Open Journal of Optimization, 2025, 14(2), 63-92
https://www.scirp.org/journal/ojop
ISSN Online: 2325-7091
ISSN Print: 2325-7105

Tabu Search Technique for Optimisation of Biomass Waste to Energy Technology

David Makumbi, Afam Uzorka, John Kelechi Ukagwu

Department of Physical Sciences, Kampala International University, Kampala, Uganda
Email: afamuzorka@gmail.com

How to cite this paper: Makumbi, D., Uzorka, A. and Ukagwu, J.K. (2025) Tabu Search Technique for Optimisation of Biomass Waste to Energy Technology. *Open Journal of Optimization*, **14**, 63-92.
https://doi.org/10.4236/ojop.2025.142005

Received: March 28, 2025
Accepted: June 24, 2025
Published: June 27, 2025

Abstract

The Tabu Search heuristic can be used to optimise the WET (waste to energy technology). Developments were made to the basic Tabu Search to adapt it to the optimisation problem. This paper explains the contribution made in development of the adaptations to the basic Tabu Search. The principle of Tabu Search is explained, followed by the statement of the optimisation problem, the description of the optimisation of WET is given, the Tabu Search algorithm is described and the experiments and results are discussed. It was found out that initial thresholds of infeasibility should be set and these should be varied during the optimisation. The multi-objective, multi-period function should be evaluated on a Pareto incumbent front. Different strategies should be used for minimisation of cost and minimisation of infeasibility, and diversification should be done by performing random restarts with the incumbent solution.

Keywords

Tabu Search, Optimization, WET, Technique

1. Introduction

The global pursuit of sustainable energy solutions has intensified the need to harness renewable resources more efficiently. One promising approach is the conversion of biomass waste into usable energy, a process that not only provides an alternative energy source but also addresses environmental concerns related to waste management. Optimising the operation of Waste-to-Energy Technologies (WET) is a complex yet critical task that involves managing multiple objectives, such as minimising operational costs, reducing environmental impact, and maximising energy yield. The WET optimisation problem, therefore, requires careful consideration of diverse factors across multiple time periods and under varying constraints of feasibility.

This paper explores the application and adaptation of the Tabu Search (TS) heuristic to optimise the WET system. Before describing the principle of Tabu Search, the terms used to describe the Tabu Search are defined. A heuristic is an iterative rule used to find an optimum solution that terminates as soon as no immediately accessible solutions can improve the incumbent solution [1]-[3]. A metaheuristic is a master strategy that modifies other heuristics, to produce solutions beyond those that are generated when searching a local optimum [3]-[5]. As such, Tabu Search is a meta-heuristic [6]-[8], which guides a local heuristic search procedure to explore the solution space beyond a local optimum [3] [9] [10].

In Tabu Search, the optimisation problem is formulated as [8]:

$$\text{minimize} \quad f(u) : u \in U, \tag{1}$$

where $f(u)$ is the objective function, and u is selected from a set of constraints U. A move n leads from one solution to the next. The move is defined as [8]:

$$n : U(n) \to U. \tag{2}$$

The moves $n \in N$ that can be applied to u form a set denoted by $N(u)$, and termed the neighbourhood of u [8]. A characteristic of the Tabu Search is to N constrain the search by restricting moves [8]. This leads to creation of an element of memory that is managed using a Tabu list. Moves that result in a good solution, are used to update the current solution and are stored in the Tabu list. The reverse moves are also stored in the Tabu list. Use of memory in the form of a Tabu list prevents cycling, which occurs if a solution is stuck in a local optimum. In the basic Tabu Search, moves that are in the Tabu list are not allowed during the optimisation, during a given number of iterations [8] [11] [12]. The Tabu list is updated by removing older entries and adding new entries with every move. The length of the Tabu list or the number of iterations for which a move is Tabu, is dependent on the optimisation strategy. The basic Tabu Search algorithm is described in Algorithm 1.

Algorithm 1 Basic Tabu Search

1. Select an initial solution $u \in U$
2. Set $u^{\text{incumbent}} \leftarrow u$
3. Set iter $\leftarrow 0$
4. Initialise the Tabu list: $T^{\text{list}} \leftarrow \emptyset$
5. **while** stopping condition is not reached **do**
6. Find the best admissible solution $u \in N(u)$ with respect to $f(u)$
7. **if** $f(u) < f(u^{\text{incumbent}})$ **then**
8. Update the incumbent solution $u^{\text{incumbent}} \leftarrow u$
9. Update the Tabu list
10. **end if**
11. **end while**

where u is the current solution, U is the constraints set, $u^{\text{incumbent}}$ is the incumbent solution, iter is the iteration counter, T is the Tabu list, $N(u)$ is the neighbourhood of solution u and $f(u)$ is the objective function.

The following is an explanation of the choice of Tabu Search over other metaheuristics for solving the optimisation problem. The successful implementation of a metaheuristic is dependent on how well it is modified for the problem being solved [13]-[15]. The waste to energy technology (WET) is a complex model constituting of components that model the energy conversion processes. The digester model, the internal combustion engine model and the induction machine models use complex non-linear differential equations. Each of the models of the WET is a difficult non-linear optimization problem that is treated as a black box. As such a metaheuristic is selected for solving the optimisation problem. The reasons for the choice of Tabu are: (i) it uses a deterministic approach for optimisation, (ii) it moves aggressively to a local optimum and (iii) it can easily be tailored to the optimisation problem. The following is an explanation of these reasons.

Tabu Search uses a deterministic approach to search the solution space, which shortens the computational time. The other metaheuristics like genetic algorithms and simulated annealing perform a random search of the solution space. This results in long computational times, making simulated annealing and genetic algorithms less suited to complex problems like the optimisation of WET. Three metaheuristics namely: Tabu Search, simulated annealing, and genetic algorithms were compared in solving facility location problems, under time-limited, solution-limited, and unrestricted conditions [16]-[18]. Tabu Search showed good performance in most of the facility location problems experimented with, compared to the simulated annealing and genetic algorithm.

Again, compared to simulated annealing, Tabu Search moves aggressively to a local optimum. Simulated annealing works on the premise that a slow decent will lead to a local optimum that is closer to a global optimum. With Tabu Search the best available move is made at each iteration, and the search does not spend time in regions whose solution are less attractive [8] [12]. In [19], a Tabu Search algorithm that diversifies the search by using 3 different neighborhoods was developed for solving a flowshop scheduling problem. The Tabu Search was compared with an ant colony algorithm that was used to solve the same problem. The Tabu Search performed better than the ant colony algorithm.

The third reason for selection of Tabu Search, is that Tabu Search can easily be tailored to take into account the nature of the optimisation problem. This is done by proper selection of variables, handling of constraints and parameter tuning. The success of Tabu Search is as a result of tuning its parameters to the problem being solved [20]-[22]. A multiple Tabu Search algorithm was developed by [23] to solve a dynamic economic generator dispatch problem. The multiple Tabu Search algorithm used additional strategies for initialisation, carried out adaptive and multiple searches, crossover and restarts. The performance of the Tabu Search was compared with that of simulated annealing, a genetic algorithm and particle swarm optimisation, in solving the problem. A higher quality solution was obtained, with better computational efficiency using the multiple Tabu Search algorithm.

2. Statement of the Optimisation of WET

2.1. Outline of the Problem

The optimisation problem consists in dimensioning the WET for a given manure input in a given time period $m \in M$. M is a set of the number of months in the multi-period dimensioning problem. The WET under study is shown in **Figure 1**, for a farm with n_{herd} cattle. Dimensioning is carried out with an adapted monthly setup, for: the backup propane flow rate, u_1^m, the split of biogas between the internal combustion engine (ICE) and the boiler, u_2^m and the volume flow rate of manure from the lagoon, u_3^m.

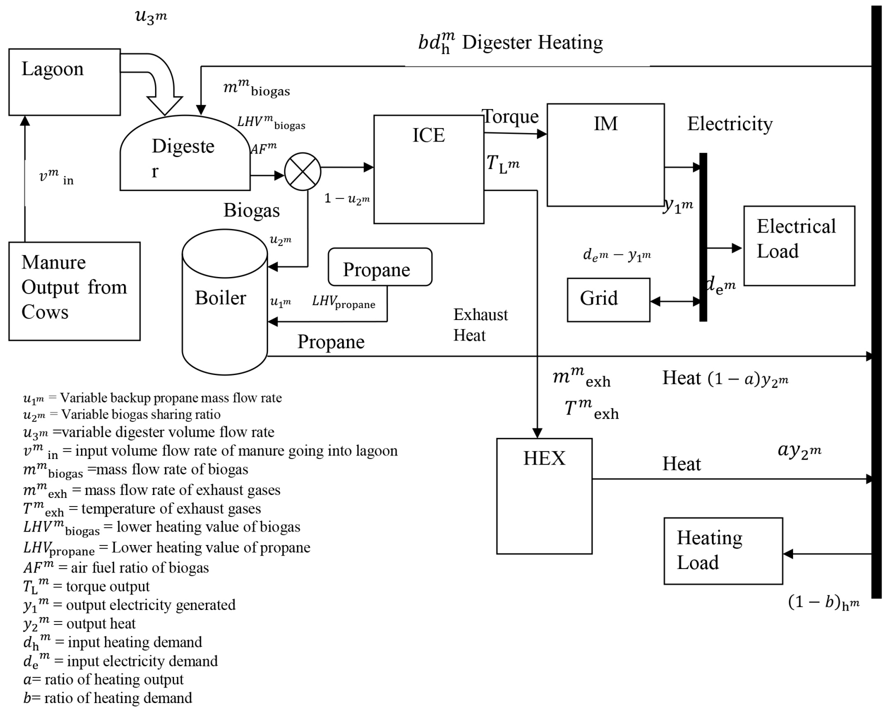


Figure 1. Waste to energy technology.

This is subject to the constraint of operating the WET such that the electricity and heating demands of the farm and the digester are met, while maximising revenue from the system. Manure from the livestock at a volume flow rate v_{in}^m in goes into a lagoon, where it is stored. The manure from the lagoon is fed to a digester at a volume flow rate, u_3^m. In the digester, the manure undergoes anaerobic digestion to produce biogas at a mass flow rate, m_{biogas}^m, air-fuel ratio, AF^m and lower heating value, LHV_{biogas}^m. The biogas produced is to be shared between an internal combustion engine and a boiler, at a ratio determined by the variable u_2^m. The mass flow rate of biogas going into the internal combustion engine is $\left(1-u_2^m\right) m_{\text{biogas}}$ and that going into the boiler is $m_{\text{biogas}} u_2^m$.

