Les Cahiers du GERAD ISSN: 0711–2440

Adaptation of Tabu Search for Optimisation of Biomass Waste to Energy Conversion Systems

R. Namuli, B. Jaumard,
P. Pillay

G–2012–48

August 2012

Adaptation of Tabu Search for Optimisation of Biomass Waste to Energy Conversion Systems

Rachel Namuli

Department of Electrical and Computer Engineering
Concordia University
Montreal (Quebec) Canada, H3G 2W1
r_namuli@encs.concordia.ca

Brigitte Jaumard

GERAD & Department of Computer Science and Software Engineering
Concordia University
Montreal (Quebec) Canada, H3G 2W1
bjaumard@cse.concordia.ca

Pragasen Pillay

Department of Electrical and Computer Engineering
Concordia University
Montreal (Quebec) Canada, H3G 2W1
pillay@encs.concordia.ca

August 2012

Les Cahiers du GERAD

G–2012–48

Abstract

We study the optimisation of a biomass waste to energy conversion system using an adapted Tabu Search heuristic. It corresponds to a non-linear and non-convex optimisation problem whose solution involves several optimisation sub-problems, including three with differential equations. In solving this complex optimisation problem, four contributions have been made to the adaptation of Tabu Search for use in the optimisation of biomass energy conversion systems. These are: multi-period and diversification strategies that lead to an effective search of the solution space, handling of constraints by development of different strategies for searching feasible regions, with some incursions into infeasible regions to find a shortcut towards feasible regions, and evaluation of a multi-objective function exploiting an approximation of the Pareto front. The results of the experiments show that the resulting Tabu Search heuristic, gives better solutions for this type of optimisation problem, compared to the basic Tabu Search. The developed Tabu Search was used to maximize revenue from biomass waste to energy conversion systems for two types of livestock (cows and swines). The Tabu Search was also used to identify the minimum herd size required for commercial viability of a biomass waste to energy conversion system. Experiments show that the adapted Tabu Search corresponds to a very useful tool for determination of commercial viability of biomass waste to energy conversion systems.

Key Words: Tabu Search, Pareto, infeasibility, constraints satisfaction, multi-period, diversification.

Acknowledgments: This project is part of the R&D program of the NSERC Chair entitled "Energy efficiency in electrical machines for small renewable energy production systems", established in 2009 at "Concordia University". The authors acknowledge the support of the Natural Sciences & Engineering Research Council of Canada and Hydro-Quebec.

1 Introduction

A biomass waste to energy conversion system (BWECS) produces heat and electricity from organic waste. The heat and electricity can be used by rural farms and remote communities to meet their needs. The schematic of the BWECS optimised in this paper is shown in Figure 1. In this system, biogas is generated from the anaerobic digestion of biomass waste and combusted in an internal combustion engine and boiler. The internal combustion engine generates a torque that is used by an induction machine to generate electricity. Exhaust heat from the internal combustion engine is captured by a heat exchanger. Heat from the system is also used to maintain the digester's temperature at its operating point. There is a backup propane supply to supplement biogas combusted in the boiler. There is a connection to an electricity grid. Electricity from the grid is used to supply the farm if the electricity generated by the BWECS cannot meet the demand. The capital expenditure on BWECS is high, making it desirable to maximise revenue from these systems. Revenue is obtained by sale of electricity generated to utility companies and from renewable energy incentives. Revenue is maximised by the minimisation of the system's costs. In order to maximise revenue from a BWECS, the system must be optimised.

This paper presents an adaptation of the Tabu Search that is used to optimise the BWECS. A multi-period optimisation strategy has been developed and a method of handling infeasible solutions suited to this application has been developed. The cost components of the objective function are evaluated separately resulting in a multi-objective function. The function is also multi-period in nature. The multi-objective and multi-period solutions are evaluated on a Pareto incumbent front. A diversification strategy suited to the multiperiod problem, that allows the solution to move to new regions has been developed. Experiments using different parameters to tune the Tabu Search optimisation are presented and the results are discussed.

The paper is organised as follows: Section 2 is on the statement of the optimisation problem, Section 3 is on the literature review carried out, Section 4 describes the optimisation problem, Section 5 reviews the aspects of the Tabu Search that were developed to suit the problem being solved and Section 6 describes the Tabu Search algorithm. The experiments carried out and results obtained are described and discussed in Section 7, and the conclusion is given in Section 8.

2 Statement of the Optimisation of BWECS

2.1 Outline of the Problem

The optimisation problem consists in dimensioning the BWECS 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 BWECS under study is shown in Figure 1, for a farm with n_{herd} livestock (cows and swines in the experimental results). Dimensioning is carried out with an adapted monthly setup, for: the backup propane flow rate, x_1^m, the split of biogas between the internal combustion engine (ICE) and the boiler, x_2^m and the volume flow rate of manure from the lagoon, x_3^m. This is subject to the constraint of operating the BWECS 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 goes into a lagoon, where it is stored. The manure from the lagoon is fed to a digester at a volume flow rate, x_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 x_2^m. The mass flow rate of biogas going into the internal combustion engine is $(1-x_2^m)m_{\text{biogas}}$ and that going into the boiler is $m_{\text{biogas}}x_2^m$. 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 BWECS it is sent to the electricity grid. The electricity sent to the grid is designated by $d_{\text{e}}^m - y_1^m$. If the electricity generated by the BWECS is insufficient to meet the demand of the farm, electricity is obtained from the grid and is designated by $y_1^m - d_{\text{e}}^m$. 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 exhaust gases is captured by the heat exchanger (HEX) and forms the heat

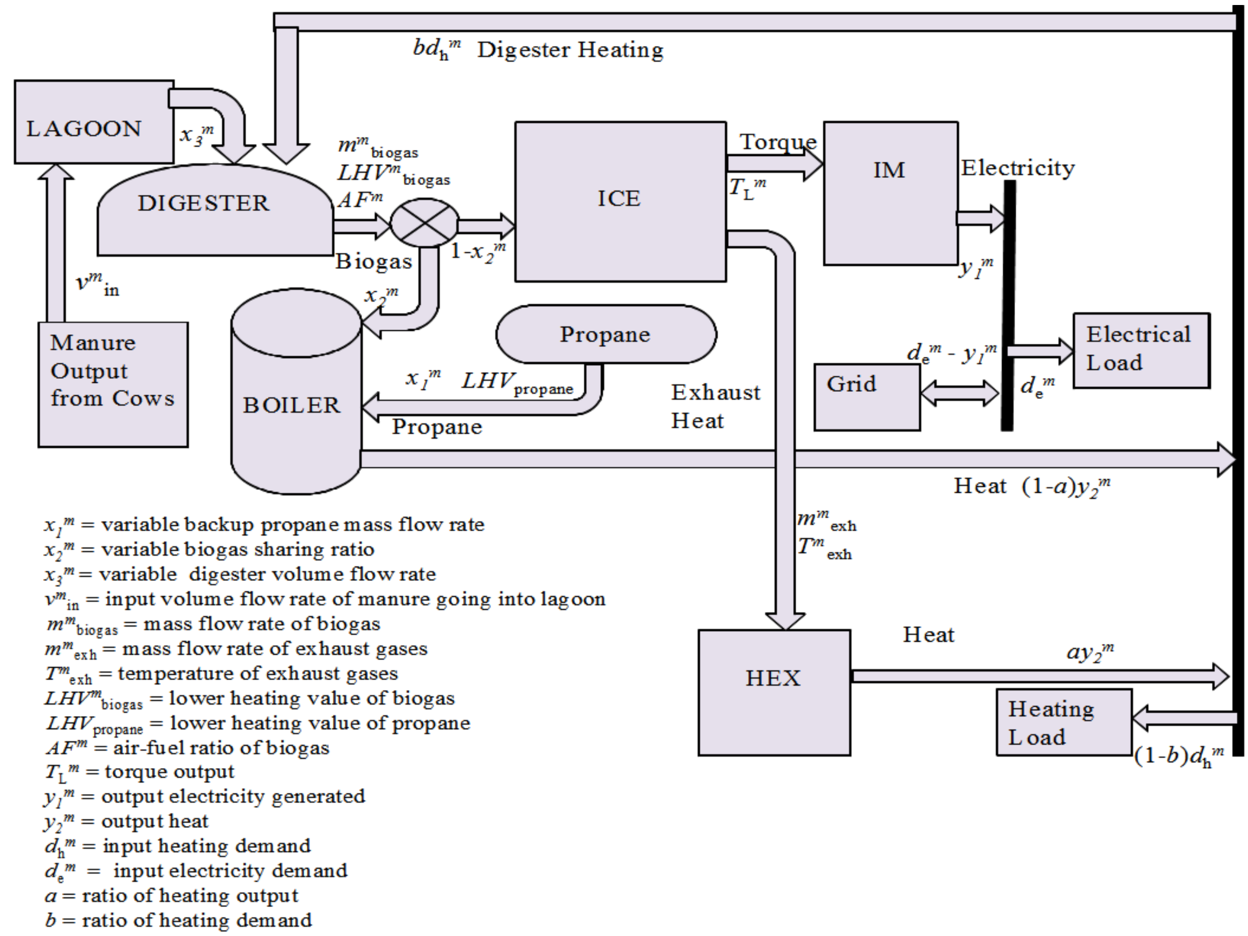


Figure 1: Biomass Waste to Energy Conversion System Model

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 $(1-b)d_h^m$. 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 x_1^m and lower heating value LHV_{propane}. The optimisation of the BWECS described is done with the objective of maximising revenue. The optimisation problem is expressed as a cost minimisation problem by:

$$\min f^{\text{ cost}}\,(x_1^m, x_2^m, x_3^m) \text{ for a given manure input } v_{\text{in}}\ , \tag{1}$$

$$\text{subject to: } C_{\text{BWECS}}(x_1^m, x_2^m, x_3^m) \leq 0 \text{ for } m \in M, \tag{2}$$

$$\text{such that }\ : x_1^m \in \{0, 0.0001,\ 0.0002,\ 0.0003, ..., 0.0036\} \text{ for } m \in M, \tag{3}$$

$$x_2^m \in \{0,\ 0.01,\ 0.02,\ 0.03, ..., 0.99\} \text{ for } m \in M, \tag{4}$$

$$x_3^m \in \{1,\ 2,\ 3, ..., 59\} \text{ for } m \in M, \tag{5}$$

$$x = (x_1^1, x_2^1, x_3^1, x_1^2, x_2^2, x_3^2, ..., x_1^{|m|}, x_2^{|m|}, x_3^{|m|}) \text{ for } m \in M, \tag{6}$$

where x_1^m, x_2^m and x_3^m are the variables: backup propane mass flow rate, biogas sharing ratio and volume flow rate of manure going into the digester respectively. C_{BWECS} denotes a set of global constraints, some of which are linear and others non-linear. The set of global constraints will be described in Section 4.2. Using the variables, x_1^m, x_2^m and x_3^m, the outputs y_1^m and y_2^m can be obtained as described in Algorithm 1. x denotes the solution of the optimisation problem as described in the Tabu Search (see Algorithm 2). The outputs y_1^m and y_2^m will be used in Section 4.1 to describe the components of the objective function.

