or to minimize the total energy cost. As a result, a pheromone level $\tau_{\mathfrak{a},\mathfrak{b}}$ will represent the desirability of assigning node $\mathfrak{b}$ to node $\mathfrak{a}$. Note that in our ACO algorithms, the initial pheromone level is calculated by $\tau_0 = \frac{1}{\Theta_{\iota\xi}}$, where $\Theta_{\iota\xi}$ is the fitness function value for a solution ι in tour ξ.

4.4 Heuristic information

Similarly, heuristic information can be associated with any of the two decisions, on total completion time or total energy cost. For example, heuristic information could bias node assignment towards those nodes that have small time—in other words, this will minimize completion time, and bias the choice of nodes in such a way that small assignment costs are incurred—in other words, this will minimize energy costs (i.e., the machine only needs a relatively small amount of cost to perform the job). In our ACO algorithms and while constructing the schedule, we apply both decisions in random iterations in one run in order to generate different solutions.

4.5 Solution construction

Construction of a solution can be performed by choosing the components to add to the partial solution from among those that, as explained above, satisfy the constraint with a probability biased by the pheromone trails and heuristic information.

In each iteration, an artificial ant determines starting "on" time and "off" time for all schedules. Each artificial ant repeatedly applies the state transition rule to select one of the decisions from the heuristic information until a complete tour is constructed. When building a tour, both the heuristic information and the pheromone amount are used to determine the on time and off time (nodes) to be chosen. Section (4.6) will explain in detail how the state transition rule can be applied.

While constructing a tour, an artificial ant also decreases the amount of pheromones between selected nodes in order to dynamically change the attractiveness of arcs by applying the local updating rule to vary other ants' visits (schedule) and to avoid leading to local optima. The local updating rule will be explained in Section (4.7).

The tour represents a feasible schedule of the on/off times for the machine and has a fixed energy cost. However, the tour does not provide any information on which job is actually scheduled (i.e., the sequence of jobs/parts of jobs). Thus, in a given tour, one has information on the on/off times for the machine which also implies a fixed energy cost. One needs to determine the sequence of each job in order to have information to evaluate each objective function. In Section (4.8), the process of evaluating each objective function will be addressed.

In each iteration, the fitness function is used to evaluate the quality of the solutions in the current population. The selection operator is capable of preserving those tour that have better fitness values and a better chance of leading to local optima. Based on the best value of the fitness function, the global updating rule is applied to increase pheromone values on solution components between nodes of the best solution up to the current iteration and decrease pheromone values between other nodes. Thus, all ants will focus on a better tour (schedule). Sections (4.9) shows fitness function and value in the proposed algorithms. Sections (4.10) and (4.11) discuss the process of selection. And, Section (4.12) shows how the fitness function value (best value) serves in the global updating rule. Finally, Section (4.13) presents the stopping criterion in the algorithm.

For any artificial ant γ, while finding appropriate solution for the problem, the set of candidate nodes that may be visited in the reminder of the tour are defined as a tabu list. In an artificial ant's tabu, those nodes that have been visited (e.g. the nodes that have been assigned already based on heuristic information) are excluded from the choice through the use of a tabu list. This is applied by the state transition rule. Until the last node is selected, the procedure is repeated. Once the algorithm reaches its stopping criterion, the obtained solutions have a well-distributed near-optimal Pareto front, which will then be used in the selection of the best appropriate solution using the MOORA method.

4.6 State transition rule

In the process of choosing the next node to move, the artificial ant γ in node $\mathfrak{b}$ selects the node $\mathfrak{a}$ to move by applying the following state transition rule:

$$\mathfrak{b} = \begin{cases} \arg\max_{u \in S_k(\mathfrak{a})} \{[\tau_{(\mathfrak{a},u)}]^{\alpha}[\eta_{(\mathfrak{a},u)}]^{\beta}\} & if\ q \leq q_0 \\ \Re & Otherwise \end{cases} \quad (10)$$

where, $\tau_{(\mathfrak{a},u)}$ is the amount of pheromone trail on arc $(\mathfrak{a},u)$, $\eta_{(\mathfrak{a},u)} = 1/d_{\mathfrak{a},u}$ is the inverse of the heuristic information $(d_{\mathfrak{a},u})$ between node $\mathfrak{a}$ and node u. $S_k(\mathfrak{a})$ is the set of feasible nodes to be selected by artificial ant γ in node $\mathfrak{a}$. Note that the set of feasible nodes not contained in tabu α is a parameter that allows us to control the relative importance of the pheromone trail $(\alpha > 0)$; β is a parameter that determines the relative importance of heuristic information $(\beta > 0)$. q is a value chosen randomly with uniform probability in $[0, 1]$; and q_0 is a parameter that determines the relative importance of exploitation versus exploration $(0 \leq q_0 \leq 1)$. $\Re$ is a random variable selected according to the following random-proportional rule probability distribution, which is the probability with which that artificial ant γ chooses to move from node $\mathfrak{a}$ to node $\mathfrak{b}$:

$$P_k(\mathfrak{a},\mathfrak{b}) = \begin{cases} \frac{[\tau_{(\mathfrak{a},u)}]^{\alpha}[\eta_{(\mathfrak{a},u)}]^{\beta}}{\sum_{u \in S_k(\mathfrak{a})}[\tau_{(\mathfrak{a},u)}]^{\alpha}[\eta_{(\mathfrak{a},u)}]^{\beta}} & if\ \mathfrak{b} \in S_k(\mathfrak{a}) \\ 0 & Otherwise \end{cases} \quad (11)$$

When an artificial ant in node $\mathfrak{a}$ chooses a node $\mathfrak{b}$ to move to, it generates a random number q. If $q \leq q_0$, then the best node is chosen according to the following equation:

$$\eta_{(\mathfrak{a},u)} = \frac{1}{d_{(\mathfrak{a},u)}} \qquad (\mathfrak{a} \neq u) \quad (12)$$

where, d_{iu} is the heuristic information between node i and node u.

4.7 Local updating rule

While constructing a tour, an artificial ant decreases the pheromone trail level between selected nodes in order to dynamically change the attractiveness of arcs by applying the following local updating rule:

$$\tau_{(\mathfrak{a},\mathfrak{b})} = (1 - \rho_l).\tau_{(\mathfrak{a},\mathfrak{b})} + \rho_l.\tau_0 \quad (13)$$

where, τ_0 is the initial pheromone level and ρ_l $(0 < \rho_l < 1)$ is the local pheromone evaporating parameter. In this way, every time an artificial ant selects an arc, it becomes slightly less desirable. Ants, therefore, adapt pheromone information better. Without local updating, all ants would explore a narrower neighborhood of the best previous tour. The pheromone associated with this arc is adjusted every time the artificial ant chooses a node.