The biogas is combusted in the internal combustion engine generating a torque T_L^m. The torque T_L^m is applied to an induction machine (IM) to generate electricity, output y_1^m. The electricity is used by the farm to meet the electricity load

d_e^m. If excess electricity is produced by the WET it is sent to the electricity grid. The electricity sent to the grid is designated by $d_e^m - y_1^m$. If the electricity generated by the WET is insufficient to meet the demand of the farm, electricity is obtained from the grid and is designated by $d_e^m - y_1^m$. Combustion of biogas in the internal combustion engine produces exhaust gases at a mass flow rate and temperature denoted by m_{exh}^m and T_{exh}^m respectively. Heat from the exhaustgases is captured by the heat exchanger (HEX) and forms the heat output ay_2^m. The biogas that goes into the boiler is combusted to generate heat, denoted by $(1-a)y_2^m$. The total heat output y_2^m has to meet the heating demand of both the digester bd_h^m and the farm $(1-b)bd_h^m$. The heating demand of the digester is calculated taking into consideration the heat losses from the walls, floor and roof of the digester, and the heat required to raise the temperature of the influent manure to the digester's operating temperature. When the boiler does not generate enough heat to meet the total heating load, propane will also be combusted in the boiler. The propane is supplied as a backup fuel from a propane tank, at a mass flow rate u_1^m and lower heating value $LHV_{propane}$.

The optimisation of the WET described is done with the objective of maximizing revenue. The optimisation problem is expressed as a cost minimisation problem by:

$$\min \quad f^{cost}\left(u_1^m, u_2^m, u_3^m\right) \quad \text{for a given manure input} \quad v_{in}, \tag{3}$$

$$\text{subject to:} \quad C_{WET}\left(u_1^m, u_2^m, u_3^m\right) \le 0 \quad \text{for} \quad m \in M, \tag{4}$$

$$\text{such that:} \quad u_1^m \in \{0, 0.0001, 0.0002, 0.0003, \cdots, 0.0036\} \quad \text{for} \quad m \in M, \tag{5}$$

$$u_2^m \in \{0, 0.01, 0.02, 0.03, \cdots, 0.099\} \quad \text{for} \quad m \in M, \tag{6}$$

$$u_3^m \in \{1, 2, 3, \cdots, 59\} \quad \text{for} \quad m \in M, \tag{7}$$

$$u = \left(u_1^1, u_2^1, u_3^1, u_1^2, u_2^2, u_3^2, \cdots, u_1^{|m|}, u_2^{|m|}, u_3^{|m|}\right) \quad \text{for} \quad m \in M, \tag{8}$$

Where u_1^m, u_2^m and u_3^m are the v1ariables: backup propane mass flow rate, biogas sharing ratio and volume flow rate of manure going into the digester respectively. C_{WET} denotes a set of global constraints, some of which are linear and others non-linear. u denotes the solution of the optimisation problem as described in the Tabu Search (see Algorithm 2).

Algorithm 2 Optimisation of a WET

Initialization

1. Inputs: n_{herd}, u_1^m, u_2^m, u_3^m for $m \in M$
2. Initialize parameters: $V_{lagoon}^0, a, b, \eta_{THEX}, \eta_{boiler}, T_{water}, LHV_{propane}$
3. **for** $m \in M$
4. Build an initial solution (u_1^m, u_2^m, u_3^m) for $m \in M$
   Calculate the outputs of the WET model components
5. $\left(V_{lagoon}^m, u_3^m\right) = \text{LAGOON}\left(v_{in}^m, V_{lagoon}^{m-1}, n_{herd}\right)$
   $\left(AF^m, LHV_{biogas}^m, m_{biogas}^m\right) = \text{DIGESTER}\left(u_3^m, bd_h^m\right)$

Continued

5.
$$\left(T_{\rm L}^m, m_{\rm exh}^m, T_{\rm exh}^m, cp_{\rm exh}^m\right) = {\rm ICE}\left(m_{\rm biogas}^m, \left(1-u_2^m\right), AF^m, LHV_{\rm biogas}^m\right)$$
$$y_1^m = {\rm IM}\left(T_{\rm L}^m\right)$$
$$ay_2^m = \eta_{\rm HEX} m_{\rm exh}^m cp_{\rm exh}^m \left(T_{\rm exh}^m - T_{\rm water}\right)$$
$$\left(1-a\right) y_2^m = \left(LHV_{\rm propane} u_1^m + m_{\rm biogas}^m LHV_{\rm biogas}^m u_2^m\right)\eta_{\rm boiler}$$

6. **end for**
7. Evaluate the objective function $f^{\rm cost}$
   Tabu Search Optimisation
8. iter ← 0
9. **while** iter ≤ max_iter **do**
10. Perform Tabu Search which includes evaluation of each of the WET model components
11. Evaluate iterative solutions and update the incumbent solutions accordingly
12. **end while**

2.2. Optimisation Process Flow

Algorithm 2 describes the process flow of the optimisation. The inputs of the WET are: herd size $n_{\rm herd}$, electricity demand $d_{\rm e}^m$, heating demand $d_{\rm h}^m$ and volume flow rate of manure from the cattle $v_{\rm in}^m$. These inputs are specified for each time period, $m \in M$. The parameters of the optimisation are initialised, i.e., $V_{\rm lagoon}^o$, volume of manure in the lagoon, a, ratio of heating output, b, ratio of heating demand, $\eta_{\rm HEX}$, efficiency of the heat exchanger, $\eta_{\rm boiler}$, efficiency of the boiler, $T_{\rm water}$, water temperature and $LHV_{\rm propane}$, lower heating value of propane. An initial solution (u_1^m, u_2^m, u_3^m) is built for each of the time periods $m \in M$. This is done by calculating the outputs of the manure storage and the energy conversion processes in each component of the WET, using the functions: LAGOON, DIGESTER, ICE, IM, and the linear equations of the heat exchanger and the boiler. The function LAGOON is linear and calculates the storage of manure from the livestock, for each of the time periods $m \in M$. The functions DIGESTER, ICE and IM include complex non-linear differential equations and are represented as component models in the WET optimization problem. Each of the component models of the functions DIGESTER, ICE and IM model a difficult nonlinear optimisation problem. A variable that determines the output of the energy conversion processes in each of these component models is selected to define the solution (u_1^m, u_2^m, u_3^m), as shown in Algorithm 2. As such the non-linear optimization problems of the component models are solved by optimisation of the WET, with the solution (u_1^m, u_2^m, u_3^m). The inputs and outputs of the component models and equations are defined in **Table 1**. The electricity and heat outputs, y_1^m and, y_2^m, respectively, are obtained and used in computation of the objective function. Once an initial solution has been found and the objective function computed, the Tabu Search optimisation is carried out to determine the near optimal solutions.

Table 1. Inputs and outputs of the model components.

Input/Output	Description
n_{herd}	herd size
d_{e}^{m}	electrical demand of the farm
d_{h}^{m}	heat demand
v_{in}^{m}	volume flow rate of the manure from the livestock
V_{lagoon}^{m}	volume of the manure in the lagoon
m_{biogas}^{m}	mass flow rate of the biogas from the digester
m_{exh}^{m}	mass flow rate of the exhaust gases
T_{exh}^{m}	temperature of the exhaust gases
cp_{exh}	specific heat capacity of the exhaust gases
AF^{m}	air-fuel ratio of the biogas
LHV_{biogas}^{m}	Lower Heating Value of the biogas
T_{L}^{m}	output torque of the internal combustion engine
y_{1}^{m}	electricity output
y_{2}^{m}	heat output

3. Description of the Optimisation of WET

The optimisation problem involves evaluation of the biogas production and electricity and heat production from the volume flow rate of manure, v_{in}^{m} for $m \in M$. Starting with v_{in}^{m}, the inputs and outputs of the WET components are calculated in turn using the functions, LAGOON, DIGESTER, ICE, IM and the linear equations of the boiler and the heat exchanger. The functions of the respective WET components are indicated in Figure 1, together with the inputs and outputs. This section describes the objective function and the constraints of the optimisation, followed by an outline of the process flow of the optimisation problem.

3.1. Objective Function

The formulation of the optimisation problem maximises revenue from a WET subject to meeting the heating demand of the farm and the digester. The objective function has four components; the cost of capital, C_{capital}^{m}, the cost of propane, C_{propane}^{m}, the cost of incentives, $C_{\text{incentives}}^{m}$ and the cost of grid electricity, $C_{\text{grid\_electricity}}^{m}$ for $m \in M$. The following is a description of the components of the objective function.

3.1.1. Cost of Capital

The cost of capital C_{capital}^{m} is calculated from the capital expenditure on the digester, lagoon, boiler and engine-generator set. The capital expenditure on these items is dependent on their sizes, which in turn depends on the herd size. The size of the digester and the lagoon are dependent on the volume flow rate of manure from the livestock, v_{in}^{m}. The cost of the boiler and engine-generator set are dependent on the ratings of the respective equipment. This capital expenditure is amortized monthly to obtain the cost of capital C_{capital}^{m}. The cost of capital is calculated using the non-linear function (9).

$$C_{\text{capital}}^{m} = \text{CAPITAL}\left(\text{HRT}, c_{\text{digester}}, c_{\text{lagoon}}, P_{\text{rated}}, c_{\text{engine}}, c_{\text{boiler}}, C_{\text{cap\_incentive}}, i_{\text{rate}}, n_{\text{period}}, v_{\text{in}}^{m}, V_{\text{lagoon\_storage}}, d_{\text{h}}^{m}, ay_{2}^{m}\right) \text{ for } m \in M. \tag{9}$$

where CAPITAL is the function for calculation of the cost of capital, v_{in}^{m} is the volume flow rate of manure from the livestock, HRT is the hydraulic retention time, c_{digester} is the cost of the digester, $V_{\text{lagoon\_storage}}$ is the storage capacity of the lagoon, c_{lagoon} is the unit cost of the lagoon, P_{rated} is the power rating of the induction machine, c_{engine} is the cost of the engine-generator set, d_{h}^{m} is the heating load, a is the ratio of heat output from the heat exchanger, y_{2}^{m} is the heat output, c_{boiler} is the cost of the boiler, $C_{\text{cap\_incentive}}$ is the capacity incentive, i_{rate} is the interest rate and n_{period} is the number of periods over which the interest is charged.