2.2 Optimisation Process Flow

Algorithm 1 describes the process flow of the optimisation. The inputs of the BWECS are: herd size n_{herd}, electricity demand d_{e}^{m}, heating demand d_{h}^{m} and volume flow rate of manure from the livestock v_{in}^{m}. These inputs are specified for each time period, $m \in M$. The parameters of the optimisation are initialised, i.e., V_{lagoon}^{o}, volume of manure in the lagoon, a, ratio of heating output, b, ratio of heating demand, η_{HEX}, efficiency of the heat exchanger, η_{boiler}, efficiency of the boiler, T_{water}, water temperature and LHV_{propane}, lower heating value of propane. An initial solution (x_1^m, x_2^m, x_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 BWECS, using the functions: LAGOON, DIGESTER, ICE, IM, and the linear equations of the heat exchanger and the boiler (see Section 4.2). 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 BWECS optimisation problem. References for these component models are given in Section 4. Each of the component models of the functions DIGESTER, ICE and IM model a difficult non-linear 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 (x_1^m, x_2^m, x_3^m), as shown in Algorithm 1. As such the non-linear optimisation problems of the component models are solved by optimisation of the BWECS, with the solution (x_1^m, x_2^m, x_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. The Tabu Search algorithm is described in Section 5.

Algorithm 1 Optimisation of a BWECS

Initialization

Inputs: $n_{\text{herd}},\ d_{\text{e}}^{m},\ d_{\text{h}}^{m},\ v_{\text{in}}^{m}$ for $m \in M$

Initialize parameters: $V_{\text{lagoon}}^{0},\ a,\ b,\ \eta_{\text{HEX}},\ \eta_{\text{boiler}},\ T_{\text{water}},\ LHV_{\text{propane}}$

for $m \in M$ **do**

  Build an initial solution (x_1^m, x_2^m, x_3^m) for $m \in M$

  Calculate the outputs of the BWECS model components

$$(V_{\text{lagoon}}^{m}, x_3^m) = \text{LAGOON}(v_{\text{in}}^{m}, V_{\text{lagoon}}^{m-1}, n_{\text{herd}})$$

$$(AF^m, LHV_{\text{biogas}}^{m}, m_{\text{biogas}}^{m}) = \text{DIGESTER}(x_3^m, bd_{\text{h}}^{m})$$

$$(T_L^m, m_{\text{exh}}^m, T_{\text{exh}}^m, cp_{\text{exh}}^m) = \text{ICE}(m_{\text{biogas}}^m, (1 - x_2^m),\ AF^m,\ LHV_{\text{biogas}}^m)$$

$$y_1^m = \text{IM}(T_{\text{L}}^m)$$

$$ay_2^m = \eta_{\text{HEX}}\, m_{\text{exh}}^m\, cp_{\text{exh}}^m (T_{\text{exh}}^m - T_{\text{water}})$$

$$(1 - a)y_2^m = (LHV_{\text{propane}}\, x_1^m + m_{\text{biogas}}^m\, LHV_{\text{biogas}}^m\, x_2^m)\, \eta_{\text{boiler}}$$

end for

Evaluate the objective function f^{cost}

Tabu Search Optimisation

iter $\leftarrow 0$

while iter $\leq$ max_iter **do**

  Perform Tabu search which includes evaluation of each of the BWECS model components

  Evaluate iterative solutions and update the incumbent solutions accordingly (see Section 4.1 on formation of Pareto incumbent solutions from the objective function)

end while

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
$LHV_{\mathrm{biogas}}^{m}$	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 Literature Review on BWECS Related Systems

This section discusses previous work done on optimisation of biomass waste to energy conversion systems. There have been studies on optimisation of biomass waste to energy conversion systems. These studies however did not use the Tabu Search technique and used simplified linear equations with linear constraints to model the optimisation problem. The optimisation problem of the BWECS in this paper is modeled differently, and it has non-linear and non-convex constraints. Two of the studies that used simplified linear equations, Rentizelas et al. [1], and Bruglieri and Liberti [2], are discussed in this section.

The optimisation in Rentizelas et al. [1] and that of the research in this paper have the same objective, maximisation of revenue, but are not solving the problem in the same manner. The variables used in Rentizelas et al. [1] were obtained from the biomass supply chain, processing and storage capacities, whereas the variables used in the research being carried out are obtained from the energy conversion processes. The problem being solved in Rentizelas et al. [1] is based on the biomass supply chain, processing and storage, whereas that being solved in the research being carried out is based on the energy conversion processes. Another difference is that Rentizelas et al. [1] used genetic algorithms and sequential quadratic programming to solve the optimisation problem.

Bruglieri and Liberti [2] optimised the energy production process for a biomass based energy production system. The objective of the optimisation was to minimise the total operational costs. This is similar to the objective of the optimisation in the research being carried out in this paper, however again, the problem is solved differently. The system model in Bruglieri and Liberti [2] comprised of process sites. The optimisation was based on a material balance of the inputs and outputs of the sites. This is different from the optimisation problem in the research being carried out whose system model comprises of energy conversion process components, and the optimisation is based on the energy conversion processes. The optimisation problem in Bruglieri and Liberti [2] was solved using CPLEX (ILOG [3]) and a spatial branch-and-bound solver.

Both Rentizelas et al. [1] and Bruglieri and Liberti [2] used techniques suited to supply chain or production process models and greatly simplified the optimisation problem. This is different from the optimisation of BWECS of this paper, in that the latter is modeled based on the energy conversion processes in the system components, which is solved by a Tabu Search algorithm. The following section describes the formulation of the BWECS optimisation problem.

4 Description of the Optimisation of BWECS

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 BWECS 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 BWECS 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.

4.1 Objective Function

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

4.1.1 Cost of Capital

The cost of capital C^m_{capital} 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^m_{in}. 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^m_{capital}. The cost of capital is calculated using the non-linear function (7), details of which can be found in Namuli et al. [4].

$$C^m_{\text{capital}} = \text{ CAPITAL }\Big(\text{HRT},\, c_{\text{digester}},\, c_{\text{lagoon}},\, P_{\text{rated}},\, c_{\text{engine}},\, c_{\text{boiler}},\, C_{\text{cap\_incentive}},\, i_{\text{rate}},\, n_{\text{period}},\, v^m_{\text{in}},\, V_{\text{lagoon\_storage}},\, d^m_{\text{h}},\, ay^m_2\Big), \qquad \text{for } m \in M, \quad (7)$$

where CAPITAL is the function for calculation of the cost of capital, v^m_{in} 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^m_{h} is the heating load, a is the ratio of heat output from the heat exchanger, y^m_2 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.

4.1.2 Cost of Propane

The monthly cost of propane, C^m_{propane} is a linear function of the backup propane mass flow rate, x^m_1 and is given by (8). For the details of the function see Namuli et al. [4].

$$C^m_{\text{propane}} = \text{PROPANE}(c_{\text{propane}},\, x^m_1) \qquad \text{for } m \in M, \quad (8)$$

where PROPANE is the function for calculating the cost of propane, c_{propane} is the unit cost of propane and x^m_1 is the backup propane mass flow rate.

4.1.3 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 (9), details of which can be found in Namuli et al. [4].

$$C^m_{\text{incentives}} = \text{INCENTIVES}(c_{\text{incentives}},\, y^m_1) \qquad \text{for } m \in M, \quad (9)$$

where $C^m_{\text{incentives}}$ is the cost of incentives, INCENTIVES is the function for calculating the cost of incentives, y^m_1 is the electricity output and $c_{\text{incentives}}$ is the unit cost of incentives.

4.1.4 Cost of Grid Electricity

The cost of grid electricity, $C^m_{\text{grid\_electricity}}$ is a non-linear function of the electricity output, y^m_1 (10), the details of which can be found in Namuli et al. [4].

$$C^m_{\text{grid\_electricity}} = \text{GRID\_ELECTRICITY}(c_{\text{tariff}},\ d^m_{\text{e}},\ y^m_1) \qquad \text{for } m \in M, \tag{10}$$

The four cost components of the objective function form a multi-objective optimisation 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 multi-objective 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:

$$z = \sum_{m=1}^{M} (C^m_{\text{capital}} + C^m_{\text{propane}} - C^m_{\text{incentives}} + C^m_{\text{grid\_electricity}}) \qquad \text{for } m \in M. \tag{11}$$

The drawback of (11) 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

$$\vec{f_k}^{\text{cost}} = \left[\sum_{m=1}^{M} C^m_{\text{capital}},\ \sum_{m=1}^{M} C^m_{\text{propane}},\ -\sum_{m=1}^{M} C^m_{\text{incentives}},\ \sum_{m=1}^{M} C^m_{\text{grid\_electricity}} \right] \qquad \text{for } k \in K \text{ and } m \in M, \tag{12}$$

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.

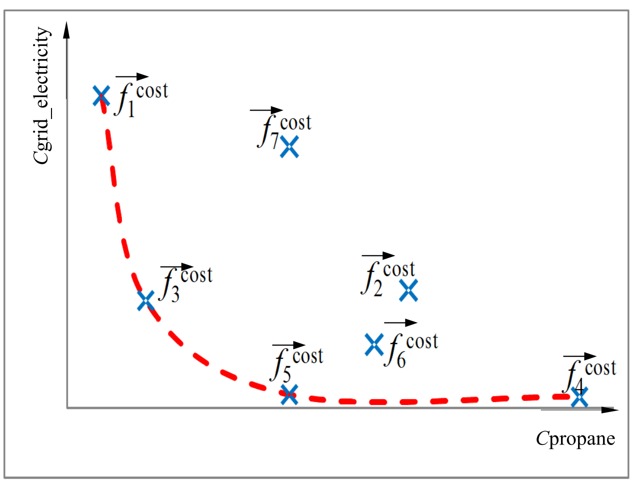


Figure 2: Illustration of Pareto Incumbent Front

4.2 Global Constraints

This section describes the global constraints, C_{BWECS}, and how they are derived from the optimisation problem. The initial solution described in Algorithm 1 satisfies the global constraints. In the Tabu Search optimisation that follows, the solution (x^m_1, x^m_2, x^m_3) has to be checked for satisfaction of the global constraints. These constraints are defined as:

$$0 \leq (v_{\text{in}}^m n_{\text{days}}^m + V_{\text{lagoon\_manure}}^{m-1} - n_{\text{days}}^m x_3^m) \leq V_{\text{lagoon\_storage}}\, v_{\text{in}}^m, \tag{13}$$

$$(V_{\text{D}} - \text{HRT}x_3^m) \geq 0, \tag{14}$$

$$(P_{\text{rated}}/\omega_{\text{mech}} - \text{ICE}(LHV_{\text{biogas}}^m,\ \omega_{\text{mech}},\ (1-x_2^m)m_{\text{biogas}}^m)) \geq 0, \tag{15}$$

$$d_{\text{h}}^m \leq \Bigg(\eta_{\text{HEX}}\, m_{\text{exh}}^m\, cp_{\text{exh}}(T_{\text{exh}}^m - T_{\text{water}}) + (LHV_{\text{propane}}\, x_1^m + m_{\text{biogas}}^m\, LHV_{\text{biogas}}^m\, x_2^m)\, \eta_{\text{boiler}}\Bigg) \leq (d_{\text{h}}^m + \delta_{\text{h}}), \tag{16}$$

$$(b_{\text{r}} - d_{\text{h}}^m + \eta_{\text{HEX}}\, m_{\text{exh}}^m\, cp_{\text{exh}}(T_{\text{exh}}^m - T_{\text{water}})) \leq 0, \tag{17}$$

for $m \in M$,

where v_{in}^m is the volume flow rate of manure from the livestock, n_{days}^m are the number of days, $V_{\text{lagoon\_manure}}^{m-1}$ is the volume of manure in the lagoon, x_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, LHV_{biogas}^m is the lower heating value of biogas, x_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_{exh}^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, x_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, x_3^m is varied to minimise the cost of the system. Constraint (13) 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 $n_{\text{day}}^m x_3^m$, 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 livestock in month m. Constraint (13) also ensures that the volume of manure in the lagoon is not greater than the storage capacity of the lagoon.