4.8 Obtaining approximate Pareto front of a tour using reduced MIP

While constructing a tour, an artificial ant generates one solution. By giving a random tour, a feasible schedule of start (on) and off times for the machine are fixed. Therefore, the fixed start (on) times are a subset of time horizon (i.e., a tour represents the subset of time horizon, $T' \subseteq T$). When this information is known, the original model of MMIP-MEC-TCT reduces to the following multiobjective mixed-integer mathematical model (MMIP-MEC-TCT-R) where R means "reduced" model:

$$Minimize \sum_{j=1}^{n} C_j \quad (14)$$

$$Minimize \sum_{j=1}^{n}\sum_{t' \in T'} E_t x_{jt'} \quad (15)$$

$$\sum_{t' \in T'} x_{jt'} = p_j \qquad \forall j \in J \quad (16)$$

$$\sum_{j=1}^{n} x_{jt'} = 1 \qquad \forall t' \in T' \quad (17)$$

$$C_j \geq t' x_{jt'} + 1 \qquad \forall j \in N;\ t' \in T' \quad (18)$$

$$C_j \leq |T'| \qquad \forall j \in N \quad (19)$$

$$C_j \geq 0 \qquad \forall j \in N \quad (20)$$

$$x_{jt'} \in \{0,1\} \qquad \forall j \in N;\ t' \in T' \quad (21)$$

where the capacity constraint (17) ensures that the machine can handle only one job (or part of it) during each time interval, i.e., $t' \in T'$. MMIP-MEC-TCT-R has $|N| + |N||T'|$ variables and $2|N| + |T'| + |N||T'|$ constraints for a problem of size T' and $|N|$ where $T' = \sum_j^n p_j$ and $T' \subseteq T$. Solving the MMIP-MEC-TCT-R will provide the set of non-dominated solution(s) corresponding to a known tour (i.e. a given machine on/off time). Note that in this formulation, C_j and p_j indicate completion time and process time of job in position j in the sequence defining the orders of the jobs.

A random tour which represents a subset of time horizon has a fixed total energy cost. Below we provide a theoretical proof showing the local optimality of the total energy cost function for given a random tour:

Proposition 1: For a given a complete tour (i.e., a tour represents the subset of time horizon, $T' \subseteq T$), the total energy cost is constant.

Proof: The total energy cost function (equation 15) is:

$$\sum_j^n \sum_{t' \in T'} E_t x_{jt'} = \sum_{t' \in T'} E_t \sum_{j=1}^n x_{jt'} \quad (22)$$

The capacity constraint ensures the machine can handle only one job (or part of it) at any given time interval (i.e., $t' \in T'$):

$$\sum_{j=1}^n x_{jt'} = 1 \quad (23)$$

Therefore, from equations (22) and (23), we can prove that the total energy cost function is fixed as follows:

$$\sum_j^n \sum_{t' \in T'} E_t x_{jt'} = \sum_{t' \in T'} E_t \sum_{j=1}^n x_{jt'} = \sum_{t' \in T'} E_t \quad (24)$$

Since proposition 1 proves that for each tour the total energy cost function is fixed. Therefore, we only solve the total completion time objective function to obtain the Pareto optimal solution corresponding to an ant's tour.

Shortest processing time (SPT) or shortest remaining processing time rules (SRPT) provide optimal solutions to total completion time problem without any inserted idle time. The question is if such rules can also provide an optimal solution for the problem that we are investigating in this paper.

In SRPT rule, we schedule all jobs in the order of increasing p_j. Note that, this information may change while a job is being processed in especially preemptive job environments with unequal release dates. Thus, a job may release to process with a large processing time and low priority respectively, but after some partial processing, it will have a smaller remaining processing time and higher priority respectively. Therefore, while constructing a new sequence, at any time, we process the unfinished job with the SRPT among the available jobs at this time moment. Therefore, we sort jobs in such a way that if $p_i \leq p_j$ then $i \leq j$. Consequently, the computational complexity of our heuristic is determined by $O(nlogn)$ on each run (i.e., complexity of a sorting algorithm). Below we provide some theoretical results showing the local optimality of this rule for the considered problem with use of inserting machine on/off time.

Theorem 1: In an optimal schedule, the total completion time is minimized by scheduling all jobs with respect to shortest remaining processing times (SRPT) sequence for a MMIP-MEC-TCT-R problem with inserted on/off time and equal release time ($t = 0$).

Proof: We prove by interchanging two neighboring jobs (or job parts). Consider two consecutive jobs $(i.j)$ that are available at the same time $(r_t = 0)$, $i < j$ and $p_i \leq p_j$. Let S represent an optimal schedule that is not the SRPT sequence, then $p_j \geq p_i$ Consider the schedule S' obtained by interchanging the jobs $(i.j)$ (see Fig. 3). We can show that $\sum_{k\in S'} C_k^{S'} < \sum_{k\in S} C_k^S$, contradicting the optimality of S.

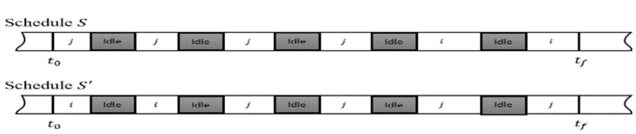

Fig. 3. A pairwise interchanging of jobs ***j*** and ***i***.

Let C_j^S and $C_j^{S'}$ denote the completion times of job j in S and S', respectively. We observe that (1) the first job starts at the same time point regardless of the order of the jobs. Furthermore, the second job finishes at the same time point regardless of the order of the jobs; and (2) idle (I) time periods are fixed over timeline and total idle period is equal to $I_{of} = t_f - t_0 - p_i - p_j$ where t_0 is the starting time of the first job and t_f is the finishing time of the second job. Therefore, we have

$$C_j^S = t_{t_0} + p_j + I_{t_0 j} \tag{25}$$

$$C_i^{S'} = t_{t_0} + p_i + I_{t_0 i} \tag{26}$$

Also, we have $p_j = p_i + \Delta$, where Δ is the difference between p_j and p_i, and $\Delta \geq 0$ since $p_j \geq p_i$. Therefore, when we move over the timeline p_i amount of time, we encounter $I_{t_0 i}$, for additional Δ we may encounter some more idle periods $\varphi \geq 0$, i.e., $I_{t_0 j} \geq I_{t_0 i}$ and $I_{t_0 j} = I_{t_0 i} + \varphi$, i.e., Now, let's compare the total completion time for each schedule which only differs in the order of jobs i and j:

$$\sum_{k\in S} C_k^S = \sum_{k\in S\setminus\{i,j\}} C_k^S + C_j^S + C_i^S = \sum_{k\in S\setminus\{i,j\}} C_k^S + t_0 + p_j + I_{t_0 j} + t_f$$

$$\sum_{k\in S'} C_k^{S'} = \sum_{k\in S'\setminus\{i,j\}} C_k^{S'} + C_i^{S'} + C_j^{S'} = \sum_{k\in S'\setminus\{i,j\}} C_k^{S'} + t_0 + p_i + I_{t_0 i} + t_f$$

Since $\sum_{k\in S\setminus\{i,j\}} C_k^S = \sum_{k\in S'\setminus\{i,j\}} C_k^{S'}$, $p_j = p_i + \Delta$, and $I_{oj} = I_{oi} + \varphi$,

$$\sum_{k\in S} C_k^S = \sum_{k\in S\setminus\{i,j\}} C_k^S + t_0 + p_i + \Delta + I_{oi} + \varphi + t_f$$

As a result. $\sum_{k\in S'} C_k^{S'} < \sum_{k\in S} C_k^S$.