3.1.2. Cost of Propane

The monthly cost of propane, C_{propane}^{m} is a linear function of the backup propane mass flow rate, u_{1}^{m} and is given by (10).

$$C_{\text{proppane}}^{m} = \text{PROPANE}\left(c_{\text{propane}}, u_{1}^{m}\right) \text{ for } m \in M, \tag{10}$$

where PROPANE is the function for calculating the cost of propane, c_{propane} is the unit cost of propane and u_{1}^{m} is the backup propane mass flow rate.

3.1.3. Cost of Incentives

A performance incentive is given for generation of renewable energy. This incentive is included in the objective function and is calculated by a linear function (11).

$$C_{\text{incentives}}^{m} = \text{INCENTIVES}\left(c_{\text{incentives}}, y_{1}^{m}\right) \text{ for } m \in M, \tag{11}$$

where $C_{\text{incentives}}^{m}$ is the cost of incentives, INCENTIVES is the function for calculating the cost of incentives, y_{1}^{m} is the electricity output and $c_{\text{incentives}}$ is the unit cost of incentives.

3.1.4. Cost of Grid Electricity

The cost of grid electricity, $C_{\text{grid\_electricity}}^{m}$ is a non-linear function of the electricity output, y_{1}^{m} (12).

$$C_{\text{grid\_electricity}}^{m} = \text{GRID\_ELECTRICTY}\left(c_{\text{tariff}}, d_{e}^{m}, y_{1}^{m}\right) \text{ for } m \in M, \tag{12}$$

The four cost components of the objective function form a multi-objective optimization problem. With the Tabu Search method used, sampling of the neighbourhood results in many solutions. Each of these solutions is to be evaluated using the multiobjective function. The incumbent solution is to be selected as the one with the minimum overall cost. In determination of a solution that will minimise the overall objective, an easy way is to compute the overall cost as:

$$z = \sum_{m=1}^{M}\left(C_{\text{capital}}^{m} + C_{\text{proppane}}^{m} - C_{\text{incentives}}^{m} + C_{\text{grid\_electricity}}^{m}\right) \text{ for } m \in M, \tag{13}$$

The drawback of (13) is the different ranges of the values of the cost components. This means that the overall objective will largely be minimising the cost components with the highest value. This can be overcome by the use of weights, but it is difficult to find the proper weights. A better method is to express the objective function as a cost vector of the components, resulting in a cost vector for each of the solutions.

Let

$$f_k^{\rightarrow\text{cost}} = \left[\sum_{m=1}^{M} C_{\text{capital}}^{m}, \sum_{m=1}^{M} C_{\text{propane}}^{m}, -\sum_{m=1}^{M} C_{\text{incentives}}^{m}, \sum_{m=1}^{M} C_{\text{grid\_electricity}}^{m}\right], \text{ or } k \in K \text{ and } m \in M \quad (14)$$

be the set of solutions. The individual cost components of the solution vectors are compared for dominance. The vectors with the non-dominant cost components form a Pareto incumbent front. The solutions on the Pareto incumbent front are selected as the incumbent solutions. There are several incumbent solutions, all of which are retained, as shown in Figure 2 for the comparison of the cost of propane and the cost of grid electricity.

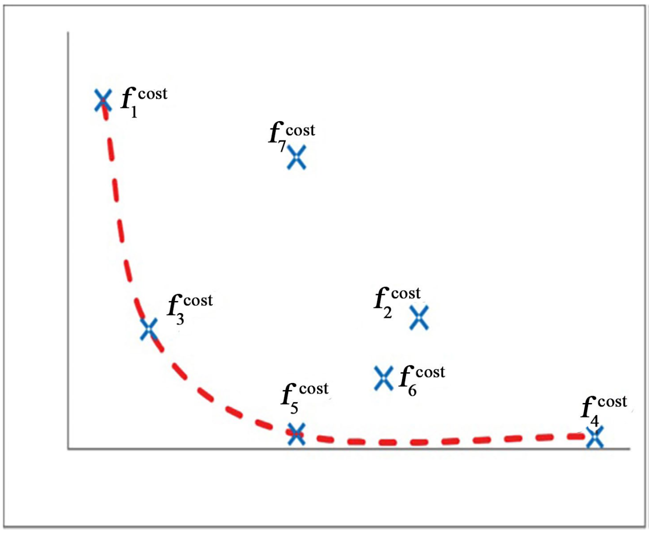


Figure 2. Illustration of pareto incumbent front.

3.2. Global Constraints

This section describes the global constraints, C_{WET}, and how they are derived from the optimisation problem. The initial solution described in Algorithm 2 satisfies the global constraints. In the Tabu Search optimisation that follows, the solution (u_1^m, u_2^m, u_3^m) has to be checked for satisfaction of the global constraints. These constraints are defined as:

$$0 \le \left(v_{\text{in}}^{m} n_{\text{days}}^{m} + V_{\text{lagoon\_manure}}^{m-1} - n_{\text{days}}^{m} u_3^{m}\right) \le V_{\text{lagoon\_storage}} v_{\text{in}}^{m}, \quad (15)$$

$$\left(V_{\text{D}} - \text{HRT} u_3^{m}\right) \ge 0, \quad (16)$$

$$\left(P_{\text{rated}} / \omega_{\text{mech}} - \text{ICE}\left(LHV_{\text{biogas}}^{m}, \omega_{\text{mech}}, \left(1 - u_2^{m}\right) m_{\text{biogas}}^{m}\right)\right) \ge 0, \quad (17)$$

$$d_{\text{h}}^{m} \le \left(\eta_{\text{HEX}} m_{\text{exh}}^{m} cp_{\text{exh}} \left(T_{\text{exh}}^{m} - T_{\text{water}}\right) + \left(LHV_{\text{propane}} u_1^{m} + m_{\text{biogas}}^{m} LHV_{\text{biogas}}^{m} u_2^{m}\right)\right) \le \left(d_{\text{h}}^{m} + \delta_h\right), \quad (18)$$

$$\left(b_r - d_{\text{h}}^{m} + \eta_{\text{HEX}} m_{\text{exh}}^{m} cp_{\text{exh}} \left(T_{\text{exh}}^{m} - T_{\text{water}}\right)\right) \le 0, \quad \text{for} \quad m \in M \quad (19)$$

where v_{in}^{m} is the volume flow rate of manure from the cattle, n_{days}^{m} are the number of days, $V_{\text{lagoon\_manure}}^{m-1}$ is the volume of manure in the lagoon, u_3^m is the volume flow rate of manure from the lagoon, $V_{\text{lagoon\_storage}}$ is the storage capacity of the lagoon, V_{D} is the volume of the digester, HRT is the hydraulic retention time of the digester, P_{rated} is the power rating of the induction machine, ω_{mech} is the speed of the internal combustion engine, ICE is the function for evaluation of the torque output of the internal combustion engine, LHV_{biogas}^{m} is the lower heating value of biogas, u_2^m is the biogas sharing ratio, m_{biogas}^{m} is the mass flow rate of biogas, d_{h}^{m} is the heating demand, η_{HEX} is the efficiency of the heat exchanger, m_{biogas}^{m} is the mass flow rate of the exhaust gases, cp_{exh} is the specific heat capacity of the exhaust gases, T_{exh}^{m} is the temperature of the exhaust gases, T_{water} is the temperature of water, u_1^m is the mass flow rate of backup propane, LHV_{Propane}^{m} is the lower heating value of propane, η_{boiler} is the efficiency of the boiler, δ_{h} is an allowance for the heating constraint and b_{r} is the boiler rating. The manure from the livestock is stored in a lagoon with a storage capacity of $V_{\text{lagoon\_storage}}$ days. The volume flow rate of manure from the lagoon into the digester, u_3^m is varied to minimise the cost of the system. Constraint (15) is set to ensure that the net volume of manure in the lagoon is not negative. In a given month m, the volume of manure that goes into the lagoon $u_{\text{day}}^{m}u_3^m$, should not be greater than the sum of the volume of the manure that was in the lagoon the previous month $V_{\text{lagoon\_manure}}^{m-1}$, and the volume of manure from the cattle in month m. Constraint (15) also ensures that the volume of manure in the lagoon is not greater than the storage capacity of the lagoon. Constraint (17) is set to ensure that the volume of manure in the digester, $\text{HRT}u_3^m$, is not greater than the volume of the digester V_{D}. The digester is modeled using non-linear differential equations. The digester model is treated as a black box for purposes of optimisation. The differential equations in the black box, DIGESTER, used to calculate the mass flow rate, m_{biogas}^{m}, the air-fuel ratio, AF^m and the lower heating value of biogas, LHV_{biogas}^{m} can be found in [24]. The output torque of the internal combustion engine is determined by applying the Newton-Raphson method to a two dimensional linear interpolation function. The linear interpolation function is multiplied by the available torque. The available torque is calculated from the mass flow rate of biogas to the internal combustion engine, the lower heating value of biogas, and the speed of the internal combustion engine. The internal combustion engine model is also treated as a black box of these functions (ICE). The details of the modeling of the internal combustion engine can be found in [25]. The internal combustion engine is coupled with an induction machine of rating, P_{rated}, that generates electric power. The induction machine is modeled using non-linear differential equations detailed in [26]. The induction machine is also treated as a black box, IM, with the input as torque and the output as electricity, y_1^m. The electricity generated is a function of the torque, which in turn is a function of the mass flow rate of biogas to the internal combustion engine. Constraint (18) is therefore set to limit the mass flow rate of biogas to not more than what is required to generate rated power, P_{rated} in the induction machine.

Sometimes the biogas generated by the digester may be insufficient for sharing between the internal combustion engine and the boiler. Priority is then given to the combustion of biogas in the internal combustion engine, and propane is combusted in the boiler. This is done to generate electricity that can be sold to the utility. The revenue from the sale of electricity to the utility will be greater than the cost savings from the avoided use of propane for heating. A propane tank that supplies propane at a mass flow rate, u_1^m is therefore included in the WET. The heat produced by the boiler is calculated from the mass flow rate of biogas to the boiler, $m_{\text{biogas}}^m u_2$, the mass flow rate of propane, u_1^m, the lower heating value of propane and the lower heating value of biogas. Exhaust heat is also produced as a result of the combustion process in the internal combustion engine. This exhaust heat is captured by the heat exchanger. Constraint (19) is set to ensure that the heat output of the WET meets the heating demand of the farm and the digester. Constraint (6.20) is set to ensure that the heat to be generated by the boiler is not greater than the boiler rating, b_r. The contribution of the heat captured by the heat exchanger is subtracted from the heat output of the boiler in formulation of Constraint (20).