Constraint (14) is set to ensure that the volume of manure in the digester, $\text{HRT}x_3^m$, 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 LHV (Lower Heating Value) of biogas, LHV_{biogas}^m can be found in **?** [5]. 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 National Renewable Energy Laboratory [6]. The internal combustion engine is coupled to an induction machine of rating, P_{rated}, that generates electric power. The induction machine is modeled using non-linear differential equations detailed in Mohan [7]. 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 (15) 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. A propane tank that supplies propane at a mass flow rate, x_1^m is included in the BWECS. The heat produced by the boiler is calculated from the mass flow rate of biogas to the boiler, $m_{\text{biogas}}^m\, x_2$, the mass flow rate of propane, x_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 (16) is set to ensure that the heat output of the BWECS meets the heating demand of the farm and the digester.

Constraint (17) 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 (17). The boiler rating is calculated by a non-linear equation given in Namuli et al. [4].

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

$$f^{\text{infeas}} = \sum_{m=1}^{M} (S^m_{\text{lagoon\_volume}} + S^m_{\text{digester\_size}} + S^m_{\text{mbiogas}} + S^m_{\text{heating\_demand}} + S^m_{\text{boiler\_rating}}) \qquad \text{for } m \in M, \quad (18)$$

where f^{infeas} is the total measure of infeasibility, $S^m_{\text{lagoon\_volume}}$, $S^m_{\text{digester\_size}}$, S^m_{mbiogas}, $S^m_{\text{heating\_demand}}$ and $S^m_{\text{boiler\_rating}}$ 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 (13), (14), (15), (16) and (17). Using the measure of infeasibility of the volume of manure in the lagoon as an example we have:

$$n^m_{\text{days}} x^m_3 + S^m_{\text{lagoon\_volume}} = V^{m-1}_{\text{lagoon\_manure}} - V_{\text{lagoon\_storage}} v^m_{\text{in}} + v^m_{\text{in}} n^m_{\text{days}} \qquad \text{for } m \in M, \qquad (19)$$

the solution is feasible for $S^m_{\text{lagoon\_volume}} \geq 0$,

where n^m_{days} are the number of days, x^m_3 is the volume flow rate of manure from the lagoon, $S^m_{\text{lagoon\_volume}}$, is the measure of infeasibility of the volume of manure in the lagoon, $V^{m-1}_{\text{lagoon\_manure}}$ is the volume of manure in the lagoon, $V_{\text{lagoon\_storage}}$ is the storage capacity of the lagoon and v^m_{in} is the volume flow rate of manure from the livestock. The other measures of infeasibility are defined similarly. The handling of infeasibility is discussed in Section 6.2.2.

5 Tabu Search

This section reviews the Tabu Search and the aspects that have been developed to adapt the Tabu Search to the optimisation of a BWECS.

5.1 Introduction

The motivation for using the Tabu Search heuristic is the discrete variables of the optimisation problem, the non-convexity of the constraints and non-linearity of the functions used to model the components of the BWECS. This makes the optimisation problem computationally complex with a number of local optima. A deterministic metaheuristic, i.e., Tabu Search is preferred in order to better take advantage of the characteristics of the BWECS. A deterministic metaheuristic moves aggressively to a local optimum, which would shorten the computational time, compared to random search metaheuristics like simulated annealing and genetic algorithms. Before going into the description of the Tabu Search, a literature review of the Tabu Search and the features that have been developed for solving the BWECS optimisation problem are discussed.

5.2 Basic Tabu Search

The Tabu Search is applicable to highly combinatorial problems (Glover [8]) which can be formulated as optimisation problems: minimise $f(x) : x \in X$ where $f(x)$ is the objective function, and x is selected from a set of constraints X. In Tabu Search, a move n leads from one solution to the next. The move is defined as $n : X(n) \to X$. The set of moves that can be applied to x is defined as $\mathcal{N}(x)$ and is termed the neighbourhood of x (Glover [8]). A characteristic of the Tabu Search is to constrain the search by restricting moves (Glover [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 optimal. In the basic Tabu Search, moves that are in the Tabu list are not allowed during the optimisation process, during a given number of iterations (Glover [8]). 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. This paper has developed strategies for application of the Tabu Search to the optimisation of a BWECS. The strategies developed are:

(i) evaluation of a multi-objective function using a Pareto front that incorporates the multi-period nature of the problem,
(ii) constraint handling using different strategies for exploring feasible regions with some incursions into infeasible regions,
(iii) multi-period optimisation and
(iv) diversification using some consecutive restarts.

Since many papers have been published on Tabu Search, we next review the most recent or relevant papers with respect to multi-objective optimisation, constraint handling, multi-period planning and diversification.

5.3 Multi-objective Optimisation

There are various methods used to evaluate multi-objective functions in an optimisation problem, some of which are: (i) the objectives can be combined into a normalised weighted function, (ii) one of the objectives can be evaluated at each iteration of the optimisation, or (iii) Pareto optimal solutions can be determined.

In Choobineh et al. [9] a single-machine scheduling problem was solved using a multi-objective Tabu Search and a weighted objective function.

In Kulturel-Konak et al. [10] a different objective was evaluated at each iteration of the optimisation. A multinomial probability mass function was used to select the objective to be evaluated at each iteration.

Baykasoglu [11] solved a mechanical component design problem using a multi-objective Tabu Search optimisation. The Pareto optimal method was used to evaluate the multi-objective function. The solution to proceed with the iteration was selected randomly from a set of Pareto solutions.

The method used to evaluate the objective functions in the research being carried out is the Pareto front method, where a set of Pareto incumbent solutions is kept. The selection of the solution to proceed with the iteration is done differently in the research being carried out. Baykasoglu [11] randomly selected a solution to proceed with the iteration, the research being carried out in this paper, selects the best solution by weighting and summing the components of the objective function. The details of the method developed are given in Section 6.2.1. Selection of the best solution is done to allow the search to proceed with the minimised value instead of a randomly selected value. In addition, the weighting of the cost components of the objective function is aimed at ensuring that their sum is not dominated by the cost component with the largest value.

5.4 Constraint Handling

In the handling of constraints, it is good to allow infeasibility for non-convex constraints. When dealing with non-convex constraints, allowing feasible solutions only, results in the solution taking a longer path towards an optimum. This is because the solution path is limited to feasible regions only. This path can be shortened by allowing infeasible solutions during the optimisation.

There are different ways of handling infeasibility. In Korsvik and Fagerholt [12], a ship routing and scheduling problem was solved using Tabu Search. The problem was divided into a main and a sub-problem. Tabu Search was used to solve the main problem for optimising the shipping route. The sub-problem which optimised the quantity of cargo being shipped, was formulated as a linear programming problem. Infeasible solutions were generated in the sub-program. A fast heuristic was used to obtain feasible solutions for the sub-problem. In the research being carried out in this paper, the problem is not split into a main and a sub-problem. The entire problem is solved using Tabu Search while allowing both infeasible and feasible solutions.

In Lim et al. [13] infeasible solutions were handled by incorporating a random move sub-routine into a Tabu Search algorithm for a freight allocation problem. The optimisation problem encountered infeasible solutions, which were allowed.

The method of constraint handling developed in the research being carried out in this paper is different from that in Korsvik and Fagerholt [12] and Lim et al. [13]. The difference with the constraint handling in the research being carried out in this paper, is in the strategy developed to handle the infeasibility within the multi-period optimisation. A second objective function that minimises infeasibility is introduced. At each iteration the optimisation is carried out for the period with the most infeasible solution. This is described further in Section 6.2.2.

5.5 Multi-period Planning

The Tabu Search optimisation in this paper is a multi-period one. Different strategies are used for Tabu Search multi-period optimisation.

In Mantawy et al. [14] a long term hydro scheduling problem was studied. The period for which the optimisation was to be carried out was selected at random. In Nayak and Rajan [15] a Tabu Search was used to minimise the cost of turning on and off generating units in a hydro-thermal power system. The period for which the optimisation was to be carried out, was also randomly selected.

A different strategy was used in Tippayachai et al. [16], where an enhanced Tabu Search algorithm was used for solving an economic dispatch problem for power generating units. Selection of generating units for optimisation was done using a round robin method. A different generating unit was selected at each iteration.

Another strategy for handling multi-period Tabu Search optimisation is to treat the period as a variable. This was done in Bolduc et al. [17]. Treating the period as a variable results in the neighbourhood consisting of Tabu moves from one period to another.

In a multi-period optimisation one has to worry about the difficulty of smoothing the transition from one period to another during the optimisation. The drawback of the multi-period optimisation strategy of the studies cited, with the exception of Bolduc et al. [17], is that the selection of the period to be optimised was being done at random. The result is that there is no smooth transition from one period to another. The strategy developed in the research being carried out in this paper is different from those reviewed. In selection of the period for which the optimisation is to be carried out, there is a balance between use of the round robin method and selection of the period with the most infeasible solution. Details of how this is done are given in Section 6.2.3.

5.6 Diversification Strategy

Diversification drives the Tabu Search into new regions. Diversification is applied if the incumbent solution does not improve after a given number of iterations. Three methods of diversification are: (i) performing random moves, (ii) performing a restart with the incumbent solution and (iii) generating a random solution as the current solution.

In Driouch et al. [18], random moves were made in order to diversify the Tabu Search in the optimal scheduling of a multiuser MIMO (Multiple Input Multiple Output) CDMA (Code Division Multiple Access) system.

Diversification by performing a restart with the incumbent solution was done in Brandao [19], where a Tabu Search algorithm was applied to a heterogeneous fixed fleet vehicle routing problem.

Scheduling of trucks in cross-docking systems was done in Vahdani and Zandieh [20] using different meta-heuristics, that included the Tabu Search. In the Tabu Search heuristic of Vahdani and Zandieh [20], diversification was applied by generating a random solution and using it as the current solution.

