IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 23, 2022, accepted April 14, 2022, date of publication April 25, 2022, date of current version May 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3170105

Task Scheduling–Review of Algorithms and Analysis of Potential Use in a Biological Wastewater Treatment Plant

TOMASZ UJAZDOWSKI AND ROBERT PIOTROWSKI
Faculty of Electrical and Control Engineering, Gdańsk University of Technology, 80-233 Gdańsk, Poland
Corresponding author: Tomasz Ujazdowski (tomasz.ujazdowski@pg.edu.pl)

ABSTRACT The idea of task scheduling is to increase the efficiency of a system by minimising wasted time, evenly loading machines, or maximising the throughput of machines. Moreover, the use of appropriate scheduling algorithms often leads to a reduction in the energy costs of the process. Task scheduling problems are found in a variety of industrial areas, and their scale changes significantly depending on the problem. This review shows the extent to which task scheduling methods are applied in industry. This paper presents methods and algorithms for solving task scheduling problems. In addition, an analysis of the possibility of using task scheduling methods to improve the efficiency of biological wastewater treatment plants was also conducted. This approach is based on the assumption of a balanced workload for multiple reactors. Analysing the case study of a wastewater treatment plant in Swarzewo in Northern Poland is applied.

INDEX TERMS Algorithms, optimisation, SBR, task scheduling, wastewater treatment plant.

I. INTRODUCTION

Optimisation and scheduling are applied wherever there is a need to reduce the cost and time of the performed processes. Scheduling problems are common in modern technologies. Many algorithms have been developed to solve simple single- or multi-machine problems [1]–[3]. However, algorithms exist for most simple scheduling problems. More efficient algorithms are still being sought, and more complex problems are being analysed. A large number of new scheduling methods and problems originate from the need to optimise computer computing. Where methods of optimal task scheduling allow for more efficient use of central processing unit (CPU) time and so result in higher performance of the operations performed.

Task scheduling issues are optimization problems where a global sequence of task execution is sought, taking into account: time constraints, prioritisation of operations, relations between tasks related to the necessary technological order. The expected result is an optimal schedule with respect to the criterion adopted. The criterion may vary depending on the task. The two basic ones are the shortest summary time for task completion (makespan) and balanced machine load.

The associate editor coordinating the review of this manuscript and approving it for publication was Shaoyong Zheng.

In task scheduling issues, a distinction is made between the occurrence of jobs, resources and operations. Jobs are executed through the realisation of operations that consume or occupy (use without consumption) resources. Depending on the type and characteristics of the tasks, it can distinguish between single-machine and multi-machine scheduling, flow-based or nest-based scheduling.

A task is a specific process at the end of which it obtain a final product. In the information technology (IT) industry, a task performed in a computing cloud or on a single processor may be the execution of calculations within a specific programme or algorithm. In electronics and automation, a task is usually a technological process. Tasks may also refer to providing the required amount of energy to perform an operation by a system component. Such problems are called Energy Harvesting issues. In many industries, there are also Crew Scheduling issues, where tasks are defined in terms of the number of people needed to accomplish a specific goal.

Resources can be classified into three types. The first one concerns consumable resources such as: materials, semi-finished products, machine elements, and chemical resources. The second type are limited resources and include: electricity, water, capital. The last type of resources are renewable resources like personnel, machines, processor time and robots.

Operations are separate activities that are necessary for the execution of a task, but can be distinguished from each other by the time of execution or the type and number of resources consumed. The necessary sequence of execution called technological order is also distinguished for operations.

In the general notation of the task scheduling problem, three sets corresponding to the mentioned arguments are used. J = {1,2, …, j} is the set of tasks (jobs), where each task consists of operations represented by the set - O = {1,2, …, o}. The last set is M = {1,2, …, m} – a set of machines, whereas in the literature one can observe various designations describing these sets. The solution to the problem defined this way are the start and end times of tasks assigned to machines.

The history of job scheduling problems began with the presentation of the job shop scheduling problem. The name of this problem comes from its original source, the problem of scheduling customer orders (jobs) in shop. Job shop is an optimisation issue commonly encountered in computer science and operations research, and at the same time the best known of the scheduling problems. Many variants of this problem have been created, which introduce additional peculiarities that change the structure of the problem. Such modifications include the presence of parallel machine definitions or the presence of necessary downtime between tasks.

One special variant of this issue is the flow-shop problem. This definition of the problem assumes a strict sequence of execution of all operations in all tasks. It corresponds to the representation of a production process or computational projects, where it is not possible to execute the next stage without finishing the previous one.

In general terms, algorithms that solve task scheduling problems are found in, among others:

- automotive industry, where they mainly cover the problems of manufacturing components for assembly or optimising the operations performed on the machines of an assembly line;
- the chemical and petrochemical industries, where batches of products are processed in chemical reactors and the characteristics and duration of the process determine the type of product obtained;
- the construction industry, both in the production of building materials and the planning of construction projects;
- the electronics industry, where automated production lines produce different types of electronic assemblies or components;
- information technology and cloud databases;
- layers of business management, including resources and crew planning.

Task scheduling problems outside the IT environment are much more difficult issues. Industrial processes have detailed and specific requirements, management objectives are diverse and often contradictory (operating costs versus quality), and the production environment is characterised by significant uncertainties.

This paper will present an overview of task scheduling methods in different domains, indicating similarities in the construction of problem models. A proposed model of a task scheduling problem biological wastewater treatment plant (WWTP) (for several sequential batch reactors (SBRs)) will also be shown.

The remainder of this paper is organized as follows. Review of algorithms used in scheduling tasks and description of selected research works are described in Section 2. Description of the biological WWTP, defining the scheduling problem and analysis of potential methods to be applied in scheduling the operation of several SBRs are presented in Section 3. The last section presents the conclusions.

II. REVIEW OF TASK SCHEDULING ALGORITHMS

A. ALGORITHMS REVIEW

Methods for solving task scheduling problems can be divided into two main groups. The first group consists of exact methods, i.e., those that perform calculations in a deterministic way. The second one is non-deterministic methods, i.e., those that use randomization mechanisms. The results obtained by this type of algorithms present only approximate solutions. Simple problems are usually solved by deterministic methods, but with the difficulty of the problem, the computational complexity and time required to find solutions increases significantly. As a result, solving complex problems with these methods is not implemented in practice. Optimisation issues concerning plants and problems belong to NP-hard issues. Therefore, the contemporary trend focuses on the development of non-deterministic algorithms. Many subgroups for these methods have been created, among others: genetic and evolutionary strategies, search methods, artificial intelligence algorithms, simulated methods.