Infeasible solutions arise if the constraints are not met. The measure of infeasibility of the solution is calculated as:

$$f^{\text{infeas}} = \sum_{m=1}^{M} \left(S_{\text{lagoon\_volume}}^m + S_{\text{digester\_size}}^m + S_{\text{mbiogas}}^m + S_{\text{heating\_demand}}^m + S_{\text{boiler\_rating}}^m \right), \quad \text{for} \quad m \in M \quad (20)$$

where f^{infeas} is the total measure of infeasibility, $S_{\text{lagoon\_volume}}^m$, $S_{\text{digester\_size}}^m$, S_{mbiogas}^m, $S_{\text{heating\_demand}}^m$ and $S_{\text{boiler\_rating}}^m$ are the measures of infeasibility of the volume of manure in the lagoon, the digester size, the mass flow rate of biogas to the engine-generator set, the total heat output and the boiler rating, respectively. The measures of infeasibility are derived from the respective Constraints (15), (17), (18), (19) and (20).

Using the measure of infeasibility of the volume of manure in the lagoon as an example:

$$n_{\text{days}}^m u_3^m + S_{\text{lagoon\_volume}}^m = V_{\text{lagoon\_manure}}^{m-1} - V_{\text{lagoon\_manure}} v_{\text{in}}^m + v_{\text{in}}^m n_{\text{days}}^m \quad \text{for} \quad m \in M, \quad (21)$$

The solution is feasible for $S_{\text{lagoon\_volume}}^m \geq 0$.

where n_{days}^m are the number of days, u_3^m is the volume flow rate of manure from the lagoon, $S_{\text{lagoon\_volume}}^m$, is the measure of infeasibility of the volume of manure in the lagoon, $V_{\text{lagoon\_manure}}^{m-1}$ is the volume of manure in the lagoon, $V_{\text{lagoon\_manure}}$ is the storage capacity of the lagoon and v_{in}^m is the volume flow rate of manure from the livestock. The other measures of infeasibility are defined similarly.

4. Description of the Tabu Search Algorithm

This section describes the adaptations of the Tabu Search algorithm developed for optimisation of a WET. The Tabu Search is described in Algorithm 3. The notation and the parameters of the Tabu Search are given in **Tables 2-3** respectively.

4.1. Basic Tabu Search Algorithm

The basic Tabu Search defines a neighbourhood of moves that can be applied to the solution, keeps a list of the forbidden moves (Tabu list) and incorporates a stopping condition. These aspects of the basic Tabu Search included in the optimisation of the WET are discussed in this section.

4.1.1. Definition of the Neighbourhood

The neighbourhood of: u_i^m is defined as

$$N\left(u_i^m\right)^{new} = \begin{Bmatrix} v : v = u_i^m + \delta_i & i = 1,2,3 \\ v = u_i^m - \delta_i & m \in M \\ & u \in U^{MODEL} \cup U^{GLOBAL} \end{Bmatrix}$$

$$LB_v \le v \le UB_v : v \in N\left(u_i^m\right),$$

where u_i^m is the optimisation variable, U^{MODEL} is the set of constraints to be satisfied by the WET black box models, U^{GLOBAL} is the set of global constraints to be satisfied by the optimisation, LB_v is the lower bound of the neighbourhood and UB_v is the upper bound of the neighbourhood. The move from u_i^m to $u_i^m \pm \delta_i$ is selected within the specific limits and step sizes for the different variables.

4.1.2. Tabu List

A Tabu list is formulated from moves that result in the current solution. Each entry of the Tabu list is a vector of the move from u_i^m to $u_i^m \pm \delta_i$, and its associated month. Reverse moves are also included in the Tabu list. The Tabu list includes a random number $N^{\text{TL\_length}}$, selected within a given interval that decides for how many iterations a Tabu condition persists.

4.1.3. Stopping Condition

The stopping condition of the Tabu Search algorithm is set to termination of the optimisation, if no improvement in the incumbent solution has been observed after max_iter iterations, following the application of diversification.

Table 2. Tabu search notation.

Input/Output	Description
$u_i^{m,\text{init}}$	initial solution
u_i^m	current solution
S^{best}	set of the Pareto incumbent solutions
S^{current}	set of the Pareto current solutions u_i^m
$N\left(u_i^m\right)$	neighbourhood of variable
LB_v	lower bound of neighbourhood
UB_v	upper bound of neighbourhood
T^{list}	Tabu list
U^{MODEL}	set of constraints to be satisfied by the WET black box models
U^{GLOBAL}	set of global constraints to be satisfied by the optimization

4.2. Adaptations of the Tabu Search

Four aspects of the Tabu Search have been developed for adaptation to the problem being solved. These are: use of the Pareto optimal front method to evaluate the multi-period and multi-objective function, constraints handling, the multi-period optimisation strategy and the diversification strategy. The adaptations made to the basic Tabu Search algorithm in the context of the Waste-to-Energy Technology (WET) optimisation problem significantly enhance the algorithm's ability to deal with the problem's inherent complexity. The WET problem is characterised by multiple objectives, multi-period variables, non-linear models treated as black boxes, and strict constraints that can conflict with cost reduction goals. The motivation behind each adaptation is rooted in addressing these specific challenges, making the modified Tabu Search more robust, flexible, and effective than the standard form. This section describes the adaptations developed.

4.2.1. Pareto Incumbent Solutions

In the WET problem, multiple objectives must be optimised—primarily cost minimisation and constraint (infeasibility) reduction—over a series of time periods (months). Standard TS typically handles single-objective optimisation and lacks a mechanism to evaluate trade-offs among conflicting objectives or time-variant effects.

During the Tabu Search optimisation a different variable is optimised for each time period, $m \in M$, for as long as the current solution is improving. This implies that only the cost components of the period for which the optimisation is carried out are modified, each time the objective function is evaluated. In order not to lose the benefit of the modified cost components, they are summed separately for all the periods to form the cost vector (15). The cost vectors are then checked for non-dominance and the non-dominated solutions form a Pareto incumbent front. Summing the cost components separately over all the periods, *M*, incorporates the multi-period nature of the optimisation into the evaluation of the objective function. The Pareto front method of evaluating multi-objective functions has therefore been modified to incorporate the multi-period nature of the optimisation problem.

Standard TS retains a single best solution; this adaptation maintains a *front* of non-dominated solutions, enabling multi-objective, multi-period optimisation and better long-term decision-making.

Algorithm 3 Tabu Search	
	Initialization
1.	Build a feasible initial solution $u_i^{m,\text{init}}$
2.	Set $u_i^m \leftarrow u_i^{m,\text{init}}, S^{\text{best}} \leftarrow \left\{u_i^{m,\text{init}}, m \in M, i = 1,2,3\right\}$
3.	Initialize the Tabu list: $T^{\text{list}} \leftarrow \varnothing$
4.	Set the bounds
5.	Evaluate $\vec{f}_{\min}^{\text{cost}}$, $\vec{f}_{\min}^{\text{infeas}}$
	Tabu Search
6.	iter ← 0

Continued

7.	**while** iter ≤ max iter **do**
8.	**while** iter ≤ max_iter_div **do**
9.	*Phase 1: Minimize Cost*
10.	iter_opt ← 0
11.	**while** iter opt ≤ max iter opt /*Attempt at finding a solution with a smaller cost regardless of the infeasibility*/**do**
12.	Perform a round robin search on the months : For a given month $m \in M$, select one variable with index $i(m)$: $i(m) = i(m-1) + 1 \pmod 3$
13.	Update the neighbourhood of the selected variable
14.	Evaluate all solutions $u_i^{m'}$ in $N\left(u_i^m\right)$ with respect to $\vec{f}_{\min}^{\text{cost}}$, $\vec{f}_{\min}^{\text{infeas}}$ (only for storage with solution)
15.	$S^{\text{current}} \leftarrow \arg\min_{u'} \vec{f}_{\min}^{\text{cost}}\left(u'\right)$ for $u = u_1^1, u_2^1, u_3^1, u_1^2, u_2^2, u_3^2, \cdots, u_1^{\lvert m\rvert}, u_2^{\lvert m\rvert}, u_3^{\lvert m\rvert}$ and $m \in M$
16.	**end while**
17.	iter_feas ← 0
18.	*Phase 2: Minimize Infeasibility*
19.	**while** iter_feas ≤ max_iter_feas /* Reducing infeasibility*/**do**
20.	Select the month $m \in M$ for which the search is to be carried out : $m \leftarrow \arg\max_{m'} \vec{f}_{\min}^{\text{infeas}}\left(u'\right)$ for $u = \left(u = u_1^1, u_2^1, u_3^1, u_1^2, u_2^2, u_3^2, \cdots, u_1^{\lvert m\rvert}, u_2^{\lvert m\rvert}, u_3^{\lvert m\rvert}\right)$
21.	Update the neighbourhood of the selected variable
22.	Evaluate all solutions $u_1^{m'}$ in $N\, u_i^m$ with respect to $\vec{f}_{\min}^{\text{cost}}$, $\vec{f}_{\min}^{\text{infeas}}$ (only for storage with solution)
23.	$S^{\text{current}} \leftarrow \arg\min_{u'} \vec{f}_{\min}^{\text{cost}}\left(u'\right)$ for $u = u_1^1, u_2^1, u_3^1, u_1^2, u_2^2, u_3^2, \cdots, u_1^{\lvert m\rvert}, u_2^{\lvert m\rvert}, u_3^{\lvert m\rvert}$ and $m \in M$
24.	**end while**
25.	**end while**
26.	Apply diversification
27.	**end while**

Table 3. Parameters of the Tabu search.