To illustrate the above theorem by means of a numerical example, consider the data in Table 1. The SRPT (S') solution with inserted on/off time is also given in Fig. 4. All jobs are available to be processed at the same time moment $(t = 0)$. In Fig. 4, schedule S is not the SRPT sequence. Let's investigate job j and i. We see that job j is being processed at time $t = 6$ while there exists an unfinished job i such that the job i is also available at the same time $t = 6$, and $p_j \geq p_i$.

Now, construct a new sequence, S' by processing job i first, and processing job j second. Since $p_i \leq p_j$ is the newly scheduled job, job i completes before the time when either job j or i is completed in the old schedule S. That is $C_i^{S'} = 8$ which is less than $\min\{C_j^S = 10, C_i^S = 12\}$. Also note that $C_j^{S'} = 12$ which is equal to $\max\{C_j^S = 10, C_i^S = 12\}$. So the jobs should be scheduled in the increasing order of the processing time p_i. Thus, the total completion time decreases by justifying the SRPT rule.

Table 1
Numerical example.

n	*k*	*j*	*i*
p_n	2	3	2
w_n	3	4	2

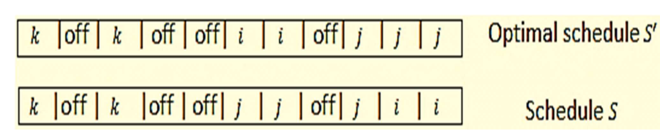

Fig. 4. Show theorem 1 in sequence of individual time slots for different jobs over 12 units of planning horizon.

4.9 Evaluation of fitness functions

Fitness functions are used to evaluate the quality of the solutions (i.e., the current ant population). Each solution has a fitness value, which is assigned according to its dominance. A solution determines the order of the jobs, the completion time of all jobs, and the energy cost of the job completed. Since a tour may characterize one or more solutions, the best measure of the fitness function over the set of solutions that a tour characterizes is kept to evaluate the tour. In the next subsections, we discuss different fitness functions and selection operators of the ACO algorithms in more details.

4.9.1 Fitness function and value for tours in ACO-DR

The fitness function in ACO-DR evaluates the solution based on its dominance rank (i.e., number of solutions by which an solution is dominated) (Fonseca and Fleming, 1993). Let $\Theta_{\iota\xi}$ be the fitness function value for a solution ι in tour ξ corresponding to an ant's tour. The rank of the solution is equal to $1 + \#_{\iota\xi}$, where $\#_{\iota\xi}$ denotes the number of solutions that dominates a solution ι in tour ξ. If the solution is non-dominated, then its rank is 1. Note that the higher the rank, the poorer the solution. This can be written as follows:

$$Fitness_1\left(\Theta_{\iota\xi}\right) = \frac{1}{rank(\Theta_{\iota\xi})} = \frac{1}{1+\#_{\iota\xi}} \tag{27}$$

The fitness function value for a tour ψ, $Fitness_\psi$, is the sum of the fitness function value for each solution obtained from a tour ψ using MMIP-MEC-TCT-R model. Note that, in order to obtain a better solution and keep the solution components consistent, the fitness function evaluates all solutions each time the MMIP-MEC-TCT-R generates a new solution in its current iteration.

4.9.2 Fitness function and value for tours in ACO-DRC

The fitness function in ACO-DRC is inspired from the Non-dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al., 2002). NSGA-II is a popular genetic algorithm for solving the multiobjective optimization. In our implementation, see Fig. 5, the solutions in level 1 (i.e., rank 1) are an exactly non-dominated set in the current population and the solution in level 2 (i.e., rank 2) is dominated by the solutions in level 1 only and the solutions in levels builds so on. Each solution in the each level is assigned fitness (rank) values or based on level (Pareto front) in which they belong. In practical, solutions in Level 1 are given a fitness value of 1 and solutions in Level 2 are assigned fitness value of 2 and so on. Note that, in Fig. 5, the level where a solution locates represents the rank, which is the most important factor of its fitness. As a result, the solution with lower rank is preferable.

In addition to fitness value a new parameter called crowding distance is computed for each solution. The crowding distance measures how close a solution is to its neighbors (see Fig. 5). In our implementation, for a solution, the crowding distance is defined as the sum of the normalized distance between its right and left neighbors for each objective function. For the first and last individuals (extreme solutions) have a crowding distance equal to infinity. In Fig. 5, the perimeter of the cuboid is estimated by using the nearest neighbor as the vertices (Deb et al., 2002). In a result, the crowding distance procedure guarantees the diversity of the population (i.e., large average crowding distance results in better diversity in the population). Finally, each individual is sorted based on rank and crowding distance value.

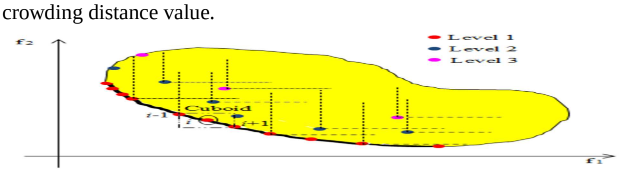

Fig. 5. Non-dominated levels and computation of the crowding distance.

4.10 Selection process in ACO-DR

The selection process aims to select the best solution among all available solutions. A tour with a better fitness function will earn more chances to be selected. We utilize the roulette wheel method, which we chose in order to select the best fitness function for the tour solutions based on rank (Pohlheim, 1998). This process is described in Fig. 6.

Step 1: Order the tours in descending fitness function value.

Step 2: Compute the probability function Pr_ψ for a tour ψ as follows:

$$Pr_\psi = \frac{Fitness_\psi}{\sum_{l=1} Fitness_l}$$

Step 3: Utilize the roulette wheel method using the probability values from step 2 until tours are selected.

Fig. 6. Determine the fitness function of each tour.

Note that, the roulette wheel method is performed sequentially according to the tabu list in order to prevent escaping tours. The tours with higher fitness function values have a higher probability to be selected every time.

4.11 Selection process in ACO-DRC

The selection is based on two comparison rules (i.e., binary tournament selection): (1) the lower rank the individual belongs to, the better the solution is; (2) if two individuals are having the same rank, the individual with greater crowding distance has a better solution because the area it belongs to is less crowded. This process can be described as shown in Fig. 7. Note that the tours with lower rank have better solutions, if they are from the same Level (i.e., Pareto front), then the tour with greater crowding distance has a better solution and survival in the next generation.

Step 1: Sort individuals based on non-domination rank $rank_\iota$ and with crowding distance assigned $Front_\#(Dis._\iota)$, where ι corresponds to the distance $Dis._\iota$ of the ι^{th}solution in $Front_\#$. Note that each solution ι belongs to one chromosome ψ.