We develop a diversification strategy that ensures that each of the variables being optimised contributes to the move to a new region. In our diversification strategy three consecutive restarts are performed with the incumbent solution, if the incumbent solution does not improve for max_iter_div iterations. If the incumbent solution improves after a restart is performed, the Tabu Search exits the diversification loop and proceeds with the optimisation. This is different from what is being done in Brandao [19], and allows the solution to test three different regions during the diversification. The diversification strategy will be discussed further in Section 6.2.4.

6 Description of the Tabu Search Algorithm

This section describes the adaptations of the Tabu Search algorithm developed for optimisation of a biomass waste to energy conversion system. The Tabu Search is described in Algorithm 2. The notation and the parameters of the Tabu Search are given in Tables 2 and 3 respectively.

Algorithm 2 Tabu Search

Initialization
Build a feasible initial solution $x_i^{m,\text{init}}$
Set $x_i^m \leftarrow x_i^{m,\text{init}}, \quad S^{\text{best}} \leftarrow \{x_i^{m,\text{init}}, \quad m \in M, \quad i = 1,2,3\}$
Initialize the Tabu list: $T \leftarrow \emptyset$
Set the bounds
Evaluate $\vec{f}_{\min}^{\text{cost}}, \quad \vec{f}_{\min}^{\text{infeas}}$
Tabu Search
iter $\leftarrow 0$
while iter $\leq$ max_iter **do**
  while iter $\leq$ max_iter_div **do**
    Phase 1: Minimize Cost
    iter_opt $\leftarrow 0$
    while iter_opt $\leq$ max_iter_opt /*Attempt at finding a solution with a smaller cost regardless of the infeasibility*/ **do**
      Perform a round robin search on the months : For a given month $m \in M$, select one variable with index $i(m) : i(m) = i(m-1) + 1 \pmod 3$
      Update the neighbourhood of the selected variable
      Evaluate all solutions $x_i^{m'}$ in $\mathcal{N}(x_i^m)$ with respect to $\vec{f}_{\min}^{\text{cost}}, \quad \vec{f}_{\min}^{\text{infeas}}$ (only for storage with solution)
      $S^{\text{current}} \leftarrow \underset{x'}{\text{argmin}} \vec{f}_{\min}^{\text{cost}}(x') \quad \text{for} \;\; x = (x_1^1, x_2^1, x_3^1, x_1^2, x_2^2, x_3^2, ..., x_1^{|m|}, x_2^{|m|}, x_3^{|m|}) \quad \text{and} \quad m \in M$
    end while
    iter_feas $\leftarrow 0$
    Phase 2: Minimize Infeasibility
    while iter_feas $\leq$ max_iter_feas /* Reducing infeasibility*/ **do**
      Select the month $m \in M$ for which the search is to be carried out : $m \leftarrow \underset{m'}{\text{argmax}} \vec{f}_{\min}^{\text{infeas}}(x')$ for $x = (x_1^1, x_2^1, x_3^1, x_1^2, x_2^2, x_3^2, ..., x_1^{|m|}, x_2^{|m|}, x_3^{|m|})$
      Update the neighbourhood of the selected variable
      Evaluate all solutions $x_i^{m'}$ in $\mathcal{N}(x_i^m)$ with respect to $\vec{f}_{\min}^{\text{infeas}}, \quad \vec{f}_{\min}^{\text{cost}}$ (only for storage of solution)
      $S^{\text{current}} \leftarrow \underset{x'}{\text{argmin}} \vec{f}_{\min}^{\text{infeas}}(x') \quad \text{for} \;\; x = (x_1^1, x_2^1, x_3^1, x_1^2, x_2^2, x_3^2, ..., x_1^{|m|}, x_2^{|m|}, x_3^{|m|}) \quad \text{and} \quad m \in M$
    end while
  end while
  Apply diversification
end while

Table 2: Tabu Search Notation

Notation	Description
$x_i^{m,\text{ init}}$	initial solution
x_i^m	current solution
S^{best}	set of the Pareto incumbent solutions
S^{current}	set of the Pareto current solutions
$\mathcal{N}(x_i^m)$	neighbourhood of variable x_i^m
LB_v	lower bound of neighbourhood
UB_v	upper bound of neighbourhood
T	Tabu list
X^{MODEL}	set of constraints to be satisfied by the BWECS black box models
X^{GLOBAL}	set of global constraints to be satisfied by the optimisation

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, 8000 swines
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)	46300 [21]
T_{water}	water temperature (°C)	35
η_{HEX}	heat exchanger efficiency (%)	70
η_{rated}	boiler efficiency (%)	70
c_{lagoon}	unit cost of lagoon ($/m^3)	2.47 [22]
c_{propane}	unit cost of propane ($/m^3)	1.98[23]
$c_{\text{incentives}}$	unit cost of incentives ($/kWh)	0.07 [24]
$n^{\text{rand\_div}}$	number of consecutive random moves for diversification Strategy D1	5
$n^{\text{nonimprov\_div}}$	number of consecutive non-improving moves after which to apply diversification	5
$n^{\text{restart\_div}}$	number of restarts with incumbent solution for 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 (cows data), 50 (swines data)
S^{infeas}	threshold of infeasibility	varies
S_0^{infeas}	initial threshold of infeasibility	varies

6.1 Basic Tabu Search Algorithm

As described in Section 5.2, 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 BWECS are discussed in this section.

6.1.1 Definition of the Neighbourhood

The neighbourhood of x_i^m is defined as:

$$\mathcal{N}(x_i^m)^{new} = \left\{ \begin{array}{ll} v : v = x_i^m + \delta_i & i = 1,2,3 \\ \quad v = x_i^m - \delta_i & m \in M \\ & x \in X^{\text{MODEL}} \cup X^{\text{GLOBAL}} \end{array} \right\}$$

$$LB_v \leq v \leq UB_v : v \in \mathcal{N}(x_i^m),$$

where x_i^m is the optimisation variable, X^{MODEL} is the set of constraints to be satisfied by the BWECS black box models, X^{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 x_i^m to $x_i^m \pm \delta_i$ is selected within the specific limits and step sizes for the different variables, specified in Section 2.1.

6.1.2 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 x_i^m to $x_i^m \pm \delta_i$, 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.

6.1.3 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.

6.2 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. This section describes the adaptations developed.

6.2.1 Pareto Incumbent Solutions

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 (12). The cost vectors are then checked for non-dominance and the non-dominated solutions form a Pareto incumbent front, as described in Section 4.1. Summing the cost components separately over all the periods, M, incorporates the multi-period nature of the optimisation into the minimisation of the objective function. This is different from the references cited in the literature review in that although the optimisation is carried out for one period, the Pareto incumbent front is formed by summing the cost components over all the periods. The Pareto front method of evaluating multi-objective functions has therefore been modified to incorporate the multi-period nature of the optimisation problem.

6.2.2 Method of Handling Constraints

There are two sets of constraints in the optimisation problem of the BWECS. X^{MODEL} is the set of constraints to be satisfied by the models of the BWECS and X^{GLOBAL} is the set of global constraints to be satisfied by the solution of the optimisation problem, i.e., $C_{\text{BWECS}}(x_1^m, x_2^m, x_3^m)$ (2). The global constraints are defined in Section 4.2. The set of constraints to be satisfied by the models of the BWECS, X^{MODEL} is not defined in this paper because the BWECS models are treated as black boxes in the optimisation problem. The method of handling constraints discussed in this paper applies to the set of global constraints, X^{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 (18). 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.

6.2.3 Multi-period Optimisation Strategy

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 in this paper 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.

6.2.4 Diversification

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.

Experiments were carried out to test the Tabu Search aspects developed in this paper. The following section describes the experiments and discusses the results.

7 Experiments

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

7.1 Data Instances

Two data instances are used in the experiments. One of the data instance is obtained from a dairy farm of herd size 500 cows (NYSERDA [25]) and the other is obtained from a swine farm of herd size 8000 swines Khakbazan [26].

7.2 Descriptions of the Strategies of the Tabu Search Experiments

The experiments carried out are grouped into strategies. Many strategies were tested and we report the most successful ones. The strategies correspond to the aspects of the Tabu Search developed and discussed in Section 6.2 and 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 minimisation 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;

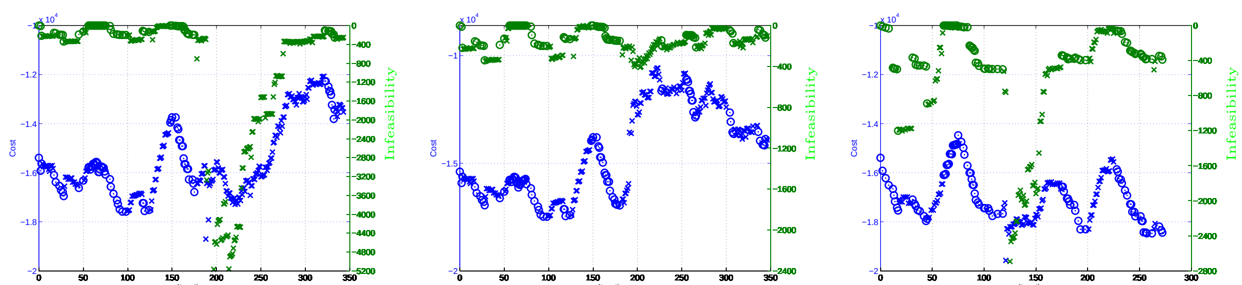
(x) Strategy MP3, round robin and updating solution with improving solution only, and sampling all variables in one month if required;

(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 multi-period 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. The summary of the experimental results is given in Tables 4 and 5, for the cows and swines data instances, respectively.