Parameter	Description	Value
$V_{\text{lagoon\_storage}}$	storage capacity of the lagoon (days)	35
HRT	hydraulic retention time (days)	20
n_{herd}	number of livestock	500 cows
n_{day}^m	number of days in a month	Varies
P_{rated}	rating of the induction machine (hp)	150
LHV_{propane}	lower heating value of propane (kJ/kg)	46,300 [27]
T_{water}	water temperature (˚C)	35
η_{HEX}	heat exchanger efficiency (%)	70
η_{rated}	boiler efficiency (%)	70
c_{lagoon}	unit cost of lagoon (USD/m3)	2.47 [28]
c_{propane}	unit cost of propane (USD/m+)	1.98 [29]

Continued

$n^{\text{rand\_div}}$	consecutive random moves (diversification Strategy D1)	5
$n^{\text{nonimprov\_div}}$	consecutive non-improving moves to apply diversification	5
$n^{\text{restart\_div}}$	restarts with incumbent solution (diversification Strategy D2)	3
max_iter_div	number of iterations for application of diversification	100
δ_{h}	allowance for heat demand constraint (kW)	10
max_iter	number of iterations for the stopping condition	150
max_iter_opt	number of iterations for the minimisation of cost	50
max_iter_feas	number of iterations for the minimisation of infeasibility	25
max_iter_div	number of iterations for the application of diversification	100
S^{infeas}	threshold of infeasibility	Varies
S_0^{infeas}	initial threshold of infeasibility	varies

4.2.2. Method of Handling Constraints

The WET system includes global constraints (U^{GLOBAL}) such as energy balance or heat demand that must be met for a feasible solution. However, strict feasibility throughout the search can prevent reaching low-cost areas.

There are two sets of constraints in the optimisation problem of the WET. U^{MODEL} is the set of constraints to be satisfied by the models of the WET and U^{GLOBAL} is the set of global constraints to be satisfied by the solution of the optimisation problem, i.e., CWET (u_1^m, u_2^m, u_3^m) (4). The set of constraints to be satisfied by the models of the WET, U^{MODEL} is not defined because the WET models are treated as black boxes in the optimization problem. The method of handling constraints discussed applies to the set of global constraints, U^{GLOBAL}. Infeasible solutions result if the global constraints are not satisfied. Infeasible solutions are allowed in the Tabu Search optimisation in order to allow the search to move to low cost regions during the minimisation of cost. To ensure that the search goes back to a feasible region, a second objective function is introduced. The second objective function minimises infeasibility (20). The Tabu Search optimisation alternates between minimising cost (Phase 1) and minimising infeasibility (Phase 2). Thresholds are set for the extent to which infeasibility is allowed. These thresholds are progressively reduced during the course of the optimisation.

Traditional TS avoids infeasible regions or uses penalty functions. This approach intentionally explores infeasible space and uses two separate objectives to alternate exploration and correction, improving convergence in constraint-heavy scenarios.

4.2.3. Multi-period Optimisation Strategy

WET operates over time (months), with different variables (e.g., feed rates, storage levels, energy outputs) for each period. Standard TS does not inherently support temporal decomposition or sequential optimisation.

A multi-period optimisation strategy is developed to ensure a smooth transition from one period to the next during optimisation. Different strategies are used for the phase for minimisation of cost (Phase 1) and minimisation of infeasibility (Phase 2). The period is measured in months. The variables are optimised for each month. During the phase for minimisation of cost, optimisation is done based on a round robin strategy of the months, starting with the month of January. If a solution is encountered that is worse than the current solution, another variable is selected for optimisation, in the same month. If all three variables do not result in an improved solution, the current solution is not updated. This is repeated for the twelve months period. If the current solution does not improve over this 12 months period, it is updated with the least non-improving solution. The optimisation strategy during the phase for minimisation of infeasibility is such that the month with the most infeasible solution is selected for optimisation. This is in contrast to the phase of minimisation of cost, where the round robin method is used. Once a feasible solution is encountered during the phase of minimisation of infeasibility, the strategy reverts to minimisation of cost.

This strategy introduces temporal awareness into the optimisation, allowing TS to behave more intelligently across time-dependent variables—something not considered in basic TS.

4.2.4. Diversification

Tabu Search is prone to getting trapped in local optima, especially in large, constrained, or multi-modal search spaces like those in WET. Diversification is essential to escape these regions.

If the incumbent solution does not improve for max iter div iterations, diversification is applied. Diversification is applied by performing three consecutive restarts with the incumbent solution. For each restart performed, a different variable is selected for optimisation. Diversification is only applied if after the max_iter_div iteration, the current solution does not improve for $n^{\text{nonimprov\_div}}$ consecutive iterations. The Tabu list is emptied on performing each of the restarts.

Standard TS relies on diversification through long-term memory (e.g., frequency-based strategies), which may not be sufficient for highly constrained problems. This explicit restart mechanism with structured randomness ensures stronger exploration and reduces premature convergence. summary of key improvements of the tabu search is presented in **Table 4**.

Table 4. Adaptations of the Tabu search.

Adaptation	Motivation	Benefit over standard TS
Pareto Front for multi-objective	Handle trade-offs in cost/infeasibility across periods	Retains multiple diverse, non-dominated solutions
Dual objective strategy	Navigate feasibility/cost conflict	Enables escape into low-cost regions and guided return to feasibility
Multi-period Round-robin + feas-targeting	Reflect time dimension and focus effort effectively	Improves search balance and precision across temporal phases
Diversification by incumbent restarts	Escape local optima in black-box models	Facilitates broader exploration with structured randomness

These enhancements make the Tabu Search algorithm better suited for the real-world complexity, dynamic constraints, and trade-offs inherent in biomass WET optimisation, producing more reliable, diverse, and cost-effective solutions than the standard approach.

5. Experiments and Results

This section begins with a description of the data instances and definitions of the experiments.

5.1. Data Instances

The data instance is obtained from a dairy farm of herd size 500 cows [30].

5.2. Descriptions of the Strategies of the Tabu Search Experiments

The experiments carried out are grouped into strategies. Many strategies were tested and the most successful ones were reported. The strategies correspond to the aspects of the Tabu Search developed are defined below:

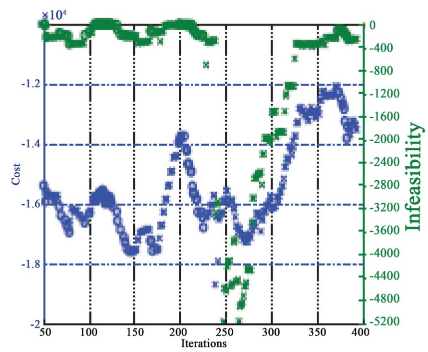
(i)	Strategy C1, the threshold of infeasibility is adjusted to handle constraints;
(ii)	Strategy C2, the number of iterations for minimisation of cost and minimization of infeasibility are varied to handle constraints;
(iii)	Strategy C3, feasible and infeasible solutions are allowed during the phase for minimisation of infeasibility;
(iv)	Strategy D1, diversification by consecutive random moves;
(v)	Strategy D2, diversification by consecutive restarts with the incumbent solution;
(vi)	Strategy MOBJ1, evaluation of Pareto incumbent solutions;
(vii)	Strategy MOBJ2, summing cost components of the objective function;
(viii)	Strategy MP1, round robin and updating current solution;
(ix)	Strategy MP2, round robin and updating solution with improving solution only;
(x)	Strategy MP3, round robin and updating solution with improving solution only, and sampling all variables in one month if required;
(xi)	Strategy MP4, round robin during the phase for minimisation of infeasibility;

Experiments with Strategies C1, C2 and C3 were developed to investigate the handling of constraints. Two diversification strategies D1 and D2 were experimented with. Experiments with Strategies MOBJ1 and MOBJ2 were developed to investigate the formation of Pareto incumbent solutions in the multi-objective and multiperiod optimisation problem. Handling of the multi-period nature of the problem was investigated in Strategies MP1, MP2, MP3 and MP4. Each of these strategies is explained in detail in the following sections.

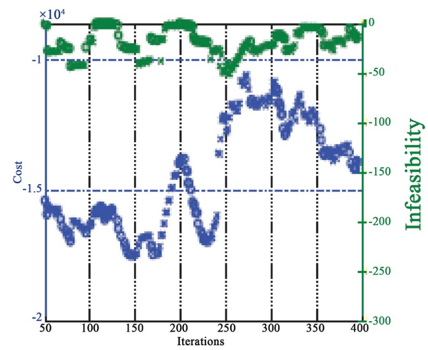
5.2.1. Constraints Handling Strategy

The aim of the experiments for constraint handling carried out in Strategies C1, C2 and C3, is to show that allowing infeasibility for a given set of parameters aids in moving towards an optimal solution faster. Two parameters are experimented with: (i) thresholds of infeasibility and (ii) number of iterations for which the cost or the infeasibility is minimised. The threshold is a value that limits the extent of infeasibility. This is required to prevent the solution from becoming too infeasible and therefore unable to return to a feasible region. In Strategy C1 the threshold of infeasibility is fixed. Three fixed thresholds are experimented with. These are S^{infeas} = −500, −200 and −100. The results of fixing the threshold of infeasibility to S^{infeas} = −200 is shown in Figure 3(a).

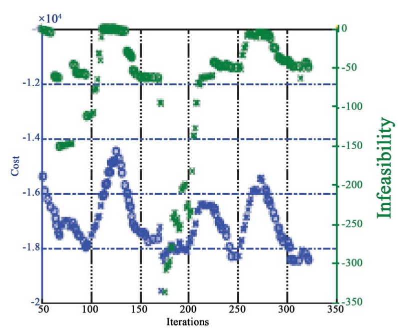
When the threshold is fixed to S^{infeas} = −500, the cost reaches low values. However these low values are in the infeasible regions. The costs of grid electricity for the current solutions are −18,278, −18,197, −19,091 and −19,949 at the 50th, 100th, 200th iterations and at termination, respectively.



(a) Fixed Threshold $S^{\text{infeas}} = -200$



(b) Varying Thresholds $S_o^{\text{infeas}} = -200$



(c) Varying Thresholds $S_O^{\text{infeas}} = -300$

Figure 3. Strategy C1.

Fixing the Threshold of infeasibility to a lower value of $S^{\text{infeas}} = -200$, gives better incumbent solutions. The iterations 100, 200 and the termination conduction have costs of grid electricity for the incumbent solutions of −16,130, −16,627, -16,730 and −16,834. The current solution also reaches relatively low values of costs of grid electricity of −17,701 and −18,302 at the 50th and 100th iterations respectively. Fixing the threshold of infeasibility to a lower value of $S^{\text{infeas}} = -100$ does not result in significantly better incumbent solutions. At $S^{\text{infeas}} = -100$, the cost of grid electricity of the incumbent solution is −16,130. This is because $S^{\text{infeas}} = -100$ is so low that it restricts the search to a local region. This is evidenced by the high values of the costs of grid electricity of the current solutions of −15,410, −13,739 and −13,022 at the 100th and 200th iterations, and at termination respectively, for the cows data instance. The respective values of infeasibility are −155, −99 and −139. The costs of grid electricity are higher than those at $S^{\text{infeas}} = -500$ and −200, at the same number of iterations. From these experiments, a good starting point for the threshold of infeasibility is identified as $S^{\text{infeas}} = -200$.