A graphical representation of the methods and subcategories is shown in Fig. 1. The list of methods discussed in the review with references are as follows:

- deterministic methods:
  - Branch and Bound (B&B) [4]–[6]
  - Dynamic Programming (DP) [7]–[10]
  - Linear Programming (LP) [9]–[13]
  - . . .
- non-deterministic methods:
  - Genetic Algorithms (GA) [14]–[18]
  - Differential Evolution (DE) [19]–[21]
  - Memetic Algorithm (MA) [22]–[25]
  - Cultural Algorithm (CA) [26]–[28]
  - Particle Swarm Optimization (PSO) [29]–[32]
  - Artificial Bee Colony (ABC) [33]–[37]
  - Tabu Search (TS) [38]–[43]
  - Local Search (LS) [44], [45]
  - Beam Search (BS) [46]–[48]
  - Simulated Annealing (SA) [49]–[52]
  - Artificial Neural Network (ANN) [14], [53]–[57]

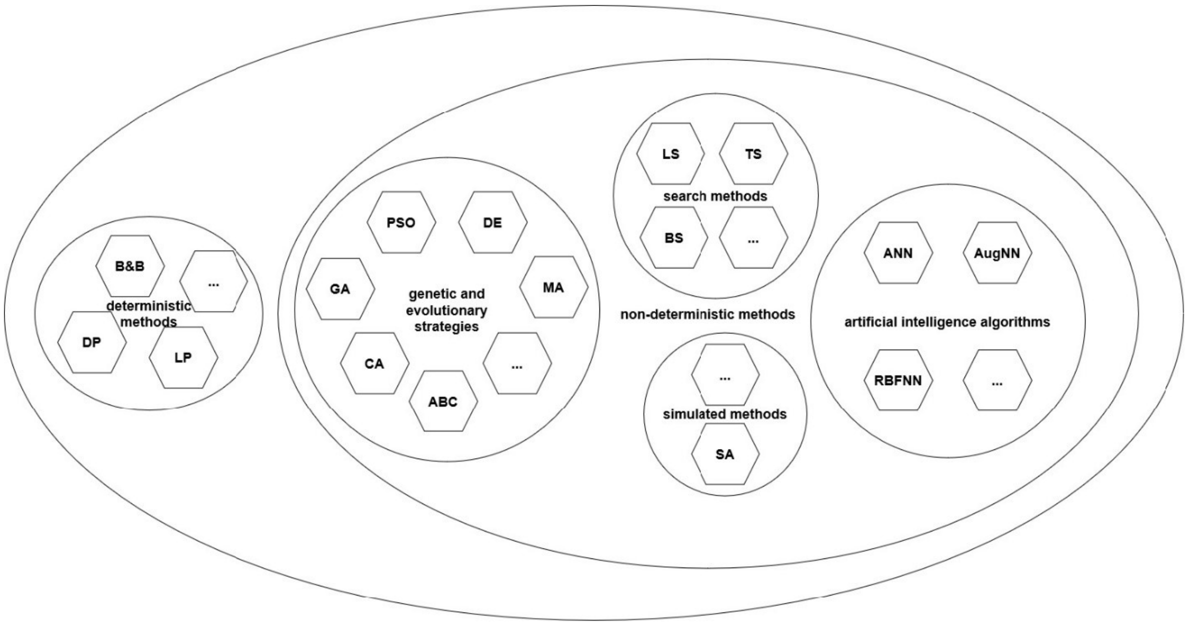

FIGURE 1. Classification of task scheduling algorithms.

- Radial Basis Function Neural Networks (RBFNN) [55]
- Augmented Neural Networks (AugNN) [56]
- ...

Papers [4], [6], [14], [15], [24], [32], [35], [38], [39], [41], [44], [48], [52] concern problems defined in terms of work shop scheduling, in different variants. Research papers [5], [20], [25], [54], [58] deal with flow shop problems. Papers [10], [11], [18], [61], [62], [45] concern problems related to human resource management and scheduling of work teams. On the other hand, papers [12], [16], [19], [21], [22], [26], [29], [30], [31], [33], [34], [51], [57] concern task scheduling problems in cloud computing. Another group are papers [23], [49], [54] that address problems related to distributed industrial systems. Other scheduling problems encountered in industry are the subject of papers [8], [9], [13], [27], [37], [40], [53].

The main method described in [4]–[6] was Branch and Bound (B&B). The name Branch and Bound comes from the two main operators used in the algorithm. Branching is a recursive process of replacing the problem space with a set of smaller subproblems. It is achieved by searching the state space. This operation makes the problem become a tree of subproblems. The found solution of the problem establishes an initial upper bound for other branches. The value of the boundary is updated if and only if a better solution is found. The limit is related to the evaluation of possible solutions in the subproblem. In this method one can also distinguish an elimination operation, which consists in eliminating subproblems that do not lead to an improvement of the currently best solution. The B&B algorithm can be represented by a search tree. The tree is rooted in the original problem. An example graphical representation of the method for a simple problem with one machine and three tasks is illustrated in Fig. 2.

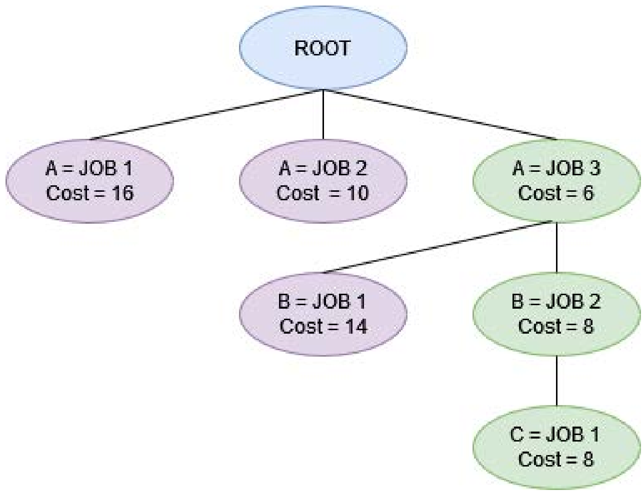

FIGURE 2. Example of branch and bound tree.

The key technique for solving task scheduling problems is Dynamic Programming (DP) [7]–[10]. Similar to the B&B method, in algorithms using the DP strategy, a division of the problem into subproblems is introduced by specifying additional parameters. The optimal solution of the full problem is a function of the solutions of the subproblems. The tasks are solved from smaller to larger ones. DP algorithms are based on finding a recursive equation that describes the optimal value of the objective function. According to [8], [10] a full implementation of dynamic programming for solving the problem of task scheduling, which are mostly NP-hard problems, is too expensive for direct implementation. In [8], modifications to the algorithm have been proposed by introducing a function approximator, whereas in [10] the use of dynamic programming to determine the lower bound of the problem based on minimum paths was described.

One of the basic exact techniques for solving mathematical programming problems is Linear Programming (LP). The use of algorithms of this type in task scheduling problems has been presented in [9], [11]–[13]. The problem of tasks scheduling, as an optimization problem, is reduced to the form of linear constraints. According to [9] reducing the scheduling process to the form of linear programming requires a simple transformation. The number of tasks, operations, or variables representing resources are transformed into constraints allowing the use of classical LP solution methods. However, such a transformation can be applied only to simple problems of task scheduling in which there are no special features such as prioritization, technological order, or required minimum completion times of specific tasks.

Genetic Algorithms (GA) are based on the mechanisms of evolution and natural selection observed in nature [14]–[18]. A special feature of these algorithms is the inclusion of a population of solutions on which crossover and mutation operations leading to the creation of new solutions are performed. There are many methods of mutation, selection or succession. Each of these methods changes the behaviour of the algorithm to a certain extent. There are many variations of GA, but they all assume an identical cycle of the algorithm (see Fig. 3).

The main parameters of these algorithms are crossover probability, mutation probability and initial population.

Selection is a method of choosing individuals from a population to be crossed and mutated. Crossover is a method of combining individuals to produce a new individual. Mutation makes a small change in an individual. Succession is the process of creating a new population for the next cycle of the algorithm. In the classical view, the new individuals created by the crossover method become the new population of solutions, but there are methods that change the rules of succession. Basic among these methods is the elitism, ensuring that the best individual from the previous population becomes part of the new one.

An individual, or a single solution to a problem in genetic algorithms is represented as a chromosome. That is, a binary sequence in which the actual numerical representation of the result is encoded.

Evolutionary strategies, have the same genesis as GA but have a different method of representing individuals. In evolutionary algorithms an individual in the population represents a specific solution to a task in real numerical form. The result of not having a limited number of bits forming a chromosome is to improve the accuracy of the computation. However, evolutionary algorithms require different (adapted to the task) crossover, mutation, selection or succession operations. An example of crossover representation using an evolutionary algorithm is shown in Fig. 4.