Step 2: Utilize a *crowded-comparison-operator* $(\prec_{\#_{\iota\xi}})$, where $\#_{\iota\xi}$ is the number of solutions ι in generation ξ, based on: (1) $rank_\iota$ and (2) $Front_\#(Dis._\iota)$.

For example, $\iota_{\#4} \prec_{\#_{\iota\xi}} \iota_{\#5}$ if

- $rank_{\iota_{\#4}} < rank_{\iota_{\#5}}$
- Or if $\iota_{\#4}$ and $\iota_{\#5}$ belong to the same $Front_\#$ then $Front_\#(Dis._{\iota_{\#4}}) > Front_\#(Dis._{\iota_{\#5}})$ i.e. the crowding distance should be more.

Step 3: Utilize a binary tournament selection with *crowded-comparison-operator* from step 2 until chromosomes are selected.

Fig. 7. Process of selection of tours for generating the next ant population.

4.12 Global updating rule

After evaluating the quality of the solutions in the current iteration, we perform the global updating rule. This aims to increase the pheromone values on those solution components that have a better fitness function value. The pheromone trail level is updated as follows:

$$\tau_{(a,b)} = \left(1 - \rho_g\right).\tau_{(a,b)} + \rho_g.\Delta\tau_{(a,b)} \tag{28}$$

where, ρ_g $(0 < \rho_g < 1)$ is the global pheromone evaporating parameter and $\Delta\tau_{(a,b)}$ is computed by the following equation:

$$\Delta\tau_{(a,b)} = \begin{cases} (\Theta_{\iota\xi})^{-1} & if\ (a,b) \in best\ tour \\ 0 & otherwise \end{cases} \tag{29}$$

where, $\Theta_{\iota\xi}$ is the fitness function value of the best solution ι in tour ξ up to the current iteration. After the search, the pheromone trail level of the new solution is updated proportionally to the improvement of the fitness function value. In this way, we avoid repeating the same solution over and over again and explore a stopping criterion which limits having all ants choose the same route every time.

4.13 Stopping criteria

The ACO algorithms are run until the maximum number of iterations. In this step, the approximate Pareto front is obtained.

5. Experimental design and evaluation

We use the General Algebraic Modeling System (GAMS) to model the WS-MMIP-MEC-TCT problem, and solve the problem using CPLEX 12.5 (ILOG, 2012) to find exact Pareto front. To evaluate the performance and effectiveness of the presented genetic algorithms, we use MATLAB R2010a and utilize IBM ILOG CPLEX 12.5 via MATLAB to solve resulting optimization sub-problems. The computational experiments are performed on an Intel i5 2.27GHz machine with 4GB of memory with a Windows 7 operating system.

6. Generation of experimental data

The problems were obtained by varying combinations of three parameters: the number of jobs (n), the planning horizon (T), and the processing times (p_j). Note that, the number of jobs $|N|$ and size of planning horizon $|T|$ have a major influence on the performance of the algorithm, since the larger the size of the problem, the more the CPU time it takes. In the experimentation, problems with 5, 10, 15 and 20, jobs are generated. For each job, an integer processing time is obtained randomly from a uniform distributed between [1,10]. An integer planning horizon $|T|$ is computed as $|T| = \sum_j^n p_j / PHF$, where PHF is the planning horizon factor which takes two values: 0.50 for longer planning horizon and 0.75 for shorter planning horizon. The electricity prices E_t are generated from three uniform distributions $[1,5], [6,10], [11,16]$ over three different periods of time to create low, moderate, and high variation (see Fig. 8). Note that, the total number of energy profile is $E_{profile} = \frac{|T|!}{(\sum_{j=1}^n p_j)!(|T| - \sum_{j=1}^n p_j)!}$.

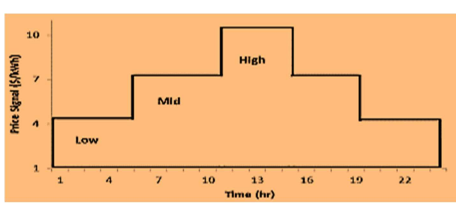

Fig. 8. Price signal ($/kWh) vs. time (hr).

6.1 Solving WS-MMIP-MEC-TCT using weighted sum method

A set of four test problems in Table 2 are randomly generated for testing and gaining insight into WS-MMIP-MEC-TCT multiobjective optimization problem. We generate 20 instances (i.e., five different instances for each test), solve every instance individually, and measure the average time in seconds.

Table 2 presents the results of weighted sum method and measure an average for each problem with the amount of execution time, and the number of non-dominated points obtained. While all generated instances for $n = 5$ are solved in less than 16,000 second, for $n = 10$ there is one instance, which weighted sum method has failed to solve to optimality within its time limit, for $n = 15$ there are already 2 timeouts, and for $n = 20$ there are 3 timeouts since CPLEX is running out of memory while solving the model. It is clear that the larger the size of the problem the more the CPU time it takes. That is, when the number of jobs increases, the weighted sum method requires a significant amount of CPU time to determine the approximate Pareto front (i.e., the non-dominated solution set), since the search space for the problem increases in terms of variables and constraints.

The CPU time can be decreased if fewer number of combinations of weights are used (i.e., we will not need as many single-objective mixed-integer problems to be solved). However, this will have significant impact on the quality of the Pareto front.

Table 2
Results of the WS-MMIP-MEC-TCT.

n	Timeouts	Execution Time (second)		Number of Non-dominated Points	
		Min	Max	Min	Max
5	0	12,243	16,000	27	31
10	1	30,004	>40,000	30	33
15	2	33,102	>40,000	33	36
20	3	37,200	>40,000	12	18

6.2 Solving WS-MMIP-MEC-TCT using ACO: parameter fine tuning

In our ACO algorithms implementation, the initial artificial ant population size in the colony is $k_{max} = 20$ and the maximum number of iteration is $t_{max} = 1000$. In order to find the best combination for $\alpha, \beta, \rho_l, \rho_g$, and q_0, and also to analyze the impact of these parameters on the computational complexity of ACO algorithms, an experimental design is performed. We apply the ACO algorithms on examples with jobs from set $n = \{10, 15, 20\}$. Initially, parameters are set as $\alpha = 1, \beta = 2, \rho_l = 0.1, \rho_g = 0.1$, and $q_0 = 0.9$. We run each test with the same parameters ten times and take the average as the measure of performance. We optimize the parameters of ACO algorithms one at a time, because considering all possible combinations of parameters is very expensive.

Therefore, we change only one parameter at a time until an initial estimate of the appropriate values is obtained while keeping all other parameters constant. The appropriate candidate set for each parameter is the parameter set that yields the appropriate overall performance of the computational complexity of our proposed algorithms.