Further investigation was required on the effect of varying the threshold of infeasibility, before a conclusion could be arrived at on the suitability of the strategy of fixing the threshold. Strategy C1 therefore also included experiments where the threshold of infeasibility was varied. The initial thresholds of infeasibility were set to $S_O^{\text{infeas}} = -500, -300, -200$ and −100. The thresholds of infeasibility were varied as follows:

$$S_O^{\text{infeas}} = -500; S^{\text{infeas}} \in \{-500, -400, -300, -200, -100, -50, -40, \cdots, -10\}, \quad (22)$$

$$S_O^{\text{infeas}} = -300; S^{\text{infeas}} \in \{-300, -200, -100, -50, -40, -30, -20, -10\}, \quad (23)$$

$$S_O^{\text{infeas}} = -200; S^{\text{infeas}} \in \{-200, -150, -100, -90, -80, -70, -60, \cdots, -10\}, \quad (24)$$

$$S_O^{\text{infeas}} = -100; S^{\text{infeas}} \in \{-100, -90, -80, -70, -60, -50, -40, -30, \cdots, -10\}, \quad (25)$$

where S_O^{infeas} is the initial threshold of infeasibility and S^{infeas} is the varying threshold of infeasibility. The results of these experiments are shown in **Figures 3 (b) and 3 (c)**.

With regard to Strategy C1, a cost of grid electricity of −16,884 for the incumbent solution, is obtained, at $S_O^{\text{infeas}} = -200$. $S^{\text{infeas}} = -200$ also gave the best incumbent solution, for the experiments of fixing the threshold of infeasibility. Varying the threshold of infeasibility gives a better incumbent solution compared to fixing the threshold of infeasibility.

Figures 3 (b), show a move towards lower costs at the beginning of the iterations, for $S_O^{\text{infeas}} = -200$. As the iterations progress, the costs tend to increase. This is because of the progressive decrease in the threshold of infeasibility, leading to large decreases in infeasibility. Decreasing infeasibility has the reverse effect of increasing cost. The increasing cost means the solution is moving away from the optimal. Diversification Strategy D1 which involves making 5 consecutive random moves was being applied after 100 iterations, to move the search to a new region. This however did not impact the optimisation significantly and the incumbent solution was obtained before the 100th iteration (**Figure 3 (b)**). In order to obtain improving incumbent solutions after the 100th iteration, diversification Strategy D2 was developed and used for subsequent experiments of Strategies C2, C3, MOBJ1, MOBJ2, MP1, MP2, MP3 and MP4. In Strategy D2, a restart was made with the incumbent solution, if there was no improvement in the incumbent solution after max-iter_div iterations.

In Strategy C2, the number of iterations for the minimisation of cost and minimisation of infeasibility were varied. In the first experiment done, the same number of iterations were allowed for minimisation of cost and minimisation of infeasibility, i.e. max_iter_opt = max_iter_feas = 50. The incumbent solutions are better with max_iter_opt = 50 and max_iter_feas = 25 than with max_iter_opt = max_iter_feas = 50.

The experiments were repeated with: (i) max_iter_opt = 75 and max_iter_feas = 50, and (ii) max_iter_opt = max_iter_feas = 75. The best parameters for Strategy C2 were found to be max_iter_opt = 75 and max_iter_feas = 50 (**Figure 4**). The cost of grid electricity of the incumbent solution for max_iter_opt = 75 and max_iter_feas = 50 was −20,545, whereas that with max_iter_opt = 50 and max_iter_feas = 25 was −19,504.

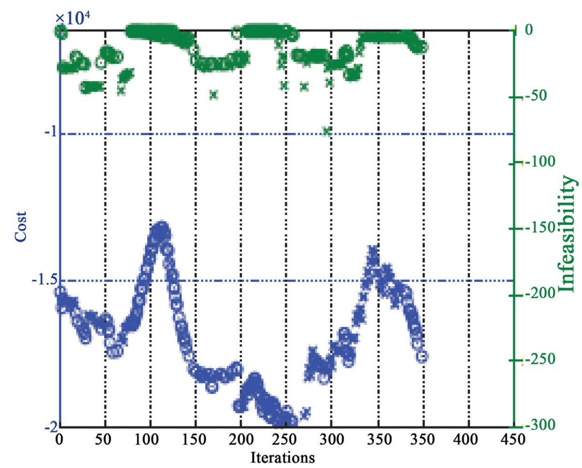


Figure 4. Strategy C2: Varying number of iterations for minimisation of cost and minimisation of infeasibility.

The handling of feasible solutions that arise during the phase of minimisation of infeasibility is investigated in Strategy C3. In this strategy, both feasible and infeasible solutions (within the threshold of infeasibility) are allowed during the phase of minimisation of infeasibility. This is compared to what is done in Strategy C2. Although Strategy C2 investigated the number of iterations for minimisation of cost and minimisation of infeasibility, it uses a different method from Strategy C3 for handling feasible solutions that arise during minimisation of infeasibility (**Figure 5**). A comparison can therefore be made between Strategy C3 and C2. In Strategy C2 only feasible solutions are allowed during the phase of minimisation of infeasibility as a first priority. If there are no feasible solutions, then infeasible solutions within the threshold of infeasibility are allowed. The results of the experiment for Strategy C3 are compared with those of Strategy C2. The cost of grid electricity of the incumbent solution for Strategy C3 is −18,308, whereas that of Strategy C2 with max_iter_opt = 75 and max_iter_feas = 50 is −20,545. It can be deduced that the strategy of allowing only feasible solutions as a first priority, during the minimisation of infeasibility (Strategy C3) is better than allowing both feasible and infeasible solutions.

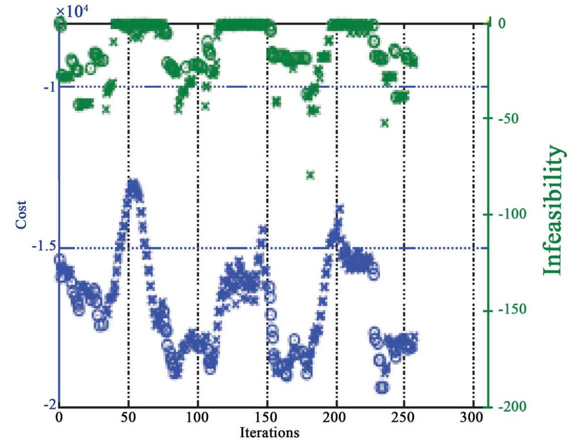


Figure 5. Strategy C3: Allowing all solutions within threshold during minimisation of infeasibility.

5.2.2. Diversification Strategy

Two strategies were applied to test diversification. In the experiments of Figure 3, diversification Strategy D1 was applied. In Strategy D1, diversification was applied if the incumbent solution did not improve for 100 iterations. The diversification was also subject to the current solution not improving for 5 consecutive iterations. Diversification was applied by making 5 consecutive random moves. The results for Strategy D1 show that this type of diversification does not result in an improvement in the incumbent solution. Experiments were performed with Strategy D2, where three consecutive restarts with the incumbent solution were performed, if the solution did not improve for 100 iterations. Strategy D2 is described by Pseudocode 1. Figure 6 shows that use of Strategy D2 for diversification results in an improvement in the incumbent solution. The cost of grid electricity of the incumbent solution is −16,884 with Strategy D1 and −19,504 with Strategy D2.

Pseudocode 1: Strategy D2

```
while iter ≤ max_iter do
    while iter ≤ max_iter_div do
        Perform Tabu Search
        Evaluate iterative solution S^current'
end while
iter_div_current ← 0
while (iter_div_current ≤ n^nonimprogv_div) and (S^current' < S^current) do
```

Continued

Perform Tabu Search
Evaluate iterative solution $S^{\text{current}'}$
end while
iter_restart $\leftarrow$ 0
while $\left(\text{iter\_div\_current} \le n^{\text{nonimprogv\_div}}\right)$ and $\left(S^{\text{current}} \le S^{\text{current}'}\right)$ **do**
Replace current solution with incumbent solution **while** $S^{\text{current}'} \leftarrow S^{\text{current}}$ **do**
Clear Tabu list $T^{\text{ist}} \leftarrow \varnothing$
Perform Tabu Search
Evaluate iterative solution $S^{\text{current}'}$
end while
end while

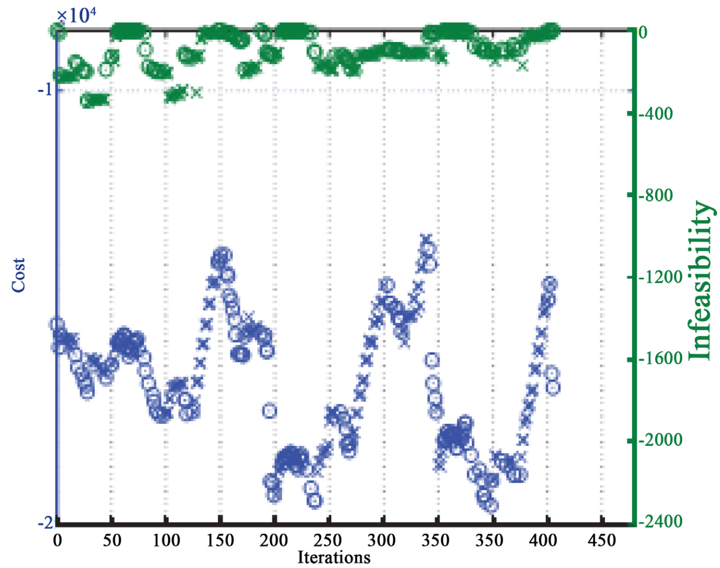


Figure 6. Strategy D2 + C2: Diversification by restarting with the incumbent solution and varying number of iterations for minimisation of cost and minimisation of infeasibility.