A subtype of evolutionary strategies is the Differential Evolution (DE) algorithm [19]–[21]. The characteristic feature of this algorithm is the creation of new individuals (solutions) by combining several individuals from the population. This is achieved by summing the vector

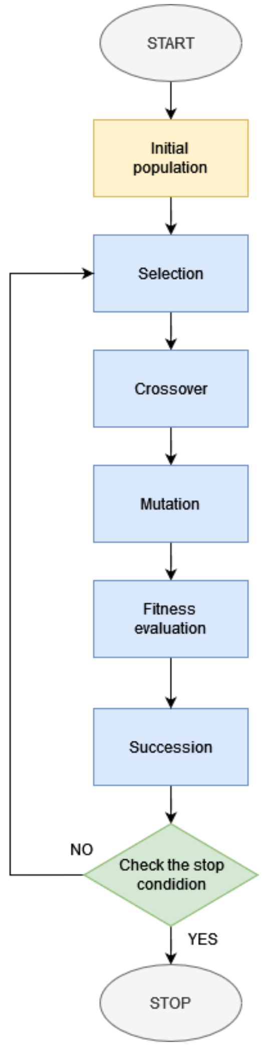

FIGURE 3. Scheme of the genetic algorithm.

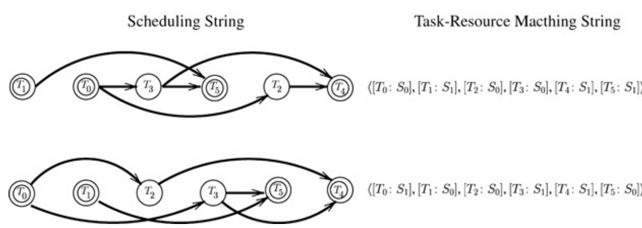

FIGURE 4. Scheduling string mutation and matching string crossover [20].

differences of any two individuals in the population with a third individual. The new individual becomes part of the next population only if its adaptation function is better than that of its parent. Such a selection scheme ensures that only solutions better than or equal to the current one are generated from iteration to iteration.

An application of the Memetic Algorithm (MA) was presented in [22]–[25]. The term "memetic" is inspired by Richard Dawkin's concept of the meme, the basic unit of cultural transmission [63]. It is an algorithm based on a GA, with a local search method introduced for a metaheuristic population. Local search is implemented through other search algorithms, so the MA is hybrid and its exact design depends on the search method applied.

Another algorithm from the evolutionary family is the Cultural Algorithm (CA) [26]–[28]. It uses the assumption that cultural evolution is seen as a process of inheritance. In the description of this algorithm, two interacting spaces are distinguished: the population space and the belief space. In the first one, the microevolutionary process takes place, i.e. concerning the individuals in the population. Each individual in this layer is defined by behavioural traits that may be socially acceptable or unacceptable. The best individuals in the population through characteristics defined as acceptable influence the development of macroevolution in the belief space. In turn, the knowledge contained in the belief space in a macroevolutionary way influences the process of creating new individuals. Due to the features of this algorithm, the search for new optimal solutions is directed and additional knowledge about the solutions found so far is stored. A representation of the spaces and their interaction in the CA is shown in Fig. 5.

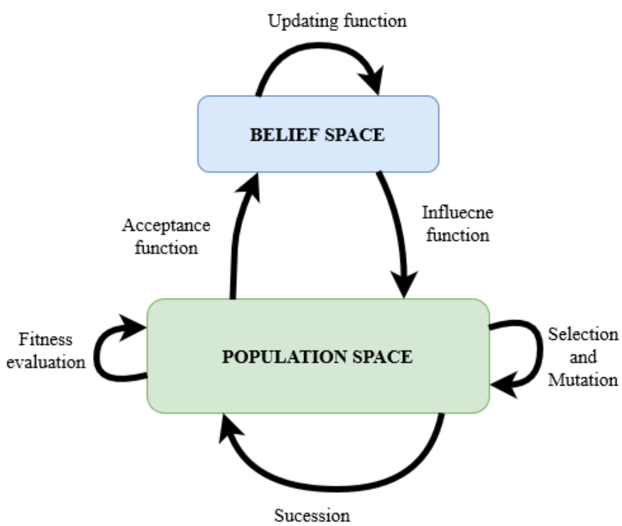

FIGURE 5. **Scheme of the cultural algorithm.**

In [29]–[32] optimization using Particle Swarm Optimization (PSO) was described. The algorithm is derived from mechanisms observed in nature for the behaviour of flocks of birds and schools of fish. PSO is an algorithm with memory. Each individual in the population, called a particle, has knowledge of its best solution, and the global best solution. The position of the particle is a representation of the solution, and is determined by the particle's own knowledge as well as interactions with its neighbours. The dual mechanism based on local (own knowledge) as well as global (neighbours' knowledge) search balances exploration and exploitation.

Another evolutionary approach is the Artificial Bee Colony (ABC). An adaptation of this algorithm to task scheduling issues was presented in [33]–[37]. This approach is relatively new, as it was first presented in 2005. It simulates the behaviour of a hive of honey bees while searching for food. The algorithm is used to solve non-convex and non-smooth optimization problems. Each food source in the algorithm represents a real solution to the problem, in turn the quality of the source (or amount of nectar) corresponds to the value of the adaptation function. The ABC algorithm assumes that there is only one bee for each food source. The population in this algorithm is divided into three groups with different tasks. A distinction is made between: employed bee, onlooker bee, and scout bee. The scout bees search for new food sources, the employed bees exploit the selected source, and the onlooker bees select the best of the current solutions and indicate.

Tabu Search (TS) is one of the most popular search algorithms used in task scheduling problems [38]–[43]. It is usually used in a hybrid setup with the Path Relinking method. TS is a strategy for solving optimization problems using a local search adaptation mechanism. It takes into account both the possibility of deteriorating the function in a given step, which allows getting out of local minima, and forbidden moves (Tabu), which introduce penalties for returning to previous solutions. This mechanism is an efficient search algorithm with a strongly developed exploitation of the solution space. Using the procedure of path relinking, which allows finding new solutions between existing ones, an exploration method - global search - is introduced into the algorithm.

A Local Search (LS) strategy in task scheduling problems has been the subject of research works [44], [45]. Algorithms of this type start with a sample solution and then proceed to neighbouring solutions, searching the local area for an optimum. To use this strategy, the problem must be represented as a space with a neighbourhood relation. LS is not a specific algorithm, but an approach to solving an optimisation problem. It is most often used as an additional element in other algorithms.

Another of the search methods is Beam Search (BS) [46]–[48]. This heuristic method is an adaptation of the exact B&B method, in which the tree search is not performed completely, but only in a certain direction related to the potential of the nodes. The evaluation of the potential of the nodes can be performed by different methods, and the nodes with the highest potential index are referred to as promising. In this search method at any level, only the promising nodes are kept for further branching and the other nodes are skipped. This approach allows for increased computational speed compared to the B&B method, but by rejecting initially inferior solutions introduces the possibility of bypassing a global extreme.

The Simulated Annealing (SA) algorithm is a method that uses the temperature behaviour of the metallurgical process. It is the main method for solving scheduling problems in [49]–[52]. Simulated annealing algorithms are characterised by their ability to search the problem globally and their ability to get out of local minima. The algorithm assumes the ability to accept potentially worse solutions, but with running time and simulated cooling this ability disappears. In the algorithm there is a continuity of transition between the exploration and exploitation phases, which are distinguished in other non-deterministic algorithms.