In order to assess and compare with different parameters, a generational distance (GD) metric (Van Veldhuizen and Lamont, 1998), found in the literature, is used to measure the performance of the algorithm's solution. GD evaluates the convergence performance of the algorithm. The metric is calculated just before a change occurs. GD metric is defined as a way of estimating how "far" the elements (i.e., non-dominated solutions) are in global Pareto front from those in global Pareto front and is computed as:

$$GD = \frac{\sqrt{\sum_{l=1}^{L} d_l^2}}{D} \tag{30}$$

where D is the total number of non-dominated solutions in global Pareto front and d_l is the Euclidean distance (measure in objective space) between each of these and the nearest member of global Pareto front. Since we consider minimization objectives, the performance of an algorithm is better when the measure has smaller values. If a value of $GD = 0$, then all elements generated are in global Pareto front. Therefore, any other value indicates how "far" we are from the global Pareto front of our problem. Table 3 summarizes these sets that are obtained from test results.

While constructing the solutions, we increase the α, β and q_0 parameters from the lowest to the highest value as shown in Table 3. In each run, only one value of the parameters is changed, and all other parameters are kept constant. Test results in Table 4 show that the α, β and q_0 parameters of $1, 0.5$ and 0.9 lead to good results for the problems with a different number of jobs.

After α, β and q_0 parameters are utilized, we vary the population size from 20 to 50 in increments of 10. The results in Table 4 show that the higher the population size, the better the solution but longer the CPU time. We decided to limit the population size in the colony to $k_{max} = 20$ to have solutions in a reasonable amount of time.

Using the α, β, q_0 and population size at their appropriate value, we increase the ρ_l and ρ_g parameters from the lowest to the highest value as shown in Table 3. In each run, only ρ_l or ρ_g is varied and keep all other parameters at their optimal. Test results in Table 4 show that the ρ_l and ρ_g parameters of 0.3 for each lead to good results for the test problems.

7. Comparison of the ACO algorithm with and without using the Theorem 1

In this section, the performance and quality of the ACO algorithm with and without using the theorem 1 in the solution process are compared. After fine-tuning the parameters, we apply the ACO algorithm on the same instances used to characterize the performance of the weighted sum method. In the first approach, we use the theorem 1 (SRPT method) in the solution process to obtain the heuristic solutions whereas in the second approach, the CPLEX solver is used to minimize the total completion time objective. Note that, we run each instance twice and take the average as the measure of performance. We also record the computational time required to solve each instance and the number of solutions obtained from each approach. For comparisons, we use the GD metric, CPU time and the number of non-dominated solution to measure the performance and quality of the applied algorithms. The results are presented in Table 5.

It is clear that the ACO algorithm with theorem 1 outperforms the ACO algorithm with CPLEX with respect to all performance metrics. A possible reason for this is that the theorem 1 uses SRPT heuristic to solve the total completion time objective to optimality whereas the CPLEX uses a branch-and-cut algorithm to solve MIP model by generating many sub-problems which can require significant amount of CPU time and physical memory.

Table 3
Best candidate set.

Parameter	Candidate set
α	(0.5, 1, 2, 5)
β	(0.5, 1, 2, 5)
ρ_l	(0, 0.1, 0.3, 0.5)
ρ_g	(0, 0.1, 0.3, 0.5)
q_0	(0, 0.25, 0.75, 0.9)

Table 4
GD metric performance depending on parameters and number of jobs.

Parameter	Parameter Values	Number of Jobs 10	15	20
α	0.50	0.283	0.296	0.304
	1	0.239	0.269	0.260
	2.00	0.314	0.329	0.333
	3.00	0.399	0.365	0.354
β	0.50	0.234	0.247	0.267
	1	0.292	0.318	0.324
	2	0.341	0.347	0.363
	3	0.352	0.368	0.368
q_0	0	0.384	0.388	0.388
	0.25	0.379	0.381	0.383
	0.75	0.310	0.314	0.316
	0.90	0.247	0.251	0.251
k_{max}	20	0.416	0.419	0.423
	30	0.404	0.408	0.414
	40	0.381	0.381	0.395
	50	0.293	0.298	0.302
ρ_l	0	0.287	0.289	0.289
	0.10	0.295	0.298	0.302
	0.30	0.248	0.253	0.264
	0.50	0.296	0.314	0.320
ρ_g	0	0.285	0.288	0.307
	0.10	0.294	0.315	0.324
	0.30	0.239	0.420	0.423
	0.50	0.302	0.308	0.311

Table 5
ACO with and without Theorem 1.

n	PHF	ACO with Theorem 1: CPU (second)	Non-dominated Points	GD	ACO with CPLEX: CPU (second)	Non-dominated Points	GD
5	0.5	205	31	0.43	720	27	0.51
10	0.75	216	34	0.51	960	29	0.68
15	0.5	292	36	0.47	1260	32	0.53
20	0.75	347	37	0.38	1620	35	0.48

8. Comparison of the ACO-DR algorithm, ACO-DRC algorithm and weighted sum method

We implement the ACO-DR and ACO-DRC algorithms by using theorem 1 to solve the multiobjective mathematical model and obtain an approximate solution at a reasonable amount of computational time without any guarantee of optimality. Another remarkable feature of ACO-DR and ACO-DRC algorithms is that they may not return the same results twice, since they are based on randomized iterations that use different random tours. Taking these features into consideration, the best way to evaluate the quality of the applied algorithms is by comparing their Pareto front with the Pareto-optimal front obtained by the WSM. In order to do the comparison, we applied the ACO-DR and ACO-DRC algorithms using the fine-tune parameters on the same instances used to characterize the performance of the weighted sum method. Also, two performance metrics: efficient set spacing (ESS) (Schott, 1995), and size of the space covered (SSC) (Caramia and Dell'Olmo, 2008) are used to measure the performance and see the dynamics of the applied algorithms. ESS evaluates the diversity performance of an algorithm; while SSC can measure the size of the global dominated set in objective space.

ESS metric measures the distribution of elements in Pareto optimal set over the non-dominated region,

$$ESS = \sqrt{\frac{1}{e-1}\sum_{i=1}^{e}(\bar{d} - d_i)^2} \quad (31)$$

where $d_i = min_j\{|f_1^i - f_1^j| + |f_2^i - f_2^j|\}, j = 1, \ldots, e,$ and $\bar{d}$ refers to the mean of all d_i and e is the number of elements of the Pareto optimal set found so far. If a value of $ESS = 0$, then the algorithm provides an ideal distribution of the elements of non-dominated vectors (i.e., all elements of e are equally spaced from one another).

SSC metric estimates the size of the global dominated set in objective space. It computes the area of objective function space covered by the non-dominated solutions vector. In our problem, each dominated vector represents a rectangle defined by the points (0,0) and $(f_1(x_i), f_2(x_i))$, where $f_1(x_i)$ and $f_2(x_i)$ represent a non-dominated solution (see Fig. 9). Therefore, SSC is computed as the union of the areas of all the rectangles that correspond to the non-dominated solutions. An algorithm resulting in low values of SSC may have a high convergence performance. In contrast, the higher values of SSC may result in better diversity performance (i.e., non-dominated solutions are distributed in Pareto optimal set over the non-dominated region).