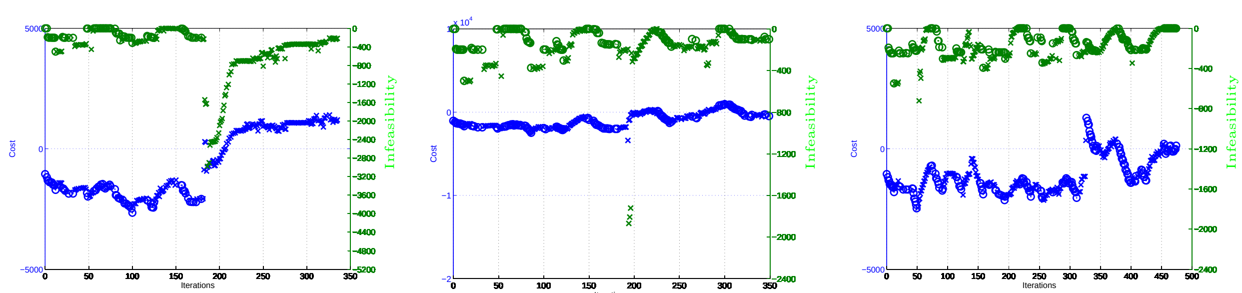
7.2.1 Constraints Handling Strategy

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, for each of the data instances. These are $S^{\text{infeas}} = -500$, -200 and -100 for the cows data instance and $S^{\text{infeas}} = -500$, -205 and -100 for the swines data instance. The results of fixing the threshold of infeasibility to $S^{\text{infeas}} = -200$ for the cows data instance and to $S^{\text{infeas}} = -205$ for the swines data instance, are shown in Figures 3(a) and 4(a), respectively. When the threshold is fixed to $S^{\text{infeas}} = -500$, the cost reaches low values. However these low values are in the infeasible regions. This is seen in Tables 4 and 5. The costs of grid electricity for the current solutions are $-18,278$, $-18,197$, $-19,091$ and $-19,949$ at the 50^{th}, 100^{th}, 200^{th} iterations and at termination, respectively for the cows data instance. The costs of grid electricity for the current solutions are -4327, -4131, -3844 and -3534 at the 50^{th}, 100^{th}, 200^{th} iterations and at termination,



(a) Fixed Threshold $S^{\text{infeas}} = -200$ (b) Varying Thresholds $S_{\text{o}}^{\text{infeas}} = -200$ (c) Varying Thresholds $S_{\text{o}}^{\text{infeas}} = -300$

Figure 3: Strategy C1 (Cows)



(a) Fixed Threshold $S^{\text{infeas}} = -205$ (b) Varying Thresholds $S_{\text{o}}^{\text{infeas}} = -205$ (c) Varying Thresholds $S_{\text{o}}^{\text{infeas}} = -250$

Figure 4: Strategy C1 (Swines)

respectively for the swines data instance. The resulting incumbent solutions have higher values of $-16,854$ for the cost of grid electricity of the cows data instance, and -4064 for the swines data instance. Fixing the threshold of infeasibility to a lower value of $S^{\text{infeas}} = -200$, for the cows data instance and $S^{\text{infeas}} = -205$, for the swines data instance, gives better incumbent solutions. Table 4 for the cows data instance shows that iterations 100, 200 and the termination condition have costs of grid electricity for the incumbent solutions of $-16,130$, $-16,525$, $-16,627$, $-16,730$ and $-16,834$. The current solution also reaches relatively low values of costs of grid electricity of $-17,071$ and $-18,302$ at the 50$^{\text{th}}$ and 100$^{\text{th}}$ iterations respectively. Better incumbent solutions are also obtained from the swines data instance at $S^{\text{infeas}} = -205$. This is shown in Table 5, where with $S^{\text{infeas}} = -205$, the incumbent solution has a cost of grid electricity of -4064 by the 50$^{\text{th}}$ iteration, and at termination the incumbent solution has a cost of grid electricity of -4630. 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 $-16,130$, for the cows data instance, and -4209, for the swines data instance. 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 100$^{\text{th}}$ and 200$^{\text{th}}$ iterations, and at termination respectively, for the cows data instance (Table 4). 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. With the swine data instance, the cost of grid electricity of the incumbent solution improves slightly from -4064 to -4209 (Table 5). This implies that the search remains in a local region. From these experiments, a good starting point for the threshold of infeasibility is identified as $S^{\text{infeas}} = -200$ for the cows data instance and $S^{\text{infeas}} = -205$ for the swines data instance.

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 $S_{\text{o}}^{\text{infeas}} = -500$, -300, -200 and -100 for the cows data instance and $S_{\text{o}}^{\text{infeas}} = -500$, -250, -205 and -100 for the swines data instance. The thresholds of infeasibility were varied as follows:

$$\text{for } S_{\text{o}}^{\text{infeas}} = -500; \qquad S^{\text{infeas}} \in \{-500, -400, -300, -200, -100, -50, -40, -30, -20, -10\}, \tag{20}$$

$$\text{for } S_{\text{o}}^{\text{infeas}} = -300; \qquad S^{\text{infeas}} \in \{-300, -200, -100, -50, -40, -30, -20, -10\}, \tag{21}$$

$$\text{for } S_{\text{o}}^{\text{infeas}} = -250; \qquad S^{\text{infeas}} \in \{-250, -245, -240, -235, -230, -225, -220, ..., 165\}, \tag{22}$$

$$\text{for } S_{\text{o}}^{\text{infeas}} = -205; \qquad S^{\text{infeas}} \in \{-205, -200, -195, -190, -185, -180, -175, ..., 120\}, \tag{23}$$

$$\text{for } S_{\text{o}}^{\text{infeas}} = -200; \qquad S^{\text{infeas}} \in \{-200, -150, -100, -90, -80, -70, -60, -50, ..., -10\}, \tag{24}$$

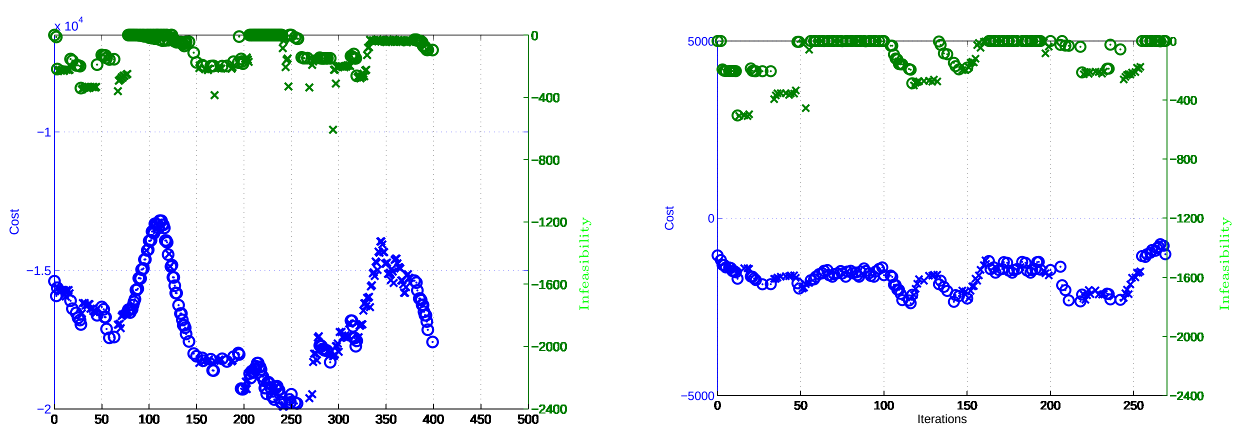
$$\text{for } S_{\text{o}}^{\text{infeas}} = -100; \qquad S^{\text{infeas}} \in \{-100, -90, -80, -70, -60, -50, -40, -30, -20, -10\}, \tag{25}$$

where $S_{\text{o}}^{\text{infeas}}$ 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) for the cows data instance, and Figures 4(b) and 4(c) for the swines data instance. These results are summarised in Tables 4 and 5. Table 4 for the cows data instance shows that with Strategy C1 and with $S_{\text{o}}^{\text{infeas}} = -500$, the cost of grid electricity of the current solution reaches very low values. However these occur in the infeasible regions. With the same strategy and $S_{\text{o}}^{\text{infeas}} = -500$, the incumbent solution of the swines data instance does not improve from the 50$^{\text{th}}$ iteration onwards (Table 5). For both data instances, the current solutions have high values of infeasibility with $S_{\text{o}}^{\text{infeas}} = -500$, compared to the other experiments with the other values of $S_{\text{o}}^{\text{infeas}}$. $S_{o}^{\text{infeas}} = -500$ is therefore too large a threshold to keep the solution in a feasible region. With $S_{\text{o}}^{\text{infeas}} = -300$, the incumbent solution obtained for the cows data instance is worse than with $S_{\text{o}}^{\text{infeas}} = -500$. The incumbent solution is the same for $S_{\text{o}}^{\text{infeas}} = -500$ and $S_{\text{o}}^{\text{infeas}} = -300$ for the swines data instance. These results can be explained as $S_{\text{o}}^{\text{infeas}} = -300$ being a large threshold of infeasibility, that keeps the solution in infeasible regions. With $S_{\text{o}}^{\text{infeas}} = -100$, there is no improvement in the incumbent solution for the swines data instance, whereas there is a slight improvement in the incumbent solution for the cows data instance. This is because $S_{\text{o}}^{\text{infeas}} = -100$ is too low for the search to move away from a local region. $S_{\text{o}}^{\text{infeas}} = -200$ and $S_{\text{o}}^{\text{infeas}} = -205$ for the cows and swines data instances, respectively, give the best results.

With regard to Strategy C1, a cost of grid electricity of $-16,884$ for the incumbent solution, is obtained, at $S_{\text{o}}^{\text{infeas}} = -200$, for the cows data instance (Table 4). $S^{\text{infeas}} = -200$ also gave the best incumbent solution for the cows data instance, 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. For the swines data instance the incumbent solution is the same with $S_{\mathrm{o}}^{\mathrm{infeas}} = S^{\mathrm{infeas}} = -205$ (Table 5), however this is the best incumbent solution from experiments of Strategy C1.

Figures 3(b) and 4(b), show a move towards lower costs at the beginning of the iterations, for $S_{\mathrm{o}}^{\mathrm{infeas}} = -200$, for the cows data instance and $S_{\mathrm{o}}^{\mathrm{infeas}} = -205$, for the swines data instance. 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 100$^{\mathrm{th}}$ iteration (Figures 3(b) and 4(b)). In order to obtain improving incumbent solutions after the 100$^{\mathrm{th}}$ 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. The discussion on the experiments of the diversification strategies is done in Section 7.2.2.

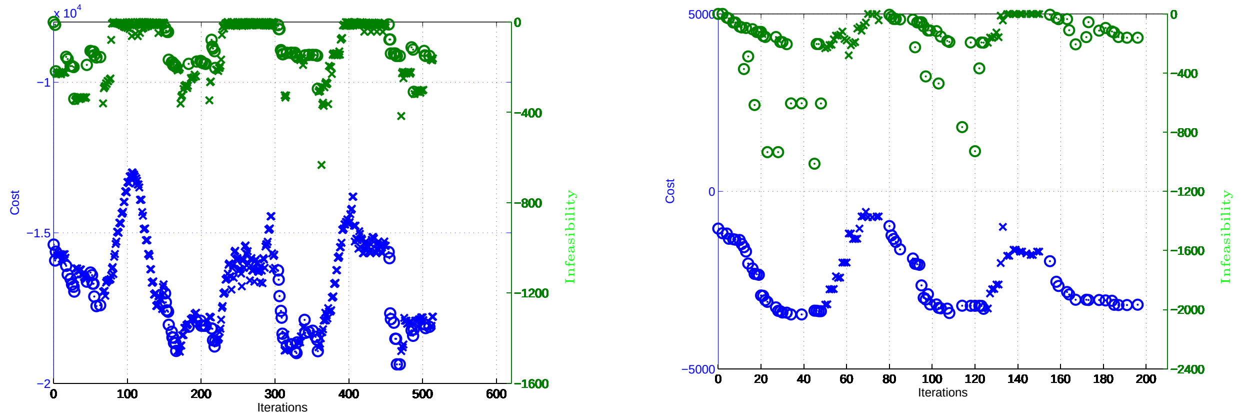
In Strategy C2, the number of iterations for the minimisation of cost and minimisation of infeasibility were varied. In the first experiment done, the same number of iterations were allowed for minimisation of cost and minimisation of infeasibility, i.e., max_iter_opt=max_iter_feas = 50. The results are shown in Tables 4 and 5 for the cows and swines data instances respectively. These results are compared to those of Strategy C1 at $S_{\mathrm{o}}^{\mathrm{infeas}} = -200$ for the cows data instance (Figure 3(b)), and $S_{\mathrm{o}}^{\mathrm{infeas}} = -205$ for the swines data instance (Figure 4(b)), where max_iter_opt = 50 and max_iter_feas = 25. For both data instances, 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. For the cows data instance the best parameters for Strategy C2 were found to be max_iter_opt = 75 and max_iter_feas = 50 (Figure 5(a)). 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 $-19,504$, for the cows data instance (Table 4). The best parameters for Strategy C2 with the swines data instance were found to be max_iter_opt = 50 and max_iter_feas = 25 (Figure 7(b)). The cost of grid electricity of the incumbent solution with max_iter_opt = 50 and max_iter_feas = 25 for the swines data instance was -5425, whereas that with max_iter_opt = 75 and max_iter_feas = 50 was -4123 (Table 5). These comparisons show that varying the number of iterations for which the minimisation of cost and the minimisation of infeasibility are carried out, impacts the incumbent solution.