Task scheduling techniques based on Artificial Neural Networks (ANN) are the subject of research in [14], [53]–[57]. ANN are a constantly developing approach inspired by the work of the human brain. The algorithms being developed simulate the information processing by neurons. In the context of neural networks, it is difficult to talk about specific algorithms, because the way of learning and the architecture of the network, vary depending on the research task.

In general terms, a neural network consists of layers of neurons. The inputs of the network are stimuli provided by the environment and the output is the response to the data stimulating specific neurons in the layers. A function describing each neuron decides whether a given stimulus causes its activation. Neural networks require a learning process, during which the weights in each layer of the network are matched. A neural network is able to learn complex relationships and consequences of actions taken in the external environment by changing these weights. Similar to evolutionary algorithms, there are many types of neural networks used in the solution of serialisation problems. The authors of the paper [55] propose the use of radial basis functions (RBF) as activation functions in neurons. On the other hand, the authors of [56] present an Augmented Neural Network (AugNN) architecture, which exploits the domain-specific knowledge of the problem by using the correspondence between the network structure and the problem structure.

B. LITERATURE REVIEW

In [50] a solution to the problem of scheduling independent tasks by using a SA algorithm was presented. The problem is an NP-hard problem. The task is defined taking into account the special real characteristics of the problem model. The presented algorithm is described including: the objective function, the termination conditions, the neighbour search method and the cooling method, which is a special feature of this class of algorithms. The authors presented test results in Matlab environment with comparison to a number of algorithms like Min-min, Min-max, and three different types of genetic algorithms.

A solution to the hybrid flow-shop scheduling problem using a MA was described in [25]. The physical representation of the problem is the scheduling of tasks in a multiprocessor. The problem description relates directly to the classical notation of the task scheduling problem, with the assumed criterion of minimising the completion time of all tasks.

In [7] was presented the use of DP in solving the task scheduling problem to guarantee optimal energy consumption for IoT devices when managing energy resources from solar panels. This paper demonstrates the applicability of task scheduling in Energy Harvesting.

More research work on energy optimization was presented in [57] where the problem of energy efficiency in cloud data centers is described. Unlike other papers addressing the task scheduling in cloud technologies, the authors of this research work focus mainly on energy resources. Similar topics was addressed in [59], [70].

An interesting adaptation of the task scheduling problem is Crew Scheduling in the building industry. The problem of multi-object undertakings was presented in [61]. The author presents more than 50% reduction of the project execution time due to the use of the TS algorithm. It is worth emphasising the complexity of the presented problem and a detailed description of its real features, having their representation in the presented data.

In [62] the balanced workload of personnel operating several WWTPs is discussed. The scheduling problem is defined as the distribution of annual working time of four types of work units: Manager, Assistant, Laboratory, Executive in a certain number of facilities.

In [38] the application of a hybrid solution combining the classical TS algorithm with the Path Relinking (PR) method was described. The authors focus on comparing the performance of the designed algorithm for a number of benchmark job shop problem installations, with known solutions in the literature. The proposed solution was tested for six problem sets, on a total of 205 known benchmark job shop scheduling problems of varying difficulty. Whereby the proposed algorithm improved the solution for as many as 49 out of 205 tested instances. In the presented solution, there is a repeated switch between the PR activity for generating new promising solutions and TS looking for local optima. This allows the algorithm to combine the exploration stage with the exploitation stage.

In [30] was presented the use of the mathematical model Load Balancing Mutation as an add-on to the PSO algorithm in a cloud computing environment. The solution has been used for multi-criteria optimization considering reliability, execution time, transmission time, Makespan, transmission cost, or load balancing between tasks in virtual machines. The solution proposed by the authors aims to take into account the reliability and availability of the cloud computing environment in the task scheduling process, which, according to the authors, is neglected in other works on similar topics due to the difficulty of achieving these parameters.

Another hybrid approach is presented in [14]. The approach is two-stage and uses two problem spaces. The authors' proposal is to use a GA in cooperation with machine learning. Operating in the heuristic space, the genetic algorithm decides on the assignment of tasks to specific machines, and the knowledge base developed in the parameter space through machine learning decides on the transfer of these tasks. In a first step, the order of task transfer is determined, then the order of tasks for specific machines is optimised.

The paper [19] describes the application of a DE algorithm to optimise resource placement in a cloud computing environment. The authors focus on the balance between exploration and exploitation, which is an important aspect of finding solutions in complex multi-criteria problems. A valuable aspect of the paper is the presentation of the adaptation of evolutionary operations mechanisms to the scheduling problem.

Aspects of energy efficiency optimization in cloud technologies are addressed in [57]. The authors propose an innovative approach using machine learning techniques to make decisions on reducing the consumption of energy resources. The presented approach distinguishes between two scheduling layers. Separate neural networks have been developed for each layer. The first layer is responsible for the selection of the computing rack and was named ANN-based Rack Scheduler (ARS). The second is ANN-based Task Scheduler (ATS), its task is to schedule resources in a particular rack. The multi-criteria optimisation task aimed to reduce makespan and energy consumption with a minimum number of active racks. Results show a reduction in energy consumption.

III. BIOLOGICAL WWTP

The analysis of the scheduling problem for WWTP was based on a real system located in Northern Poland, in the Swarzewo. This facility was modernized in 2018 and it is a mechanical-biological-chemical treatment plant. On average, about 5,000 m^3 of wastewater per day is treated there annually. The paper considers only the biological part (see Fig. 6).

Technological scheme of the WWTP in Swarzewo is shown in Fig. 7. Swarzewo WWTP consists of six SBRs (three small: SBR1 – SBR3 and three large: SBR4 – SBR6). Small tanks have a capacity of about 5000 m^3, a diameter of 30 m. The large ones, on the other hand, have a capacity of about 6500 m^3, and their diameter is 34 m. All tanks have the same height of 7 m.

Air is supplied to the reactors from the aeration system. Each reactor has its own installation. Depending on the size of the reactor, blowers with a capacity of 115 kW (for small SBR) or 150 kW (for large SBR) are used. The blowers come from a common manufacturer and their performance characteristics are very similar. This arrangement allows for separate control of the phases of each of the reactors. Detailed information on the reactor aeration system relates to the diffuser system. They have two branches, the left branch consisting of 616 diffusers and the right branch consisting of 600 diffusers. Their opening takes place after exceeding 2 kPa pressure. Sanitaire membrane diffusers of Flygt company are installed in large SBR. Both branches have 900 diffusers each. Their opening takes place after exceeding the pressure of 1.5 kPa.

Detailed information on the reactor aeration system relates to the diffuser system. They have two branches, the left branch consisting of 616 diffusers and the right branch consisting of 600 diffusers. Their opening takes place after exceeding 2 kPa pressure. Sanitaire membrane diffusers of Flygt company are installed in large SBR. Both branches have 900 diffusers each. Their opening takes place after exceeding the pressure of 1.5 kPa.

The amount of pollution in summer is even three times higher than in other seasons. This is mainly related to tourism in the region of the facility in question. In summer, the quality requirements for treated wastewater also change. Information on acceptable levels of pollutants concentration in treated wastewater is presented in European and Polish standards. The values compliant with the water permit for the Swarzewo treatment plant are presented in Table 1.

FIGURE 6. Biological part of the WWTP in Swarzewo [64].

TABLE 1. Maximum pollutant concentrations in effluent.