We run the ACO-DR and ACO-DRC algorithms twice for each instance and take the average as the measure of performance. We also record the computational time required to obtain the Pareto fronts. Table 6 summarizes the results obtained by the applied algorithms and WSM.

Table 6 indicates that the applied algorithms are able to obtain more non-dominated solutions than the WSM. Also, it is clear that the applied algorithms are able to obtain non-dominated solutions in about 0.8% of the WSM CPU time. For example, for instance 1 and 3, we observe that the difference between their execution time is more than ten hours, on the WSM, while on the ACO-DR algorithm is just about 3 minutes.

Table 6 also shows that, the WSM achieves better ESS performance metric than the applied algorithms. In contrast, the applied algorithms achieve better SSC performance metric than the WSM. A possible reason for this is that a less number of non-dominated solutions enable the WSM to result a low value of SSC, but meanwhile it leads to low diversity among the elements in Pareto-optimal set over the non-dominated region. This means that the applied algorithms perform better with respect to high diversity.

The ACO-DRC algorithm results better than the ACO-DR algorithm in solving MMIP-MEC-TCT-R model (see Fig. 9). From Table 6, it is clear that the ACO-DRC algorithm achieves better number of non-dominated solutions, CPU time, and SSC performance metrics than ACO-DR algorithm. A possible reason for the ACO-DRC performs the best is that the ACO-DRC has characteristics (i.e., sorting methods) that allow it to maintain enough non-dominated solutions in the final ant population.

However, in order to gain more insight into these results, from Table 6, the percentage of solutions for the applied algorithms and WSM indicates what percentage of Pareto front solutions are from each method if Pareto fronts from all methods are combined. Fig. 10 illustrates the combined Pareto front from the applied algorithms and WSM for instance 2 (i.e., considers only the non-dominated solutions that form the local Pareto-optimal sets). For example, in instance 2, 33% of the combined Pareto front comes from the ACO-DRC algorithm, whereas 85% comes from the WSM, but the ACO-DR algorithm is not in the local Pareto-optimal sets. In addition, Fig. 10 shows that ACO-DRC and ACO-DR have obtained an identical local Pareto-optimal solution from the exact solutions found in the WSM method (see the red circle on the Fig. 10), as well as identical solutions from one another. Although the numbers of non-dominated solutions are similar, on average 39% of the combined Pareto-optimal solutions are from the ACO-DRC algorithm, 78% of the solutions are from the WSM, and only 6% of the solutions are from the ACO-DR algorithm. Therefore, we can conclude that the ACO-DRC algorithm outperforms the performance of the ACO-DR algorithm with respect to solution quality and also outperforms the WSM with respect to computational CPU time for obtaining the approximate Pareto front. More importantly, the applied algorithms have been able to come closer to the WSM's Pareto-optimal front.

9. Selecting a solution from the Pareto front for implementation: a case study

After the performance of the proposed ACO-DRC algorithm is examined, we consider a case study to show how the decision maker can select the best solution among the obtained approximate Pareto front. The algorithm uses the values obtained by parameter fine tuning. In the case study, 40 jobs are available simultaneously and are sequenced on a single-machine preemptive scheduling. In Fig. 11, we can see the obtained Pareto fronts. We observe that the solutions on the Pareto fronts are well distributed and there is no cluster of solutions. Each solution represents sequence of jobs with their total completion time and total energy cost. This wide range of solutions allows the decision maker to select the best solution.

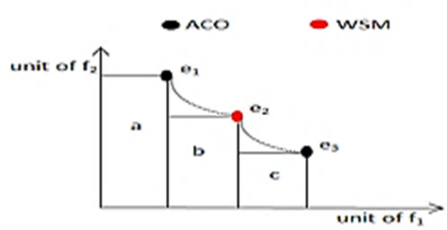

Fig. 9. Measure of SSC performance of a combined set of non-dominated solutions.

Table 6
Results of the WSM and applied algorithms.

	Instance	No. of Jobs	PFH	CPU (sec)	# of Non-dominated	% of solutions	ESS	SSC
ACO-DR	1	10	0.75	240	30	0.105	4.937	27957
	2	20	0.5	366	33	0.00	7.695	33064
	3	30	0.75	417	35	0.059	6.326	29358
	4	50	0.5	735	38	0.159	6.841	38441
	5	100	0.75	1257	43	0.371	8.315	53717
ACO-	1	10	0.75	216	34	0.421	5.246	29375
	2	20	0.5	347	37	0.333	7.870	34131
	3	30	0.75	382	36	0.412	6.387	32717
	4	50	0.5	686	42	0.486	6.315	39155
	5	100	0.75	993	47	0.519	7.931	57907
WSM	1	10	0.75	>40,000	15	-	3.892	24931
	2	20	0.5	>40,000	18	-	5.953	28795
	3	30	0.75	>40,000	12	-	2.748	22467
	4	50	0.5	>40,000	28	-	2.879	23086
	5	100	0.75	>40,000	37	-	3.292	38251

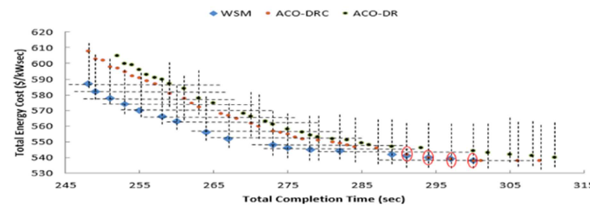

Fig. 10. The Pareto exact and approximation obtained by WSM and applied algorithms respectively for instance 2 from Table 6.

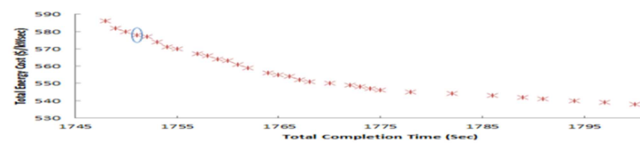

Fig. 11. Pareto fronts for 40 random jobs with optimized parameter and optimal Pareto front with MOORA selection.

MOORA is the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints. In a decision making problem, the objectives (attributes) must be measurable and their outcomes can be measured for every decision alternative. Moreover, interested reader can also refer to Karande and Chakraborty (2012) for more information on MOORA process. Fig. 12, shows the MOORA procedure. Raw data form the basis of a decision matrix with objectives (attributes) as columns and alternatives as rows. In our case, the attributes taken into the consideration are ranked as follows: 1) total energy cost; 2) total number of off-times on the planning horizon; 3) total completion time; 4) makespan (i.e., completion time of the last job). The decision matrix ($\exists_{mb}$) showing the performance of different alternatives with respect to various attributes as shown in Table 8.