5.2.3. Multi-objective Optimisation Strategy

The experiments in this section are to investigate Strategy MOBJ1, developed to evaluate the multi-objective function, on a Pareto incumbent front, while taking into consideration its multi-period nature. Strategy MOBJ1 is compared to Strategy MOBJ2. In Strategy MOBJ2, the sum of the cost components of the objective function is calculated and the solution with the least sum is selected as the current solution. In Strategy MOBJ1 the multi-period cost components of the objective function are evaluated for non-dominance and form a Pareto incumbent front. Figure 7 shows the improvement in the incumbent solution using Strategy MOBJ1. The cost of grid electricity of the incumbent solution for Strategy MOBJ1 is −20,545, whereas there is no improvement in the incumbent solution with Strategy MOBJ2. As such, Strategy MOBJ1 where a Pareto incumbent front is used to evaluate the objective function is better than Strategy MOBJ2 which sums the cost components of the objective function.

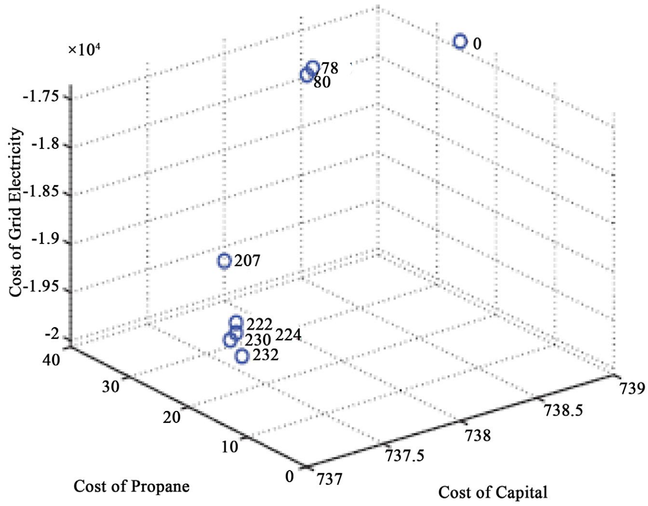


Figure 7. Strategy MOBJ1: Multi-objective optimisation using pareto incumbent front.

5.2.4. Multi-period Optimisation Strategy

The aim of the experiments in this section is to investigate the strategies for handling the multi-period nature of the optimisation problem, in a manner that will ensure continuity from one period to the next. The Tabu Search has two phases: (i) minimization of cost and (ii) minimisation of infeasibility. Different strategies for handling multi-periodicity are applied to the different phases. Each of these strategies is discussed next under the appropriate phase of the Tabu Search.

Round Robin in Phase 1 of Minimisation of Cost

In Strategy MP1 round robin of the months is carried out while updating the current solution, whether it is improving or not as described in Pseudocode 2.

In Strategy MP2, round robin of the months is carried out while updating the current solution with an improved solution only (Pseudocode 3). The results of experiments using Strategy MP2 are shown in Figure 8.

Pseudocode 2: Strategy MP1
1. **for** iter =1.12 **do**
2. Select a variable u_i^m for optimization
3. Perform Tabu Search
4. Evaluate iterative solution $S^{\text{current}'}$
5. Update the current solution $S^{\text{current}} \leftarrow S^{\text{current}'}$
6. Select the next month for which to carry out the optimization $m \leftarrow m+1$
7. Select the index of the next variable to be optimised $i \leftarrow i+1 \pmod 3$
8. **end for**

Pseudocode 3: Strategy MP2
1. **for** iter =1.12 **do**
2. iter_m ← 0
3. Select a variable u_i^m for optimization
4. Perform Tabu Search
5. Evaluate iterative solution $S^{\text{current}'}$

Continued

while $S^{current'} > S^{current}$ and iter_m≤12 **do**
Select the next month for which to carry out the optimization $m \leftarrow m+1$
Select the index of the next variable to be optimised $i \leftarrow i+1$ (mod 3)
Select a variable u_i^m for optimization
Perform Tabu Search
Evaluate iterative solution $S^{current'}$
iter_m ← iter_m + 1
end while
Update the current solution $S^{current} \leftarrow S^{current'}$
Select the next month for which to carry out the optimization $m \leftarrow m+1$
Select the index of the next variable to be optimised $i \leftarrow i+1$ (mod 3)
end for

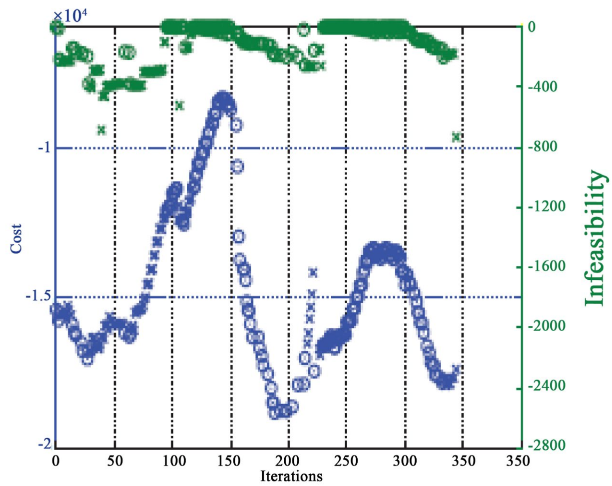


Figure 8. Strategy MP2: Round robin & updating current solution with an improving solution only.

The third multi-period strategy investigated is MP3, where round robin of the months is carried out and more than one variable is sampled in a given month, in order to obtain an improving solution. Strategy MP3 is described by Pseudocode 4. Strategy MP3 was investigated together with the Strategy C2 with max_iter_opt = 75 and max_iter_feas = 50.

Pseudocode 4: Strategy MP3

for iter =1.12 **do**
iter_m ← 0
Select a variable u_i^m for optimization
Perform Tabu Search
Evaluate iterative solution $S^{current'}$
while $S^{current'} > S^{current}$ and iter_m≤12 **do**
while $i \le 3$ and $S^{current'} > S^{current}$ **do**

Continued

Select a variable u_i^m for optimization
Perform Tabu Search
Evaluate iterative solution $S^{\text{current}'}$
Select the index of the next variable to be optimised $i \leftarrow i+1$ (mod 3)
end while
Select the next month for which to carry out the optimization $m \leftarrow m+1$
Select the index of the next variable to be optimised $i \leftarrow i+1$ (mod 3)
Select a variable u_i^m for optimization
Perform Tabu Search
Evaluate iterative solution $S^{\text{current}'}$
iter_m ← iter_m + 1
end while
Update the current solution $S^{\text{current}} \leftarrow S^{\text{current}'}$
Select the next month for which to carry out the optimization $m \leftarrow m+1$
Select the index of the next variable to be optimised $i \leftarrow i+1$ (mod 3)
end for

Strategies MP1 and MP2 are compared to Strategy MP3. Strategy MP3 gives the best cost of grid electricity of −20,545 for the incumbent solution. Strategies MP1 and MP2 give costs of grid electricity of −17,169 and −17,534, respectively.

The best strategy with regard to round robin, during the phase of minimisation of cost is MP3, where the current solution is updated with improving solutions only. This is done while trying out all the variables in turn in the same month, until the current solution improves or until after 12 iterations. A non improving solution is allowed only after the current solution has not been updated for 12 iterations. This strategy ensures that there is an attempt to find an improving solution in every month, and tries to build continuity from one month to the next during the optimisation.

Round Robin in Phase 2 of Minimisation of Infeasibility

Round robin is also investigated in Phase 2 where infeasibility is being minimized (Strategy MP4). The results are compared with those of Strategy MP3. Strategy MP3 also investigated the selection of the month for which to carry out the optimisation, during the phase for minimisation of infeasibility. In Strategy MP3 optimisation of the variables is done for the month with the least infeasible solution. The incumbent solution with Strategy MP3, is better than with Strategy MP4. −20,545 is obtained as the cost of grid electricity with Strategy MP3 and −16,691 with Strategy MP. During the minimisation of infeasibility, selection of the month with the most infeasible solution for optimisation (Strategy MP3) is therefore better than round robin of the months (Strategy MP4).

6. Limitations and Future Research Directions

While the proposed adaptations to the Tabu Search (TS) algorithm significantly improve its performance for the Waste-to-Energy Technology (WET) optimisation problem, certain limitations remain. These limitations stem from the algorithm's design choices, inherent complexity of the problem domain, and practical implementation constraints. Identifying these shortcomings can guide further refinement and inspire new avenues of research.

6.1. Sensitivity to Parameter Settings

The algorithm depends heavily on various predefined parameters such as max_iter, max_iter_opt, S_infeas, diversification thresholds, and the structure of the neighbourhood search. These parameters are problem-specific and must be finely tuned to achieve optimal performance.

Improper tuning can lead to premature convergence, inefficient exploration, or failure to find feasible solutions.

Incorporating learning-based or self-adaptive parameter adjustment strategies could allow the algorithm to dynamically fine-tune its behaviour during the search. In addition, a systematic study of parameter influence could help identify robust settings or provide guidelines for tuning in similar WET scenarios.

6.2. Computational Cost in Multi-Period, Multi-Objective Evaluation

The use of a Pareto incumbent front across multiple periods requires maintaining and evaluating a large number of non-dominated solutions. This becomes computationally intensive as the number of time periods and decision variables increases.

Scalability may be limited for larger problem instances (e.g., multi-year planning, more variables per period), leading to long runtimes.

Introducing surrogate (meta) models to approximate objective function evaluations can reduce computational burden. Also, combining TS with Genetic Algorithms (GA) or NSGA-II can improve scalability while maintaining diversity and exploration.

6.3. Dependence on Black-Box WET Models

The algorithm treats the WET models as black boxes, with no gradient or structural information used in the search. While this increases generality, it limits the efficiency of the search, especially in identifying and avoiding infeasible regions. Search may waste time evaluating infeasible or non-informative regions, particularly in large and sparse feasible spaces.

Incorporate domain knowledge or partial model understanding (e.g., constraint bounds, sensitivity analysis) to guide search more intelligently. Also, the use machine learning to learn feasibility boundaries or infeasibility trends from historical evaluations.

6.4. Limited Exploration Beyond Local Regions

Although diversification strategies are introduced (e.g., restarts with incumbent solution), they are still based on the local region around the best-known solution. There is limited capability for global exploration or large structural changes in the solution space.

The search may remain confined to local basins of attraction, especially in highly non-convex or discontinuous spaces.