Parameter	Maximum acceptable value	Unit
Chemical oxygen demand	125	g O_2/m^3
Total nitrogen (Ntot)	10 (from July 1 to August 31) 15 (for the rest of the year)	g N/m^3
Total phosphorus (Ptot)	1 (from July 1 to August 31) 2 (for the rest of the year)	g P/m^3

Wastewater treatment is a long process consisting of several stages. In the model presented here, only biological treatment is included. It is assumed that the initial mechanical treatment processes are performed according to a strict technological order and are continuous, i.e., there is no need to schedule them. These are mechanical treatment processes using screens, grit chambers or sand traps. Biological treatment takes place with the help of microorganisms that make up the activated sludge. The removal of different compounds requires different conditions in the SBR. Five main phases of the treatment process can be distinguished. These are (see Fig. 8):

- Filling phase – untreated wastewater is pumped into the SBR.
- Reaction phase – biological processes take place in the SBR.
- Sedimentation phase – treated wastewater accumulates in the upper part of the SBR.
- Decantation phase – treated wastewater is pumped out into natural water reservoirs, rivers, lakes.
- Shutdown phase – waiting time for the next batch of wastewater.

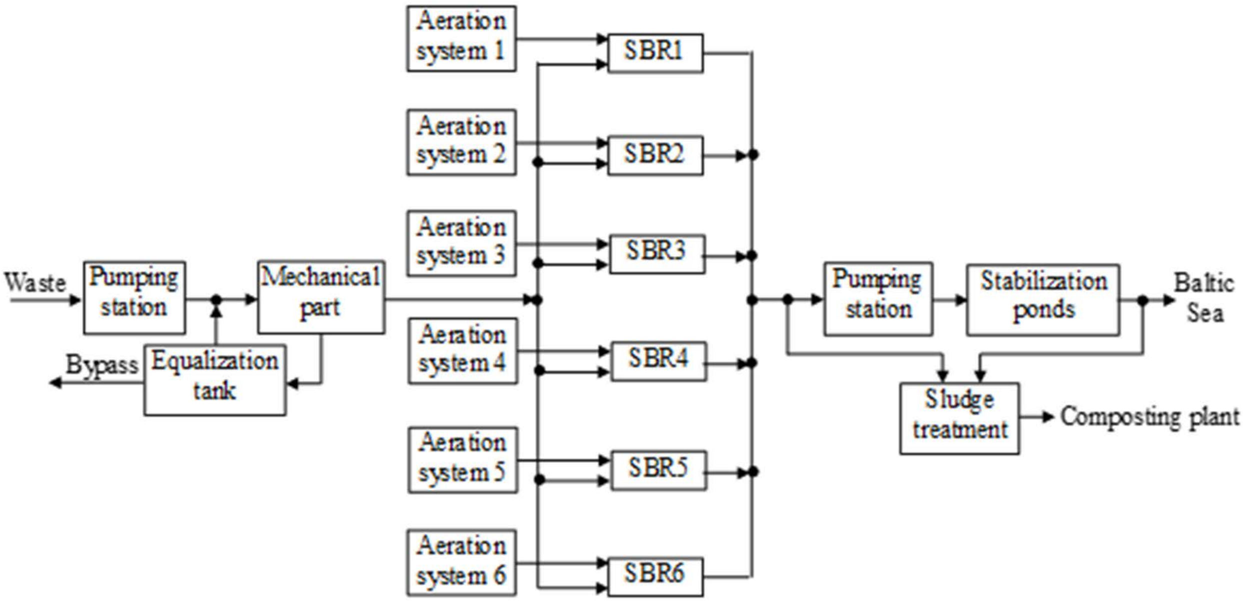

FIGURE 7. Technological scheme of the WWTP in Swarzewo [65].

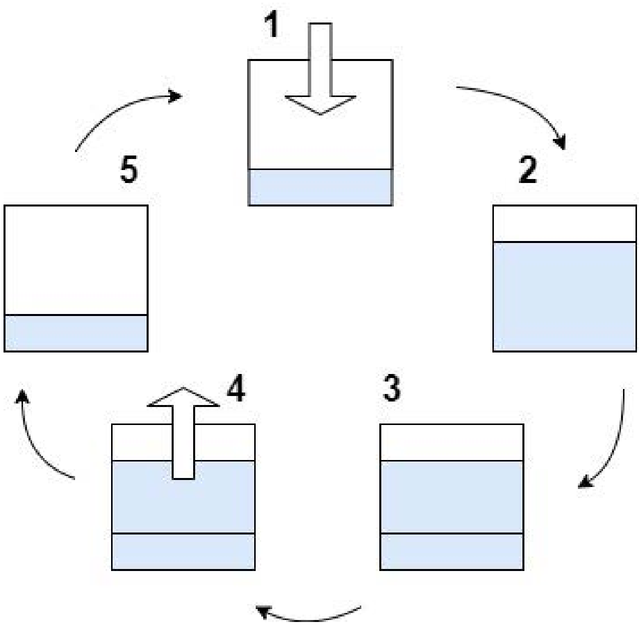

FIGURE 8. Symbolic diagram of the operating phases of SBR. 1. Filling phase, 2. Reaction phase, 3. Sedimentation phase, 4. Decantation phase, 5. Shutdown phase.

The reaction phase consists of alternating aerobic and anaerobic conditions, the change of conditions occurs several times and the durations are of variable length. Additionally, during the reaction phase portions of untreated wastewater are pumped into the SBR. This process can be considered as an additional operation.

IV. TASK SCHEDULING IN BIOLOGICAL WWTP

The task in the considered problem of task scheduling is a single cycle of SBR tank operation - i.e. the process of treating a given volume of wastewater. In view of the presented work cycle, it can distinguish the operations that are part of one task, where the reaction phase is divided into several aerobic and anaerobic phases, interrupted by short refilling operations.

The machines on which the operations are carried out consist of six SBRs divided into two classes depending on volume - small tanks and large tanks. The size of the tanks affects the volume of treated wastewater portions, as well as the energy costs associated with blower operation. The actual level of wastewater in the reactor through the hydrostatic pressure affects the operating costs of the blowers. The constraints in the scheduling task are mainly the minimum and maximum volume of the SBRs.

The wastewater for the six SBRs is spilled from the retention tank. In the scheduling issue, the current state of the retention tank is responsible for the resources in the process, and its maximum and minimum volume levels are additional limits for the process. Based on historical data, externally supplied resources can be determined. Defining the retention tank as a resource storage with a limited storage space also makes it possible to take into account changes in the inflow of wastewater due to weather conditions (significant rainfall or drought) and seasons.

Each refill operation is defined in terms of duration, operating costs and volume of wastewater transported from the retention tank to the SBR. Sedimentation and reaction operations under anaerobic conditions are defined by duration only. Reaction operations under aerobic conditions are represented by duration and energy consumption for blower operation (from the aeration system). Tank emptying operation is defined analogously to refilling, but uses only energy resources. Examples of average operation times for historical

TABLE 2. Example duration of each operation of the SBR cycle.

Operation	Large SBR operation time	Small SBR operation time
Anaerobic 1	3h 10min	5h 3min
Refill 1	3h 25min	3h 38min
Aerobic 1	6h 16min	5h 21min
Anaerobic 2	4h 16min	3h 50min
Refill 2	3h 24min	2h 43min
Aerobic 2	5h 26min	4h 30min
Anaerobic 3	3h 35min	2h 7min
Refill 3	1h 30min	23min
Aerobic 3	1h 55min	4h 53min
Sedimentation	1h	58min
Decantation	33min	31min
Pumping out	31min	35min

data of the five main phases of the treatment process are shown in Table 2.

The aim of scheduling is to evenly load the SBRs and optimise operating costs, within the constraints of the volumes of storage tanks and SBRs. The time horizon for work scheduling can be a month or a quarter. The problem of scheduling the work of several SBRs assumes that there is no change of machines between operations within a specific task. On the other hand, the times of adding pollutants to the tanks and their quantity are an important aspect. The decision variables in the process are the start times of the tasks and the assignment of tasks to machines (see Fig. 9). These variables actually create a schedule and through the parameters presented in the following section, affects the value of the objective function.