(a) Cows: max_iter_opt = 75, max_iter_feas = 50

(b) Swines: max_iter_opt = 50, max_iter_feas = 50

Figure 5: Strategy C2: Varying Number of Iterations for Minimisation of Cost and Minimisation of Infeasibility



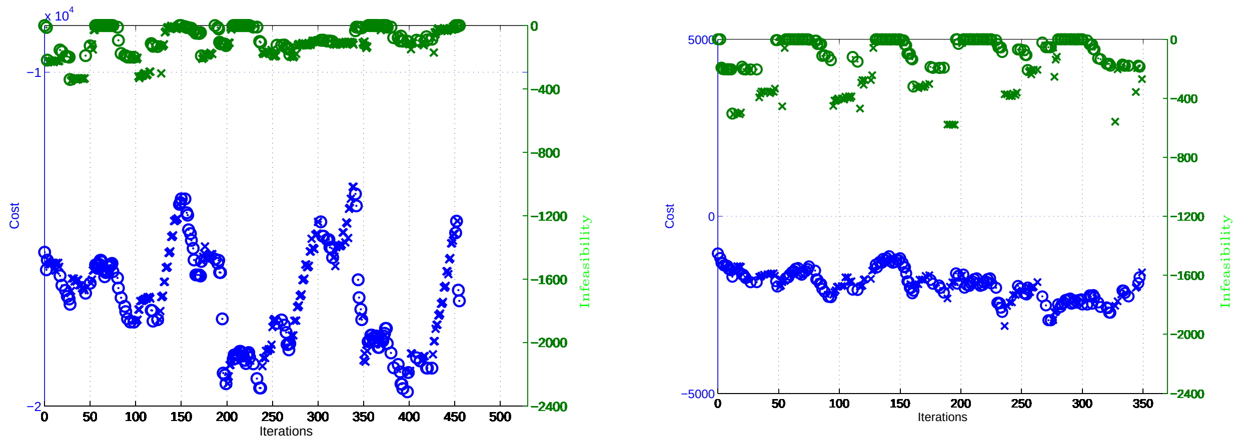
(a) Cows (b) Swines

Figure 6: Strategy C3: Allowing All Solutions Within Threshold During 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. 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 (Tables 4 and 5). From Table 4, of the cows data instance, 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 $-20,545$. For the swines data instance, the cost of grid electricity of the incumbent solution for Strategy C3 is -4870, whereas that of Strategy C2 with max_iter_opt $= 50$ and max_iter_feas $= 25$ is -5425 (Table 5). 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 during minimisation of infeasibility.

7.2.2 Diversification Strategy

Two strategies were applied to test diversification. In the experiments of Figures 3 and 4, 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 (Figures 3 and 4 and Tables 4 and 5), show that this type of diversification does not result in an improvement in the incumbent solution, except in one case (Figure 4(c)). 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, for the cows data instance and for 50 iterations, for the swines data instance. A different number of iterations for application of diversification was used for the cows and swines because the the data instances had different ranges. This meant the Tabu Search progressed at different rates for the two data instances. Strategy D2 is described by Pseudocode 1. Figure 7 shows that use of Strategy D2 for diversification results in an improvement in the incumbent solution. For the cows data instance, the cost of grid electricity of the incumbent solution is $-16,884$ with Strategy D1 and $-19,504$ with Strategy D2 (Table 4). For the swines data instance, the cost of grid electricity of the incumbent solution is -4630 with Strategy D1 and -5222 with Strategy D2 (Table 5).



(a) Cows:max_iter_opt = 50 and max_iter_feas = 25 (b) Swines:max_iter_opt = 50 and max_iter_feas = 25

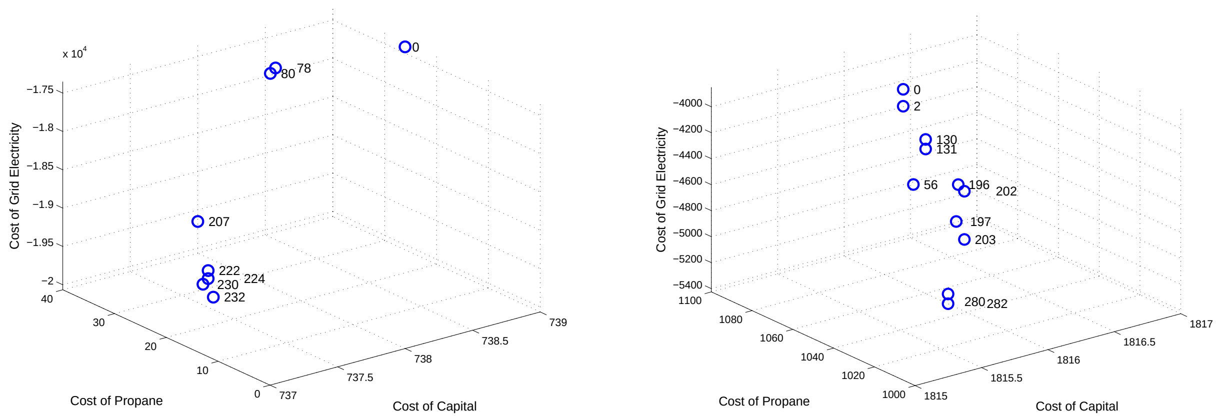
Figure 7: Strategy D2+C2: Diversification by Restarting with the Incumbent Solution and Varying Number of Iterations for Minimisation of Cost and Minimisation of Infeasibility

Pseudocode 1: Strategy D2

while iter $\leq$ max_iter **do**
  while iter $\leq$ max_iter_div **do**
    Perform Tabu Search
    Evaluate iterative solution $S^{\text{current}'}$
  end while
  iter_div_current $\leftarrow 0$
  while (iter_div_count $\leq n^{\text{nonimprov\_div}}$) and ($S^{\text{current}'} < S^{\text{current}}$) **do**
    Perform Tabu Search
    Evaluate iterative solution $S^{\text{current}'}$
  end while
  iter_restart $\leftarrow 0$
  while (iter_restart $\leq n^{\text{restart\_div}}$) and ($S^{\text{incumbent}} \leq S^{\text{current}'}$) **do**
    Replace the current solution with the incumbent solution $S^{\text{current}'} \leftarrow S^{\text{incumbent}}$
    Clear Tabu list $T \leftarrow \emptyset$
    Perform Tabu Search
    Evaluate iterative solution $S^{\text{current}'}$
  end while
end while

7.2.3 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 8 shows the improvement in the incumbent solution using Strategy MOBJ1. From Table 4 of the cows data instance, 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. Table 5 for the swines data instance shows a cost of grid electricity of -5425 for the incumbent solution, with Strategy MOBJ1, and -5255 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.



(a) Cows (b) Swines

Figure 8: Strategy MOBJ1: Multi-Objective Optimisation using Pareto Incumbent Front

7.2.4 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) minimisation 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

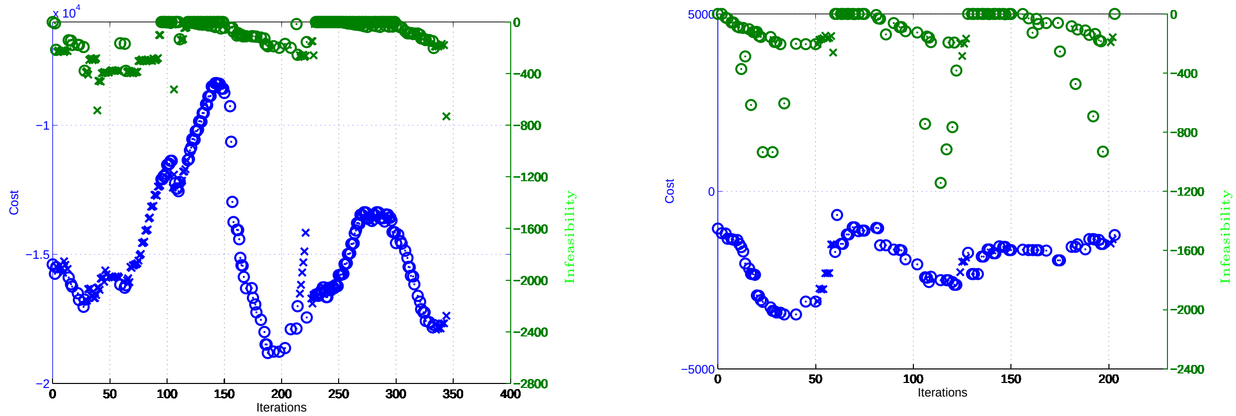
In Strategy MP1 round robin of the months is carried out while updating the current solution, whether it is improving or not as described in Pseudocode 2. The results of the experiments using Strategy MP1 are shown in Tables 4 and 5, for the cows and swines data instances respectively.

Pseudocode 2: Strategy MP1

for iter $= 1 : 12$ **do**
  Select a variable x_i^m for optimisation
  Perform Tabu Search
  Evaluate iterative solution $S^{\text{current}'}$
  Update the current solution $S^{\text{current}} \leftarrow S^{\text{current}'}$
  Select the next month for which to carry out the optimisation $m \leftarrow m + 1$
  Select the index of the next variable to be optimised $i \leftarrow i + 1 (\text{mod } 3)$
end for

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 9 for both the cows and swines data instances.

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. For the cows data instance, Strategy MP3 was investigated together with the Strategy C2 with max_iter_opt = 75 and max_iter_feas = 50. Similarly, for the swines data instance, Strategy MP3 was investigated together with the Strategy C2 with max_iter_opt = 50 and max_iter_feas = 25.