Table 8 presents the performance measure of the m^{th} alternatives ($\mathcal{M} = 32$) and b^{th} decision attributes ($\mathcal{B} = 4$). The following is the matrix of the entropies of the b^{th} attribute:

$$\upsilon_b = [0.23, 0.26, 0.24, 0.27]$$

After applying the MOORA procedure, the decision maker is able to compute the $\bar{y}_m$ normalized assessment value of m^{th} alternative with respect to all the attributes, and υ_b weight vector. Note that, the $\bar{y}_m$ value can be positive or negative depending on the totals of its maxima and minima in the decision matrix. An ordinal ranking of $\bar{y}_m$ shows the final preference. Thus, the best alternative has the highest $\bar{y}_m$ value, while the worst alternative has the lowest $\bar{y}_m$ value. Table 9 represents the obtained $\bar{y}_m$ values where the decision maker is able to rank them in decreasing order and the preferred solution is the one with maximum value of $\bar{y}_m$. Therefore, using the results from Table 9, the decision maker can select the best solution among all non-dominated solutions based on his/her preference. According to the decision maker's preferences and MOORA, the best solution is solution number eleven (see Table 9 and highlighted solution in Fig. 11).

10. Conclusion

Given dynamic energy prices over the last ten years, energy efficiency scheduling which involves saving energy and reducing environmental impacts provided an immediate opportunity to decrease energy intensity. Therefore, this study provides a tool for manufacturing professionals to reduce energy costs via production scheduling. Here, we have proposed a mixed-integer multiobjective mathematical model to minimize the total completion time and total energy cost by inserting machine on/off time on a single-machine preemptive scheduling setting (i.e. by simply changing the state of the machines between on/off modes), where preemptions are allowed. We use the WSM to illustrate the mathematical model and to gain insight into the multiobjective problem. Due to the computational complexity involved in solving the mathematical model (i.e., problem is NP-hard), an efficient multiobjective ACO algorithm to solve larger-sized problems in a reasonable amount of CPU time was proposed to obtain an approximate set of non-dominated solutions. The proposed multiobjective ACO-DRC algorithm outperforms ACO-DR algorithm with respect to the quality of approximate Pareto front. GA-3's approximate Pareto front has diverse solutions and is converging to the WSM's Pareto-optimal front with lower computational CPU time. The MOORA method to select the most appropriate solution based on different criteria was utilized.

Extensive computational experimentation is performed to test the performance and effectiveness of the developed algorithms by varying several parameter values. It was observed that selecting good parameters can play a vital role in the CPU's running time. Furthermore, results for ACO-DRC perform the best in solving MMIP-MEC-TT-R model compared to ACO-DR. And also, results show the abilities of ACO-DRC in finding diverse solutions in the front and converging to the true Pareto front with a better spread and more non-dominated solutions in the Pareto-optimal region than other algorithms. After fine tuning the ACO-DRC, the algorithm was illustrated on a case study. A MOORA method was used to assist the decision maker in choosing the most efficient schedule with an appropriate energy cost level.

A direct extension of this research would be to solve the MMIP-MEC-TCT model by other exact solution techniques that can generate supported and non-supported Pareto-optimal solutions such as eps-constraint method, since the WSM can only generate the supported Pareto-optimal solutions (i.e., the solutions that lie on the convex envelop of the Pareto frontier). Also, different metaheuristic implementations (nondominated sorting genetic algorithm II (NSGA-II), GRASP, etc.) can be proposed and their solution quality can be compared to determine the most effective metaheuristics to solve the MMIP-MEC-TCT problem.

Another research direction would be to study other problems with conflicting objectives on various operating environments (i.e., with different scheduling objectives, having sequence-dependent setup times, or on multi-machine settings). For example, the formulation may change significantly when other scheduling objectives are considered on a single-machine setting, but the proposed solution approach will be quite similar, except for the mixed-integer linear programming, which is used to generate the approximate Pareto front and may change in ACO algorithm based on the type of scheduling objective that is employed. For example, if we are interested in the makespan objective (i.e., the maximum completion time objective (C_{max})) instead of total completion time, then we just remove the completion time objective (1), and then add the following constraint $C_{max} - \sum_{j=1}^{n} C_j \geq 0$ and makespan objective $Min\ C_{max}$. A more detailed experiment is needed to determine if there is a clear pattern between parameters of the experimental design and the objective could be investigated. Finally, some other Multiple-criteria decision analysis methods can be implemented as a future research scope and the results would be compared to identify the efficiency and significance of the Multiple-criteria decision analysis methods in determining the preferred alternatives.

Table 8
Decision matrix with initial values.

Alternative (m)	**Attributes (b)**			
	Total Energy Cost	**Total Number of off-times**	**Total Completion Time**	**Makespan**
1	586	35	1748	66
2	582	35	1749	66
3	580	35	1750	66
4	578	35	1751	66
5	577	39	1752	70
6	574	39	1753	70
7	571	42	1754	73
8	570	42	1755	73
9	567	42	1757	73
10	566	49	1758	80
11	564	51	1759	82
12	563	53	1760	84
13	561	53	1761	84
14	559	53	1762	84
15	556	58	1764	89
16	555	58	1765	89
17	554	62	1766	93
18	552	64	1767	95
19	551	66	1768	97
20	550	66	1770	97
21	549	68	1772	99
22	548	73	1773	104
23	547	75	1774	106
24	546	75	1775	106
25	545	80	1778	111
26	544	80	1782	111
27	543	85	1786	116
28	542	88	1789	119
29	541	90	1791	121
30	540	90	1794	121
31	539	95	1797	126
32	538	95	1800	126

Table 9
Ranking of alternatives applying ratio system.

Alternative (m)	$\bar{y}_m$	**Rank**
1	-0.14235	4
2	-0.14248	3
3	-0.14260	2
4	-0.14288	1
5	-0.14698	6
6	-0.14718	5
7	-0.15021	9
8	-0.15039	8
9	-0.15044	7
10	-0.15870	10
11	-0.16101	11
12	-0.16315	14
13	-0.16327	13
14	-0.16340	12
15	-0.16902	16
16	-0.16907	15
17	-0.17384	17
18	-0.17616	18
19	-0.17852	20
20	-0.17854	19
21	-0.18093	21
22	-0.18697	22
23	-0.18931	24
24	-0.18936	23
25	-0.19540	25

26	-0.19542	26
27	-0.20154	27
28	-0.20519	28
29	-0.20760	30
30	-0.20760	29
31	-0.21369	32
32	-0.21369	31

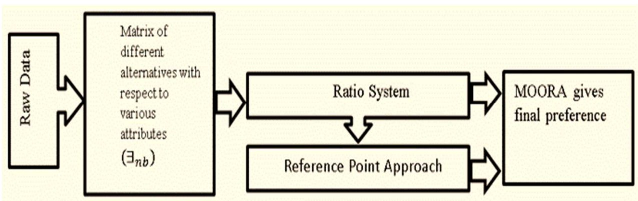

Fig. **12.** Diagram of MOORA.

References

Akturk, M. S., Ghosh, J. B., & Gunes, E. D. (2003). Scheduling with tool changes to minimize total completion time: a study of heuristics and their performance. Naval Research Logistics (NRL), 50(1), 15-30.