The problem of scheduling the operation of several SBR tanks can be defined as follows. A set J = {J1, J2, ..., Jn} of independent tasks is given, which correspond to complete work cycles of an SBR. Each task is described by a sequence of operations O1, O2, ..., On, which must be performed according to the technological order assuming that operation Oi+1 can be performed only after the completion of operation Oi.

The operations include: O1 – pumping the wastewater into the empty SBR, O2a,i – reaction phase under anaerobic conditions, O2b,i – refilling the wastewater during the reaction phase, O2c,i – reaction phase under aerobic conditions, O3 – sedimentation and decantation, O4 - emptying the tank. Index i is related to the given number of aerobic and anaerobic phases in the process, which can be variable depending on the task (see Table 2).

The set of machines is represented by M = {M1, M2, M3, M4, M5, M6}, where M1, M2, M3 correspond to small SBRrs and M4, M5, M6 to large reactors.

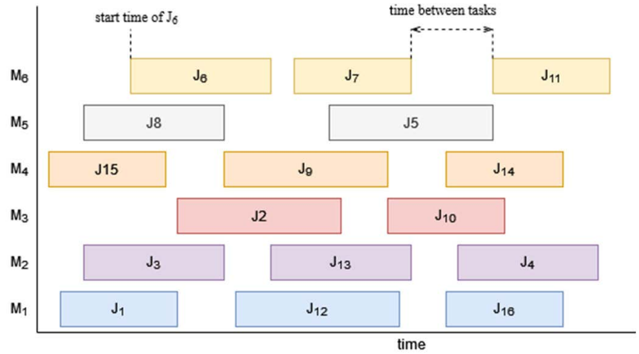

FIGURE 9. Symbolic representation of the task schedule, J1, J2, ..., J16 – tasks; M1, M2, ..., M6 – machines.

Two sets are distinguished as resources. The first Rv = $\{Rv_1, Rv_2, \ldots, Rv_n\}$ is related to the volume of raw wastewater pumped from the retention tank into SBR. The second Re = $\{Re_1, Re_2, \ldots, Re_n\}$ is related to the energy resources used for pumping and aerating wastewater.

Limitation of the volume of the retention tank:

$$V_r^{min} \leq V_r \leq V_r^{max} \tag{1}$$

Initial volume of the retention tank:

$$V_r^0 = V_r^{initial} \tag{2}$$

Limiting the volume of SBRs:

$$V_i^{min} \leq V_i \leq V_i^{max} \tag{3}$$

where i represents the machine index.

Objective function:

$$F = \min \left\{ w1 \cdot Var\left(M_{time}\right) + w2 \cdot \sum_{i=1}^{k_e} \left(C_m \cdot Re_i\right) - w3 \cdot \sum_{i=1}^{k_v} Rv_i \right\} \tag{4}$$

where: w1, w2, w3 – weights of the components of the objective function, Mtime – matrix of summary working times of machines, Cm – table of energy costs per day, k_e – number of operations using the first resource, k_v – number of operations using the second resource, Re_i – values of resource one for the i-th operation, Rv_i – values of resource two for the i-th operation.

The objective function is represented as an equation with three components. The first one of the elements represents the variance between the total task times on individual machines. By minimising the variance, we ensure a balanced workload. The second component represents the operational costs of the treatment plant. These are related to resource consumption and electricity prices, which vary according to the time of day. The third component refers to the volume of treated wastewater.

The complexity of the task scheduling in biological WWTP means that deterministic algorithms will not be applicable in practice due to the computation time required. Solutions to these problems should be sought in heuristic algorithms. Their use to optimise the length and sequence of phases of a single SBR has so far been successfully applied and presented in [66], [67]. The direction of adaptation of these algorithms to the management of several SBRs should therefore be a natural path for development and further research. A binary encoded notation of the problem, would require extremely long chromosomes describing the individual solution to the problem, so algorithms using a real-number notation of the solution are suggested.

V. CONCLUSION

This paper presents a review of algorithms used for task scheduling in various industrial fields. A classification into solution finding strategies has been presented. The presented review shows that adaptation of optimization algorithms to task scheduling problems is possible and requires appropriate problem definition. A task scheduling problem model for the management of several SBRs has been defined, which has not been presented in the literature so far.

WWTPs are critical facilities for water management. Taking care of the quality of treated water has an impact on the environment. Thus, reducing energy costs leads to less production of carbon footprint related pollutants. Therefore, optimising the operating schedule of wastewater treatment plants, as well as improving the control algorithms themselves, has a positive impact on the environment.

Reducing operating costs, related to electricity consumption, is also a welcome development for facility managers. The costs of implementing advanced algorithms are also incomparably lower than a full hardware upgrade installation of more efficient, environmentally friendly equipment.

Task scheduling is common in industry. New more efficient algorithms are constantly being designed because real-world problems belong to multi-criteria and NP-hard issues, so finding exact, globally optimal solutions is practically implementable.

To the best of the authors' knowledge, research on the introduction of scheduling algorithms for the management of several SBRs has not been the subject of research work to date.

REFERENCES

[1] E. Balas, "Machine sequencing via disjunctive graphs: An implicit enumeration algorithm," *Oper. Res.*, vol. 17, no. 6, pp. 941–957, Dec. 1969.

[2] J. Grabowski, "Formulation and solution of the sequencing problem with parallel machines," *Appl. Mathematicas*, vol. 16, pp. 215–232, 1978.

[3] S. M. Johnson, "Optimal two- and three-stage production schedules with setup times included," *Nav. Res. Logistics Quart.*, vol. 1, no. 1, pp. 61–68, Mar. 1954.

[4] A. Dabah, A. Bendjoudi, and A. AitZai, "Efficient parallel B&B method for the blocking job shop scheduling problem," in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Innsbruck, Austria, Jul. 2016, pp. 784–791, doi: 10.1109/HPCSim.2016.7568414.

[5] G. B. Mcmahon and P. G. Burton, "Flow-shop scheduling with the branch-and-bound method," *Oper. Res.*, vol. 15, no. 3, pp. 473–481, Jun. 1967, doi: 10.1287/opre.15.3.473.

[6] P. Brucker, B. Jurisch, and B. Sievers, "A branch and bound algorithm for the job-shop scheduling problem," *Discrete Appl. Math.*, vol. 49, nos. 1–3, pp. 107–127, 1994, doi: 10.1016/0166-218X(94)90204-6.

[7] A. Caruso, S. Chessa, S. Escolar, X. del Toro, and J. C. Lopez, "A dynamic programming algorithm for high-level task scheduling in energy harvesting IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2234–2248, Jun. 2018, doi: 10.1109/JIOT.2018.2828943.

[8] J. H. Lee and J. M. Lee, "Approximate dynamic programming based approach to process control and scheduling," *Comput. Chem. Eng.*, vol. 30, nos. 10–12, pp. 1603–1618, Sep. 2006, doi: 10.1016/j.compchemeng.2006.05.043.

[9] W. Zhang, Y. Hu, H. He, Y. Liu, and A. Chen, "Linear and dynamic programming algorithms for real-time task scheduling with task duplication," *J. Supercomput.*, vol. 75, no. 2, pp. 494–509, Feb. 2019, doi: 10.1007/s11227-017-2076-9.

[10] J. E. Beasley and B. Cao, "A dynamic programming based algorithm for the crew scheduling problem," *Comput. Oper. Res.*, vol. 25, nos. 7–8, pp. 567–582, 1998, doi: 10.1016/S0305-0548(98)00019-7.

[11] M. N. Atika, N. S. Zyurina, and R. Ahmad, "Binary linear programming model in solving bus crew problem as tactical fixed task scheduling," in *Proc. J. Phys., Conf.*, vol. 1212, 2019, Art. no. 012030, doi: 10.1088/1742-6596/1212/1/012030.