(a) Cows (b) Swines

Figure 9: Strategy MP2: Round Robin & Updating Current Solution with an Improving Solution Only

Pseudocode 3: Strategy MP2

for iter = 1 : 12 **do**
  iter_m ← 0
  Select a variable x_i^m for optimisation
  Perform Tabu Search
  Evaluate iterative solution $S^{\text{current}'}$
  while $S^{\text{current}'} > S^{\text{current}}$ and iter_m ≤ 12 **do**
    Select the next month for which to carry out the optimisation $m \leftarrow m + 1$
    Select the index of the next variable to be optimised $i \leftarrow i + 1 (\text{mod } 3)$
    Select a variable x_i^m for optimisation
    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 optimisation $m \leftarrow m + 1$
  Select the index of the next variable to be optimised $i \leftarrow i + 1 (\text{mod } 3)$
end for

Strategies MP1 and MP2 are compared to Strategy MP3. Table 4 shows that Strategy MP3 gives the best cost of grid electricity of $-20,545$ for the incumbent solution, for the cows data instance. Strategies MP1 and MP2 give costs of grid electricity of $-17,169$ and $-17,534$, respectively, for the same data instance. For the swines data instance (Table 5), the cost of grid electricity of the incumbent solution is -5425 with Strategy MP3. With Strategy MP2, the cost of grid electricity of the incumbent solution is -5160. Of the three strategies, Strategy MP1 gives the worst value of the cost of grid electricity of the incumbent solution, i.e., -4064. Strategy MP1 is not good because the search constantly updates the current solution with a poorer solution. 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 minimised (Strategy MP4). The results are compared with those of Strategy MP3 (Tables 4 and 5). 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 the month with the least infeasible solution is selected for optimisation. The incumbent solution with Strategy MP3, is better than with Strategy MP4 for both data instances. $-20,545$ is obtained as the cost of grid electricity with Strategy MP3 and $-16,691$ with Strategy MP4, for the cows data instance. For the swines data instance, -5425 is obtained as the cost of grid electricity of the incumbent solution, with Strategy MP3 and -5263 with Strategy MP4. 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).

Pseudocode 4: Strategy MP3

for iter = 1 : 12 **do**
  iter_m $\leftarrow$ 0
  Select a variable x_i^m for optimisation
  Perform Tabu Search
  Evaluate iterative solution $S^{\text{current}'}$
  while $S^{\text{current}'} > S^{\text{current}}$ and iter_m ≤ 12 **do**
    while $i \leq 3$ and $S^{\text{current}'} > S^{\text{current}}$ **do**
      Select a variable x_i^m for optimisation
      Perform Tabu Search
      Evaluate iterative solution $S^{\text{current}'}$
      Select the index of the next variable to be optimised $i \leftarrow i + 1 (\text{mod } 3)$
    end while
    Select the next month for which to carry out the optimisation $m \leftarrow m + 1$
    Select the index of the next variable to be optimised $i \leftarrow i + 1 (\text{mod } 3)$
    Select a variable x_i^m for optimisation
    Perform Tabu Search
    Evaluate iterative solution $S^{\text{current}'}$
    iter_m $\leftarrow$ 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 optimisation $m \leftarrow m + 1$
  Select the index of the next variable to be optimised $i \leftarrow i + 1 (\text{mod } 3)$
end for

This section has explained experiments carried out to investigate the Tabu Search optimisation strategies developed. The best strategies are highlighted in bold text in Tables 4 and 5. This is where; the threshold of infeasibility is varied, with the initial threshold set to $S_{\text{o}}^{\text{infeas}} = -200$, for the cows data instance and $S_{\text{o}}^{\text{infeas}} = -205$, for the swines data instance, the objective function is evaluated by forming pareto incumbent solutions, as a first priority only feasible solutions are allowed during the phase for minimisation of cost, round robin of the months is done after sampling all the variables for a given month during the phase for minimisation of cost, and the month with the most infeasible solution is selected for optimisation during the phase for minimisation of infeasibility.

8 Conclusion

Adaptations to the basic Tabu Search, for application to the optimisation of BWECS have been developed in this paper. These adaptations have been developed to handle constraints, multi-objectives, multi-periods and to perform diversification. The following is a highlight of the conclusions arrived at from the experiments undertaken, and the applications of the Tabu Search Algorithm developed.

8.1 Conclusions from the Experiments

Experiments were done to test the adaptations developed. It was found out that for optimisation of BWECS, constraints are best handled by alternating between allowing feasible and infeasible solutions. The algorithm was split into two phases. In Phase 1, cost was minimised whereas in Phase 2 infeasibility was minimised. Thresholds for minimisation of infeasibility and the number of iterations for which each phase is applied were experimented with. The best strategy is to set an initial threshold of infeasibility of $S_o^{\text{infeas}} = -200$ for the cows data instance and $S_o^{\text{infeas}} = -205$ for the swines data instance, and reduce these gradually as the optimisation proceeds. In addition minimisation of cost and infeasibility should be carried out alternately for a different number of iterations. The optimum number of iterations for which each phase is to be applied is dependent on the data instance. Only feasible solutions should be allowed during the optimisation, as a first priority. Diversification should be applied by performing 3 consecutive restarts with the incumbent solution, if the incumbent solution does not improve for a given number of iterations, to be determined for each data instance. 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. Different multi-period optimisation strategies should be applied for the two different phases of the Tabu Search algorithm. During the phase of minimisation of cost, round robin of the months should be carried out, however the best strategy is to update the current solution with improving solutions only. If there are no improving solutions, another variable is optimised for the same period, until all the variables have been optimised for that period. If the current solution does not improve for 12 periods, then it can be updated with a non-improving solution. During the phase of minimisation of infeasibility, the optimisation should be carried out for the period with the most infeasible solution.

8.2 Applications of the Tabu Search Adaptation Developed

The Tabu Search algorithm developed in this paper has been applied to the dimensioning of a BWECS to determine the maximum revenue that can be obtained for a given herd size Namuli et al. [4]. The adapted Tabu Search was also used as a tool to determine the threshold herd size at which BWECS become commercially viable Namuli et al. [27]. A tool for the analysis of the commercial viability of BWECS is currently unavailable. This is therefore an important contribution to the planning of programs for promotion of installation of BWECS on rural farms.

The Tabu Search algorithm developed can also be applied to multi-period scheduling problems with the objective of minimising cost. A work shift scheduling problem where the objective is to minimise the cost of overtime is an example. Infeasible solutions will be generated due to constraints like the number of hours one can work for in a week, and thus the constraint handling technique can be applied. This scheduling problem is also a multi-objective one as the objective function can be split into cost components namely:cost of overtime, cost of payment in lieu of leave and cost of temporary staffing.

Appendix

This section summarises the results of the experiments described in Section 7. Table 4 gives results of the experiments for the cows data instance whereas Table 5 gives results of the experiments for the swines data instance.

Table 4: Experiments for Adapted Tabu Search Optimisation - Cows Data Instance

	iter=50			iter=100			iter=200			Stopping Condition			
	$S^{current}$	S^{infeas}	S^{inc}	$S^{current}$	S^{infeas}	S^{inc}	$S^{current}$	S^{infeas}	S^{inc}	$S^{current}$	S^{infeas}	iter	S^{inc}
C1 S^{infeas}=-500	14 -18278	-873	0 -16130	16 -18197	-492	0 -16130 23 -16854	0 -19091	-500	0 -16130 23 -16854	25 -19949	-465	274	0 -16130 23 -16854
C1 S^{infeas}=-200	11 -17071	-128	0 -16130	19 -18302	-201	0 -16130 17 -16525 18 -16627 19 -16730 21 -16834	110 -16410	-4213	0 -16130 17 -16525 18 -16627 19 -16730 21 -16834	52 -14318	-245	344	0 -16130 17 -16525 18 -16627 19 -16730 21 -16834
C1 S^{infeas}=-100	13 -17099	-90	0 -16130	13 -15410	-155	0 -16130	45 -13739	-99	0 -16130	65 -13022	-139	262	0 -16130
C1 S_o^{infeas}=-500	14 -18278	-873	0 -16130	16 -18197	-492	0 -16130 23 -16854	35 -19081	-395	0 -16130 23 -16854	27 -19209	-390	274	0 -16130 23 -16854
C1 S_o^{infeas}=-300	12 -17193	-279	0 -16130	16 -17883	-188	0 -16130 18 -16618	38 -16682	-200	0 -16130 18 -16618	51 -12937	-219	265	0 -16130 18 -16618
C1 S_o^{infeas}=-100	13 -17099	-90	0 -16130	24 -15786	-134	0 -16130	45 -12428	-92	0 -16130	49 -13764	-32	262	0 -16130
C1+D1 S_o^{infeas} =-200	**11 -17071**	**-128**	**0 -16130**	**19 -18199**	**-201**	**0 -16130 17 -16525 18 -16627 20 -16678 21 -16884**	**113 -13591**	**-343**	**0 -16130 17 -16525 18 -16627 20 -16678 21 -16884**	**57 -14713**	**-101**	**347**	**0 -16130 17 -16525 18 -16627 20 -16678 21 -16884**
C2+D2 max_iter_opt =50 max_iter_feas =25	**11 -17071**	**-128**	**0 -16130**	**19 -18199**	**-201**	**0 -16130 17 -16525 18 -16627 20 -16678 21 -16884**	**18 -20087**	**-127**	**0 -16130 17 -16525 18 -16627 20 -16678 21 -16884**	**42 -17631**	**0**	**445**	**0 -16130 17 -16525 18 -16627 20 -16678 21 -16884 33 -19369 38 -19407 39 -19504**
C2 max_iter_opt =50 max_iter_feas =50	11 -17071	-128	0 -16130	29 -14412	-16	0 -16130 17 -16525 18 -16627 19 -16730 20 -16782	58 -16244	-14	0 -16130 17 -16525 18 -16627 19 -16730 20 -16782	35 -18934	-265	230	0 -16130 17 -16525 18 -16627 19 -16730 20 -16782
C2+MP3+ MOBJ1 max_iter_opt =75 max_iter_feas =50	**11 -17071**	**-128**	**0 -16130**	**36 -14708**	**-17**	**0 -16130 25 -17183 26 -17285**	**19 -20050**	**-184**	**0 -16130 25 -17183 26 -17285**	**50 -18374**	**-96**	**400**	**0 -16130 25 -17183 26 -17285 27 -19651 33 -20022 35 -20230 37 -20335 38 -20439 39 -20545**
C2 max_iter_opt =75 max_iter_feas =75	11 -17071	-128	0 -16130	36 -14708	-17	0 -16130 25 -17183 26 -17285	32 -19070	-201	0 -16130 25 -17183 26 -17285 37 -17913 39 -17918 40 -18162 41 -18212	40 -17416	-15	300	0 -16130 25 -17183 26 -17285 34 -17323 35 -18048 40 -18162 41 -18212
C3	11 -17071	-128	0 -16130	41 -14096	-23	0 -16130 25 -17183	47 -18865	-174	0 -16130 25 -17183 39 -18208 41 -18308	43 -18565	-152	515	0 -16130 25 -17183 39 -18208 41 -18308
MOBJ2	18 -16754	-237	0 -16130	36 -13912	-141	0 -16130	20 -17332	-267	0 -16130	29 -17289	-153	273	0 -16130
MP1	17 -18329	-174	0 -16130	52 -14409	-125	0 -16130	31 -18500	-226	0 -16130	54 -17438	-103	380	21 -17169
MP2	13 -16634	-385	0 -16130	35 -12305	-3	0 -16130	20 -19530	-200	0 -16130	22 -18142	-726	344	0 -16130 6 -17302 7 -17534 9 -17404
MP4	13 -17285	-192	0 -16130	28 -16564	-7	0 -16130	23 -17986	-182	0 -16130	44 -16320	-806	399	0 -16130 10 -16157 12 -16519 13 -16691