Annual Energy Outlook. (2013). U.S. Energy Information Administration (EIA) Department of Energy (Vol. 2014). http://www.eia.gov/forecasts/aeo/pdf/0383(2013).pdf.

Belton, V., & Stewart, T. (2002). Multiple criteria decision analysis: an integrated approach. Springer Science & Business Media.

Caramia, M., & Dell'Olmo, P. (2008). Multi-objective Management in Freight Logistics Increasing Capacity, Service Level and Safety with Optimization Algorithms. London: Springer http://www.springer.com/gp/book/9781848003811.

Chang, Q., Xiao, G., Biller, S., & Li, L. (2013). Energy Saving Opportunity Analysis of Automotive Serial Production Systems (March 2012). Automation Science and Engineering, IEEE Transactions on, 10(2), 334-342.

Che, A., Zhang, S., & Wu, X. (2017). Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs. Journal of Cleaner Production, 156, 688-697.

Che, A., Zeng, Y., & Lyu, K. (2016). An efficient greedy insertion heuristic for energy-conscious single machine scheduling problem under time-of-use electricity tariffs. Journal of cleaner production, 129, 565-577.

Chen, G., Zhang, L., Arinez, J., & Biller, S. (2013). Energy-efficient production systems through schedule-based operations. Automation Science and Engineering, IEEE Transactions on, 10(1), 27-37.

Da Xu, Li, Wu He, and Shancang Li. "Internet of things in industries: A survey." IEEE Transactions on industrial informatics 10.4 (2014): 2233-2243.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on, 6(2), 182-197.

Ding, J. Y., Song, S., Zhang, R., Chiong, R., & Wu, C. (2016). Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches. IEEE Transactions on Automation Science and Engineering, 13(2), 1138-1154.

Dorigo, M., & Stützle, T. (2003). The ant colony optimization metaheuristic: Algorithms, applications, and advances Handbook of metaheuristics (pp. 250-285): Springer.

Drake, R., Yildirim, M. B., Twomey, J. M., Whitman, L. E., Ahmad, J. S., & Lodhia, P. (2006). Data collection framework on energy consumption in manufacturing. Paper presented at the IIE Annual Conference and Exposition, May 20, 2006 -May 24, 2006. Orlando, FL, United states: Institute of Industrial Engineers. http://hdl.handle.net/10057/3422

Fang, K., Uhan, N., Zhao, F., & Sutherland, J. W. (2011). A new shop scheduling approach in support of sustainable manufacturing Glocalized Solutions for Sustainability in Manufacturing (pp. 305-310): Springer.

Fang, K., Uhan, N. A., Zhao, F., & Sutherland, J. W. (2014). Scheduling on a single machine under time-of-use electricity tariffs. Annals of Operations Research, 1-29.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: FormulationDiscussion and Generalization. Paper presented at the ICGA.

Gutowski, T., Murphy, C., Allen, D., Bauer, D., Bras, B., Piwonka, T., Wolff, E. (2005). Environmentally benign manufacturing: observations from Japan, Europe and the United States. Journal of Cleaner Production, 13(1), 1-17.

ILOG, I. (2012). Inc. CPLEX 12.5 User Manual: Somers, NY, USA.

Jaszkiewicz, A. (2001). Multiple objective metaheuristic algorithms for combinatorial optimization. Poznan University of Technology, Poznon, Poland.

Kanet, J. J., & Sridharan, V. (2000). Scheduling with inserted idle time: problem taxonomy and literature review. Operations Research, 48(1), 99-110.

Karande, P., & Chakraborty, S. (2012). Application of multi-objective optimization on the basis of ratio analysis (MOORA) method for materials selection. Materials & Design, 37, 317-324.

Kim, I. Y., & De Weck, O. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. Structural and multidisciplinary optimization, 29(2), 149-158.

Lee, S., Do Chung, B., Jeon, H. W., & Chang, J. (2017). A dynamic control approach for energy-efficient production scheduling on a single machine under time-varying electricity pricing. Journal of Cleaner Production, 165, 552-563.

Lei, M., & Feng, Z. (2012). A proposed grey model for short-term electricity price forecasting in competitive power markets. International Journal of Electrical Power & Energy Systems, 43(1), 531-538.

Pinedo, M. (2005). Planning and scheduling in manufacturing and services (Vol. 24): Springer Series in Operations Research. Springer Science and Business Media, Inc.

Pohlheim, H. (1998). Genetic and Evolutionary Algorithm Toolbox for use with MATLAB. Dept. Comput. Sci., Univ. Ilmenau, Ilmenau, Germany.

Prabhu, V. V., Jeon, H. W., & Taisch, M. (2012). Modeling green factory physics—An analytical approach. Paper presented at the Automation Science and Engineering (CASE), 2012 IEEE International Conference on.

Rubaiee, S., Cinar, S., & Yildirim, M. B. (2018). An Energy-Aware Multiobjective Optimization Framework to Minimize Total Tardiness and Energy Cost on a Single-Machine Nonpreemptive Scheduling. IEEE Transactions on Engineering Management, (99).Salido, M. A., Escamilla, J., Barber, F., Giret, A., Tang, D., & Dai, M. (2013). Energy-aware parameters in job-shop scheduling problems. GREEN-COPLAS 2013: IJCAI 2013 Workshop on Constraint Reasoning, Planning and Scheduling Problems for a Sustainable Future, 44-53.

Schipper, M. (2006). Energy-related carbon dioxide emissions in US manufacturing Energy Information Administration (EIA) Report #: DOE/EIA-0573(2005).

Schott, J. R. (1995). Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master of Science Thesis, Massachusetts Institute of Technology, MIT.

Sharma, A., Zhao, F., & Sutherland, J. W. (2015). Econological scheduling of a manufacturing enterprise operating under a time-of-use electricity tariff. Journal of Cleaner Production, 108, 256-270

Swaminathan, V., & Chakrabarty, K. (2003). Energy-conscious, deterministic I/O device scheduling in hard real-time systems. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 22(7), 847-858.

Van Veldhuizen, D. A., & Lamont, G. B. (1998). Multiobjective evolutionary algorithm research: A history and analysis. In T. R. TR-98-03 (Ed.). Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998: Citeseer.

Yildirim, M. B., & Mouzon, G. (2012). Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm. Engineering Management, IEEE Transactions on, 59(4), 585-597.

Zhang, H., Zhao, F., Fang, K., & Sutherland, J. W. (2014). Energy-conscious flow shop scheduling under time-of-use electricity tariffs. CIRP Annals-Manufacturing Technology, 63(1), 37-40.

Zhang, H., Zhao, F., & Sutherland, J. W. (2015). Energy-efficient scheduling of multiple manufacturing factories under real-time electricity pricing. CIRP Annals-Manufacturing Technology, 64(1), 41-44.

Zhang, S., Che, A., Wu, X., & Chu, C. (2018). Improved mixed-integer linear programming model and heuristics for bi-objective single-machine batch scheduling with energy cost consideration. Engineering Optimization, 50(8), 1380-1394.