[12] K. Chakrabarty, "Test scheduling for core-based systems using mixed-integer linear programming," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 10, pp. 1163–1174, 2000, doi: 10.1109/43.875306.

[13] R. Manzini, R. Accorsi, T. Cennerazzo, E. Ferrari, and F. Maranesi, "The scheduling of maintenance. A resource-constraints mixed integer linear programming model," *Comput. Ind. Eng.*, vol. 87, pp. 561–568, Sep. 2015, doi: 10.1016/j.cie.2015.06.006.

[14] C.-Y. Lee, S. Piramuthu, and Y.-K. Tsai, "Job shop scheduling with a genetic algorithm and machine learning," *Int. J. Prod. Res.*, vol. 35, no. 4, pp. 1171–1191, 1997, doi: 10.1080/002075497195605.

[15] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Comput. Oper. Res.*, vol. 35, no. 10, pp. 3202–3212, 2008, doi: 10.1016/j.cor.2007.02.014.

[16] Y. Xu, K. Li, J. Hu, and K. Li, "A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues," *Inf. Sci.*, vol. 270, pp. 255–287, Jun. 2014, doi: 10.1016/j.ins.2014.02.122.

[17] J. Singh and G. Singh, "Improved task scheduling on parallel system using genetic algorithm," *Int. J. Comput. Appl.*, vol. 39, no. 17, pp. 17–22, Feb. 2012, doi: 10.5120/4912-7449.

[18] A. G. dos Santos and G. R. Mateus, "General hybrid column generation algorithm for crew scheduling problems using genetic algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, May 2009, pp. 1799–1806, doi: 10.1109/CEC.2009.4983159.

[19] J.-T. Tsai, J.-C. Fang, and J.-H. Chou, "Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm," *Comput. Oper. Res.*, vol. 40, pp. 3045–3055, Dec. 2013, doi: 10.1016/j.cor.2013.06.012.

[20] A. K. M. K. A. Talukder, M. Kirley, and R. Buyya, "Multiobjective differential evolution for scheduling workflow applications on global grids," *Concurrency Comput., Pract. Exper.*, vol. 21, no. 13, pp. 1742–1756, Sep. 2009, doi: 10.1002/cpe.1417.

[21] Y. Li, S. Wang, X. Hong, and Y. Li, "Multi-objective task scheduling optimization in cloud computing based on genetic algorithm and differential evolution algorithm," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 4489–4494, doi: 10.23919/ChiCC.2018.8483505.

[22] B. Keshanchi and N. J. Navimipour, "Priority-based task scheduling in the cloud systems using a memetic algorithm," *J. Circuits, Syst. Comput.*, vol. 25, no. 10, Oct. 2016, Art. no. 1650119, doi: 10.1142/S021812661650119X.

[23] M. H. Kashani and M. Jahanshahi, "A new method based on memetic algorithm for task scheduling in distributed systems," *Int. J. Simul., Syst. Sci. Technol.*, vol. 10, no. 5, pp. 25–31, 2009.

[24] M. Frutos, A. C. Olivera, and F. Tohmé, "A memetic algorithm based on a NSGAII scheme for the flexible job-shop scheduling problem," *Ann. Oper. Res.*, vol. 181, no. 1, pp. 745–765, Dec. 2010, doi: 10.1007/s10479-010-0751-9.

[25] A. Jouglet, C. Oğuz, and M. Sevaux, "Hybrid flow-shop: A memetic algorithm using constraint-based scheduling for efficient search," *J. Math. Model. Algorithms*, vol. 8, no. 3, pp. 271–292, Aug. 2009, doi: 10.1007/s10852-008-9101-1.

[26] J.-W. Li and C.-W. Qu, "Cloud computing task scheduling based on cultural genetic algorithm," in *Proc. MATEC Web Conf.*, vol. 40, 2016, Art. no. 09008, doi: 10.1051/matecconf/20164009008.

[27] X. Yuan, H. Nie, Y. Yuan, A. Su, and L. Wang, "Hydrothermal systems generation scheduling using cultural algorithm," *J. Hydroinform.*, vol. 11, no. 1, pp. 65–78, 2009, doi: 10.2166/hydro.2009.056.

[28] L. Shutian, Y. Da, H. Fuben, and Y. Chunjing, "An improved cultural algorithm for scheduling of precedent tasks," in *Proc. IEEE Int. Conf. Artif. Intell. Ind. Design (AIID)*, Guangzhou, China, May 2021, pp. 169–173, doi: 10.1109/AIID51893.2021.9456558.

[29] T. Chen, B. Zhang, X. Hao, and Y. Dai, "Task scheduling in grid based on particle swarm optimization," in *Proc. 5th Int. Symp. Parallel Distrib. Comput.*, Timisoara, Romania, Jul. 2006, pp. 238–245, doi: 10.1109/ISPDC.2006.46.

[30] A. I. Awad, N. A. El-Hefnawy, and H. M. Abdel-Kader, "Enhanced particle swarm optimization for task scheduling in cloud computing environments," *Proc. Comput. Sci.*, vol. 65, pp. 920–929, Jan. 2015, doi: 10.1016/j.procs.2015.09.064.

[31] F. Ramezani, J. Lu, and F. Hussain, "Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization," in *Service-Oriented Computing*, vol. 6470, P. P. Maglio, M. Weske, J. Yang, and M. Fantinato, Eds. Berlin, Germany: Springer, 2013, pp. 237–251, doi: 10.1007/978-3-642-45005-1_17.

[32] G. Zhang, X. Shao, P. Li, and L. Gao, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem," *Comput. Ind. Eng.*, vol. 56, no. 4, pp. 1309–1318, May 2009, doi: 10.1016/j.cie.2008.07.021.

[33] J.-Q. Li and Y.-Q. Han, "A hybrid multi-objective artificial bee colony algorithm for flexible task scheduling problems in cloud computing system," *Cluster Comput.*, vol. 23, no. 4, pp. 2483–2499, Dec. 2020, doi: 10.1007/s10586-019-03022-z.

[34] N. J. Navimipour, "Task scheduling in the cloud environments based on an artificial bee colony algorithm," in *Proc. Int. Conf. Image Process.*, Istanbul, Turkey, Jun. 2015, pp. 38–44.

[35] L. Wang, G. Zhou, Y. Xu, S. Wang, and M. Liu, "An effective artificial bee colony algorithm for the flexible job-shop scheduling problem," *Int. J. Adv. Manuf. Technol.*, vol. 60, nos. 1–4, pp. 303–315, Apr. 2012, doi: 10.1007/s00170-011-3610-1.

[36] J.-Q. Li, Q.-K. Pan, and M. F. Tasgetiren, "A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities," *Appl. Math. Model.*, vol. 38, no. 3, pp. 1111–1132, Feb. 2014, doi: 10.1016/j.apm.2013.07.038.

[37] W. Kimpan and B. Kruekaew, "Heuristic task scheduling with artificial bee colony algorithm for virtual machines," in *Proc. Joint 8th Int. Conf. Soft Comput. Intell. Syst. (SCIS) 17th Int. Symp. Adv. Intell. Syst. (ISIS)*, Hokkaido, Japan, Aug. 2016, pp. 281–286, doi: 10.1109/SCIS-ISIS.2016.0067.

[38] B. Peng, Z. Lü, and T. C. E. Cheng, "A Tabu search/path relinking algorithm to solve the job shop scheduling problem," *Comput. Oper. Res.*, vol. 53, pp. 154–164, Jan. 2015, doi: 10.1016/j.cor.2014.08.006.