Table 5: Experiments for Adapted Tabu Search Optimisation - Swines Data Instance

	iter=50			iter=100			iter=200			Stopping Condition			
	$s^{current}$	s^{infeas}	s^{inc}	$s^{current}$	s^{infeas}	s^{inc}	$s^{current}$	s^{infeas}	s^{inc}	$s^{current}$	s^{infeas}	iter	s^{inc}
C1 s^{infeas} =-500	1021 -4327	-500	1071 -4064	1037 -4131	-576	1051 -3372 1071 -4064	973 -3844	-502	1051 -3372 1071 -4064	990 -3534	-500	263	1051 -3372 1071 -4064
C1 s^{infeas} =-205	1065 -4866	-6	1071 -4064	1044 -5515	-316	1066 -4630	1018 -3253	-2023	1066 -4630	1047 -1664	-214	335	1066 -4630
C1 s^{infeas} =-100	1051 -4928	-542	1071 -4064	1049 -4926	-101	1062 -4209	1034 -4266	-148	1061 -4016 1062 -4209	1006 -1347	-442	365	1061 -4016 1062 -4209
C1 s^{infeas}_o =-500	1021 -4327	-500	1071 -4064	927 -1572	-576	1051 -3372 1071 -4064	927 -1572	-421	1051 -3372 1071 -4064	926 -1699	-411	263	1051 -3372 1071 -4064
C1 s^{infeas}_o =-250	1019 -5308	-225	1071 -4064	1048 -3897	-291	1071 -4064 1059 -3799	1079 -4795	-273	1071 -4064 1059 -3799	1044 -2731	-1	479	1044 -3069 1056 -4348
C1 s^{infeas}_o =-100	1051 -4928	-542	1071 -4064	1040 -4679	-253	1062 -4209	970 -5087	-5235	1058 -3959 1062 -4209	983 -1272	-494	262	1058 -3959 1062 -4209
C1+D1 s^{infeas}_o =-205	**1065 -4866**	**-6**	**1071 -4064**	**1045 -4844**	**-401**	**1066 -4630**	**1018 -4511**	**-191**	**1061 -4240 1066 -4630**	**1035 -2130**	**-10**	**362**	**1042 -2062 1061 -4240 1066 -4630**
C2+D2 MOBJ2 +MP3 max_iter_ opt=50 max_iter_ feas=25	**1065 -4866**	**-6**	**1071 -4064**	**1045 -4844**	**-401**	**1066 -4630**	**1032 -4529**	**-32**	**1044 -4472 1045 -4763**	**1037 -4437**	**-270**	**349**	**1041 -4873 1049 -5425**
C2 max_iter_ opt=50 max_iter_ feas=50	1065 -4866	-6	1071 -4064	1066 -4488	0	1066 -4630	1024 -4299	0	1024 -4375 1066 -4630	1063 -3873	0	277	1024 -4375 1024 -4729
C2 max_iter_ opt=75 max_iter_ feas=50	1065 -4866	-6	1071 -4064	1059 -3522	0	1059 -3922 1071 -4064	1013 -4739	-193	1059 -3922 1071 -4064	1029 -3905	0	211	1029 -4123
C2 max_iter_ opt=75 max_iter_ feas=75	1065 -4866	-6	1071 -4064	1059 -3522	0	1059 -3922 1071 -4064		n/a		1028 -4363	-279	167	1059 -3922 1071 -4064
C3	1001 -6006	-230	1071 -4064	1029 -6030	-112	1054 -3599		n/a		1011 -6018	-161	198	1035 -4870
MOBJ2	1001 -5914	-198	1071 -4064	1031 -5258	-132	1056 -4739		n/a		1028 -4225	-34	195	1041 -5255
MP1	1038 -2710	-329	1071 -4064	1017 -3385	-222	1062 -3553 1071 -4064	1036 -1504	-321	1062 -3553 1071 -4064	1054 -2499	0	218	1062 -2499 1062 -3553 1071 -4064
MP2	1003 -5924	-205	1071 -4064	1019 -4749	-118	1029 -4551	977 -4275	-182	1016 -5160	999 -4047	0	203	999 -4047 1016 -5160
MP4	1001 -5914	-198	1071 -4064	1026 -5393	-136	1021 -2806 1055 -3614 1064 -3975 1065 -4870 1067 -4983 1069 -5020	1073 -2752	-81	1021 -2806 1049 -3254 1055 -3614 1064 -3975 1065 -4870 1067 -4983 1069 -5020 1070 -5263	1071 -555	0	214	1021 -2806 1049 -3254 1055 -3614 1064 -3975 1065 -4870 1067 -4983 1069 -5020 1071 -5263

References

[1] A. Rentizelas, I. Tatsiopoulos, A. Tolis, An Optimization Model for Multi-biomass Tri-generation Energy Supply, Biomass and Bioenergy 33 (2009) 223-233.

[2] M. Bruglieri, L. Liberti, Optimal Running and Planning of a Biomass-based Energy Production Process, Energy Policy 36 (2008) 2430-2438.

[3] ILOG, ILOG CPLEX 10.0 User's Manual, 2005. ILOG S.A, Gentilly, France.

[4] R. Namuli, B. Jaumard, A. Awasthi, P. Pillay, Optimisation of Biomass Waste to Energy Conversion Systems for Rural Grid-Connected Applications, Applied Energy (2012). In Press.

[5] D. Batstone, J. Keller, I. Angelidaki, S. Kalyuzhnyi, S. Pavlostathis, A. Rozzi, W. Sanders, H. Siegrist, V. Vavilin, Anaerobic Digestion Model No. 1 (ADM1), 2002. IWA Publishing, 2002. Scientific and Technical Report. 1 900222 78 7.

[6] National Renewable Energy Laboratory, Advanced Vehicle & Fuels Research, http://www.nrel.gov/vehiclesandfuels/vsa/related_links.html#advisor, 2002. Accessed December 2009.

[7] N. Mohan, Advanced Electric Drives Analysis, Control and Modeling using Simulink, MNPERE, Minneapolis, pp. 3–1, 3–26.

[8] F. Glover, Tabu Search - Part I, ORSA Journal on Computing 1 (1989) 190–206.

[9] F. Choobineh, E. Mohebbi, H. Khoo, A Multi-objective Tabu Search for a Single-machine Scheduling Problem with Sequence-dependent Setup Times, European Journal of Operational Research 175 (2006) 318-377.

[10] S. Kulturel-Konak, A. Smith, B. Norman, Multi-objective Tabu Search Using a Multinomial Probability Mass Function, European Journal of Operational Research 169 (2006) 918-931.

[11] A. Baykasoglu, Applying Multiple Objective Tabu Search to Continuous Optimization Problems with a Simple Neighbourhood Strategy, International Journal for Numerical Methods in Engineering 65 (2006) 406-424.

[12] J. Korsvik, K. Fagerholt, A Tabu Search Heuristic for Ship Routing and Scheduling with Flexible Cargo Quantities, Journal of Heuristics 16 (2008) 117-137.

[13] A. Lim, H. Qin, J. Xu, Random Move Tabu Search for Freight Proportion Allocation Problem, in: 20^{th} IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 307–314.

[14] A. Mantawy, S. Soliman, M. El-Hawary, The Long-term Hydro-Scheduling Problem - A New Algorithm, Electrical Power Systems Research 64 (2003) 67-72.

[15] N. Nayak, C. Rajan, Hydro-thermal Commitment Scheduling by Tabu Search Method with Cooling-Banking Constraints, SWARM, EVOLUTIONARY AND MEMETIC COMPUTING, Lecture Notes in Computer Science 6466 (2010) 739-752.

[16] J. Tippayachai, W. Ongsakul, I. Ngamroo, Non-convex Economic Dispatch by Enhanced Tabu Search Algorithm, in: IEEE Power Engineering Society General Meeting, volume 2, pp. 1–6.

[17] M. Bolduc, G. Laporte, J. Renaud, F. Boctor, A Tabu Search Heuristic for the Split Delivery Vehicle Routing Problem with Production and Demand Calendars, European Journal of Operational Research 202 (2010) 122-130.

[18] E. Driouch, W. Ajib, M. Gaha, A Tabu Search Scheduling Algorithm For MIMO CDMA Systems, in: IEEE Global Telecommunications Conference - GLOBECOM, pp. 1–5.

[19] J. Brandao, A Tabu Search Algorithm for the Heterogeneous Fixed Fleet Vehicle Routing Problem, Computers & Industrial Engineering 38 (2011) 140–151.

[20] B. Vahdani, M. Zandieh, Scheduling Trucks in Cross-docking Systems: Robust Meta-heuristics, Computers & Industrial Engineering 58 (2010) 12–24.

[21] National Gas, Characteristics of LP Gas, http://www.nationalgasco.net/portals/0/Characteristics%20of %20LPGas.pdf, 2012. Accessed May 2012.

[22] P. Lusk, Methane Recovery from Animal Manures. The Current Opportunities Casebook, http://agrienvarchive.ca/bioenergy/download/methane.pdf, 1998.

[23] R. Burns, Status of Manure Anaerobic Digestion in the United States, in: International Conference on 21st Century Challenges to Sustainable Agri-Food Systems: Biotechnical, Environment, Nutrition, Trade and Policy, pp. 476–481.

[24] NYSERDA, Renewable Portfolio Standard Customer-Sited Tier Anaerobic Digester Gas-to-Electricity Program, www.nyserda.org/funding/2138pon.pdf, 2011. Program Opportunity Notice (PON) 2138 accessed September 2011.

[25] NYSERDA, DG/CHP Integrated Data System. A.A. Dairy, http://chp.nyserda.org/facilities/details.cfm?facility=33, 2012. Accessed January 2012.

[26] M. Khakbazan, A Comparative Study of Energy Use in Hog Barns on the Prairies, http://www.usask.ca/agriculture/caedac/transportation/energy.PDF, 1999. Accessed May 2012.

[27] R. Namuli, B. Jaumard, P. Pillay, C. Laflamme, Threshold Herd Size for Commercial Viability of Biomass Waste to Energy Conversion Systems on Rural Farms, Technical Report, Les Cahiers du GERAD, G-2012-34, July 2012. HEC Montréal.