Introduce frequency-based memory or historical solution archives to promote broader exploration. Also, periodically apply large perturbations or problem-specific heuristics to explore distant regions of the search space.

6.5. No Explicit Handling of Uncertainty or Dynamic Changes

The current algorithm assumes deterministic input parameters and static operating conditions. In real-world WET scenarios, parameters such as feedstock availability, energy demand, and cost factors can vary over time or be uncertain.

Solutions may be suboptimal or infeasible under real operating conditions if the model does not account for uncertainty.

Adapt the algorithm to handle uncertainty by optimising expected performance or worst-case outcomes. In addition, use multiple demand or supply scenarios in the objective evaluation and solution comparison. The summary of limitations and research directions are presented in **Table 5**.

Table 5. Summary of Limitations and Research Directions.

Limitation	Impact	Suggested future work
Sensitivity to parameters	May cause inefficiency or convergence issues	Adaptive parameter tuning, sensitivity analysis
Computational burden of Pareto evaluation	Limits scalability to large/multi-year problems	Surrogate models, hybridisation with GA or NSGA-II
Black-box model treatment	Inefficient navigation of feasible space	Constraint learning, domain-informed guidance
Limited global exploration	Risk of premature convergence to local optima	Long-term memory, Large Neighbourhood search
Lack of uncertainty handling	Reduced real-world applicability	Robust/stochastic optimisation, scenario-based approaches

7. Conclusion

The developments to the basic Tabu Search were designed to handle constraints, multi-objectives, multi-periods and diversification. Experiments were done, to test the adaptations developed. It was found out that for optimisation of WET, constraints are best handled by alternating between allowing feasible and infeasible solutions. The minimisation of cost should also be alternated with the minimisation of infeasibility, for different numbers of iterations. Diversification should be applied by performing restarts with the incumbent solution. It was also found out that evaluation of the multi-period cost components of the objective function on a Pareto incumbent front, is better than summing the cost components of the objective function. During minimization of cost, a round robin strategy of the months, should be applied, whereas during minimisation of infeasibility, the period with the most infeasible solution should be selected for optimisation.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Razali, M.K.M., Ayob, M., Rahman, A.H.A., Jarmin, R., Liu, C.Y., Maaya, M., *et al.* (2025) Unveiling Effective Heuristic Strategies: A Review of Cross-Domain Heuristic Search Challenge Algorithms. *Computer Modeling in Engineering & Sciences*, **142**, 1233-1288. https://doi.org/10.32604/cmes.2025.060481

[2] Arık, O.A. (2024) Fuzzy Rule-Based Variable Neighborhood Search Algorithm for Single-Machine Weighted Earliness/Tardiness Scheduling with Common Due Date. *Neural Computing and Applications*, **37**, 3355-3371. https://doi.org/10.1007/s00521-024-10844-5

[3] Glover, F. and Laguna, M. (1997) Tabu Search. Kluwer Academic. https://doi.org/10.1007/978-1-4615-6089-0

[4] Srivastava, S., Tripathi, A., Bansal, S. and Vuppuluri, P.P. (2025) Introduction to Optimization: Techniques and Applications in Engineering. In: Bansal, S., Tripathi, A., Srivastava, S. and Vuppuluri, P.P., Eds., *Nature-Inspired Metaheuristic Algorithms*, CRC Press, 1-28. https://doi.org/10.1201/9781003612858-1

[5] Martí, R., Sevaux, M. and Sörensen, K. (2025) Fifty Years of Metaheuristics. *European Journal of Operational Research*, **321**, 345-362. https://doi.org/10.1016/j.ejor.2024.04.004

[6] Ben Abdellafou, K., Hadda, H. and Korbaa, O. (2018) An Improved Tabu Search Meta-Heuristic Approach for Solving Scheduling Problem with Non-Availability Constraints. *Arabian Journal for Science and Engineering*, **44**, 3369-3379. https://doi.org/10.1007/s13369-018-3525-3

[7] Sharma, S., Khanum, S., Madhumala, R.B., Maroju, A. and Aggarwal, S. (2025) Prioritization and Meta-Heuristic Approach for Efficient Congestion Control in Vehicular Ad Hoc Networks (VANETs). *International Journal of Information Technology*. https://doi.org/10.1007/s41870-025-02542-9

[8] Glover, F. (1997) Tabu Search and Adaptive Memory Programming—Advances, Applications and Challenges. In: Barr, R.S., Helgason, R.V. and Kennington, J.L., Eds., *Interfaces in Computer Science and Operations Research*, Springer, 1-75. https://doi.org/10.1007/978-1-4615-4102-8_1

[9] Laguna, M. (2018) Tabu Search. In: Martí, R., Pardalos, P. and Resende, M., Eds., *Handbook of Heuristics*, Springer, 741-758. https://doi.org/10.1007/978-3-319-07124-4_24

[10] Zhou, X., Ma, H., Gu, J., Chen, H. and Deng, W. (2022) Parameter Adaptation-Based Ant Colony Optimization with Dynamic Hybrid Mechanism. *Engineering Applications of Artificial Intelligence*, **114**, Article ID: 105139.

https://doi.org/10.1016/j.engappai.2022.105139

[11] Bou Saleh, M., Chariete, A., Schwartz, L., Grunder, O. and El Hassani, A.H. (2024) Reactive Tabu Search and Mixed-Integer Linear Programming for Multi-Day Assignment, Scheduling, and Routing Problems of Specialised Education and Home-Care Services. *International Journal of Production Research*, **63**, 1779-1802. https://doi.org/10.1080/00207543.2024.2391947

[12] David, M., Uzorka, A. and Makeri, Y.A. (2022) Optimisation of a Renewable Energy System for Rural Electrification. *Journal of Power and Energy Engineering*, **10**, 1-15. https://doi.org/10.4236/jpee.2022.1011001

[13] Iliopoulou, C., Kepaptsoglou, K. and Vlahogianni, E. (2019) Metaheuristics for the Transit Route Network Design Problem: A Review and Comparative Analysis. *Public Transport*, **11**, 487-521. https://doi.org/10.1007/s12469-019-00211-2

[14] Pirim, H., Bayraktar, E. and Eksioglu, B. (2008) Tabu Search: A Comparative Study (Vol. 278). Intech Open Access Publisher. https://doi.org/10.5772/5637

[15] Makumbi, D., Uzorka, A. and Ajiji Makeri, Y. (2022) Number of Cattle for Commercialising Electricity from Cattle Waste to Energy Technology. *International Research Journal of Applied Sciences*, *Engineering and Technology*, **8**, 1-14. https://cirdjournals.com/index.php/irjaset/article/view/824

[16] Arostegui, M.A., Kadipasaoglu, S.N. and Khumawala, B.M. (2006) An Empirical Comparison of Tabu Search, Simulated Annealing, and Genetic Algorithms for Facilities Location Problems. *International Journal of Production Economics*, **103**, 742-754. https://doi.org/10.1016/j.ijpe.2005.08.010

[17] Zhi, J. and Keskin, B.B. (2018) A Multi-Product Production/distribution System Design Problem with Direct Shipments and Lateral Transshipments. *Networks and Spatial Economics*, **18**, 937-972. https://doi.org/10.1007/s11067-018-9436-8

[18] Mota, A., Ávila, P., Bastos, J., Roque, L.A.C. and Pires, A. (2025) Comparative Analysis of Simulated Annealing and Tabu Search for Parallel Machine Scheduling. *Procedia Computer Science*, **256**, 573-582. https://doi.org/10.1016/j.procs.2025.02.154

[19] Ekşioğlu, B., Ekşioğlu, S.D. and Jain, P. (2008) A Tabu Search Algorithm for the Flowshop Scheduling Problem with Changing Neighborhoods. *Computers & Industrial Engineering*, **54**, 1-11. https://doi.org/10.1016/j.cie.2007.04.004

[20] Ansótegui, C., Heymann, B., Pon, J., Sellmann, M. and Tierney, K. (2018) Hyper-reactive Tabu Search for MaxSAT. In: Battiti, R., Brunato, M., Kotsireas, I. and Pardalos, P., Eds., *Learning and Intelligent Optimization*, Springer, 309-325. https://doi.org/10.1007/978-3-030-05348-2_27

[21] Chou, X., Gambardella, L.M. and Montemanni, R. (2021) A Tabu Search Algorithm for the Probabilistic Orienteering Problem. *Computers & Operations Research*, **126**, Article ID: 105107. https://doi.org/10.1016/j.cor.2020.105107

[22] Uzorka, A., Kibirige, D., Mustafa, M.M. and Ukagwu, J.K. (2025) Design and Implementation of a Photovoltaic System for Health Facilities in Rural Areas of Uganda. *Discover Applied Sciences*, **7**, Article No. 197. https://doi.org/10.1007/s42452-025-06640-y

[23] Pothiya, S., Ngamroo, I. and Kongprawechnon, W. (2008) Application of Multiple Tabu Search Algorithm to Solve Dynamic Economic Dispatch Considering Generator Constraints. *Energy Conversion and Management*, **49**, 506-516. https://doi.org/10.1016/j.enconman.2007.08.012

[24] Batstone, D.J., Keller, J., Angelidaki, I., Kalyuzhnyi, S.V., Pavlostathis, S.G., Rozzi, A.,

et al. (2002) The IWA Anaerobic Digestion Model No 1 (ADM1). *Water Science and Technology*, **45**, 65-73. https://doi.org/10.2166/wst.2002.0292

[25] National Renewable Energy Laboratory (2002) Advanced Vehicle & Fuels Research. https://docs.nrel.gov/docs/fy02osti/31967.pdf

[26] Mohan, N. (2014) Advanced Electric Drives: Analysis, Control, and Modeling Using MATLAB/Simulink. John Wiley & Sons. https://doi.org/10.1002/9781118910962

[27] National Gas (2022) Characteristics of LP Gas. https://nationalgasco.net/characteristics-of-lpg/

[28] Lusk, P. (1998) Methane Recovery from Animal Manures. The Current Opportunities Casebook. https://docs.nrel.gov/docs/fy99osti/25145.pdf

[29] Burns, R. (2007) Status of Manure Anaerobic Digestion in the United States. *International Conference on* 21*st Century Challenges to Sustainable Agri-Food Systems*: *Biotechnical*, *Environment*, *Nutrition*, *Trade and Policy*, 15-17 March 2007, 476-481.

[30] NYSERDA (2022) DG/CHP Integrated Data System. https://der.nyserda.ny.gov/data/