[39] S. Jia and Z. H. Hu, "Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem," *Comput. Oper. Res.*, vol. 47, no. 1, pp. 11–26, Jul. 2014, doi: 10.1016/j.cor.2014.01.010.

[40] X. Zhang, X. Li, and J. Wang, "Local search algorithm with path relinking for single batch-processing machine scheduling problem," *Neural Comput. Appl.*, vol. 28, no. S1, pp. 313–326, Dec. 2017, doi: 10.1007/s00521-016-2339-z.

[41] M. M. Nasiri and F. Kianfar, "A guided Tabu search/path relinking algorithm for the job shop problem," *Int. J. Adv. Manuf. Technol.*, vol. 58, nos. 9–12, pp. 1105–1113, Feb. 2012, doi: 10.1007/s00170-011-3441-0.

[42] F. Glover, "Tabu search—Part I," *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, 1989, doi: 10.1287/ijoc.1.3.190.

[43] F. Glover, "Tabu search—Part II," *ORSA J. Comput.*, vol. 2, no. 1, pp. 4–32, 1990, doi: 10.1287/ijoc.2.1.4.

[44] B. M. Ombuki and M. Ventresca, "Local search genetic algorithms for the job shop scheduling problem," *Int. J. Speech Technol.*, vol. 21, no. 1, pp. 99–109, Jul. 2004, doi: 10.1023/B:APIN.0000027769.48098.91.

[45] B. Estellon, F. Gardi, and K. Nouioua, "High-performance local search for task scheduling with human resource allocation," in *Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics*, vol. 5752, T. Stützle, M. Birattari, and H. H. Hoos, Eds. Berlin, Germany: Springer, 2009, pp. 1–15, doi: 10.1007/978-3-642-03751-1_1.

[46] P. S. Ow and T. E. Morton, "Filtered beam search in scheduling," *Int. J. Prod. Res.*, vol. 26, no. 1, pp. 35–62, 1988, doi: 10.1080/00207548808947840.

[47] I. Sabuncuoglu and M. Bayiz, "Job shop scheduling with beam search," *Eur. J. Oper. Res.*, vol. 118, no. 2, pp. 390–412, Oct. 1999, doi: 10.1016/S0377-2217(98)00319-1.

[48] W. Shi-Jin, Z. Bing-Hai, and X. Li-Feng, "A filtered-beam-search-based heuristic algorithm for flexible job-shop scheduling problem," *Int. J. Prod. Res.*, vol. 46, no. 11, pp. 3027–3058, Jun. 2008, doi: 10.1080/00207540600988105.

[49] M. H. Kashani and M. Jahanshahi, "Using simulated annealing for task scheduling in distributed systems," in *Proc. Int. Conf. Comput. Intell., Modeling Simulation*, Brno, Czech Republic, 2009, pp. 265–269, doi: 10.1109/CSSim.2009.36.

[50] A. A. P. Kazem, A. M. Rahmani, and H. H. Aghdam, "A modified simulated annealing algorithm for static task scheduling in grid computing," in *Proc. Int. Conf. Comput. Sci. Inf. Technol.*, Singapore, Aug. 2008, pp. 623–627, doi: 10.1109/ICCSIT.2008.163.

[51] X. Liu and J. Liu, "A task scheduling based on simulated annealing algorithm in cloud computing," *Int. J. Hybrid Inf. Technol.*, vol. 9, no. 6, pp. 403–412, Jun. 2016, doi: 10.14257/ijhit.2016.9.6.36.

[52] S. Chakraborty and S. Bhowmik, "Job shop scheduling using simulated annealing," in *1st Int. Conf. Comput. Commun. Advancement*, vol. 1, no. 1, Kalyani, India, Jan. 2013, pp. 69–73.

[53] B. Dorronsoro and F. Pinel, "Combining machine learning and genetic algorithms to solve the independent tasks scheduling problem," in *Proc. 3rd IEEE Int. Conf. Cybern. (CYBCONF)*, Exeter, U.K., Jun. 2017, pp. 1–8, doi: 10.1109/CYBConf.2017.7985766.

[54] A. V. Gritsenko, N. G. Demurchev, V. V. Kopytov, and A. O. Shulgin, "Decomposition analysis and machine learning in a workflow-forecast approach to the task scheduling problem for high-loaded distributed systems," *Mod. Appl. Sci.*, vol. 9, no. 5, pp. 38–49, 2015, doi: 10.5539/mas.v9n5p38.

[55] B. Tripathy, S. Dash, and S. K. Padhy, "Dynamic task scheduling using a directed neural network," *J. Parallel Distrib. Comput.*, vol. 75, no. 5, pp. 101–106, Jan. 2015, doi: 10.1016/j.jpdc.2014.09.015.

[56] A. Agarwal, H. Pirkul, and V. S. Jacob, "Augmented neural networks for task scheduling," *Eur. J. Oper. Res.*, vol. 151, no. 3, pp. 481–502, 2003, doi: 10.1016/S0377-2217(02)00605-7.

[57] M. Sharma and R. Garg, "An artificial neural network based approach for energy efficient task scheduling in cloud data centers," *Sustain. Comput., Inform. Syst.*, vol. 26, Jun. 2020, Art. no. 100373, doi: 10.1016/j.suscom.2020.100373.

[58] H. Wang, V. Jacob, and E. Rolland, "Design of efficient hybrid neural networks for flexible flow shop scheduling," *Expert Syst.*, vol. 20, no. 4, pp. 208–231, Sep. 2003, doi: 10.1111/1468-0394.00245.

[59] G. V. Merrett and B. M. Al-Hashimi, "Energy-driven computing: Rethinking the design of energy harvesting systems," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 960–965, doi: 10.23919/DATE.2017.7927130.

[60] L. Wang and Y. Xiao, "A survey of energy-efficient scheduling mechanisms in sensor networks," *Mobile Netw. Appl.*, vol. 11, no. 5, pp. 723–740, Oct. 2006, doi: 10.1007/s11036-006-7798-5.

[61] M. Podolski, "Management of resources in multiunit construction projects with the use of a Tabu search algorithm," *J. CIVIL Eng. Manage.*, vol. 23, no. 2, pp. 263–272, Jun. 2016, doi: 10.3846/13923730.2015.1073616.

[62] G. De Feo, S. De Gisi, and M. Galasso, "A simple method to equalize the workload when operating several small wastewater treatment plants: A case study," *Environ. Technol.*, vol. 34, no. 12, pp. 1533–1541, Jun. 2013, doi: 10.1080/09593330.2012.758664.

[63] R. Dawkins, *The Selfish Gene*, 3rd ed. Oxford, U.K.: Oxford Univ. Press, 2006.

[64] *Water and Sewage Company 'SWARZEWO' (in Polish)*. Accessed: Jan. 10, 2022. [Online]. Available: https://www.sws-swarzewo.pl/

[65] R. Piotrowski, H. Sawicki, and K. Żuk, "Novel hierarchical nonlinear control algorithm to improve dissolved oxygen control in biological WWTP," *J. Process Control*, vol. 105, pp. 78–87, Sep. 2021, doi: 10.1016/j.jprocont.2021.07.009.

[66] R. Piotrowski, M. Lewandowski, and A. Paul, "Mixed integer nonlinear optimization of biological processes in wastewater sequencing batch reactor," *J. Process Control*, vol. 84, pp. 89–100, Dec. 2019, doi: 10.1016/j.jprocont.2019.10.002.

[67] R. Piotrowski, A. Paul, and M. Lewandowski, "Improving SBR performance alongside with cost reduction through optimizing biological processes and dissolved oxygen concentration trajectory," *Appl. Sci.*, vol. 9, no. 11, p. 2268, May 2019, doi: 10.3390/app9112268.