Biogas plant optimization using Genetic Algorithms and Particle Swarm Optimization

Christian Wolf[ϕ], Seán McLoone[+], Michael Bongards[*]

[ϕ] *Department of Electronic Eng. National University of Ireland, Maynooth, Maynooth, Co Kildare, Ireland, E-mail: wolf@gm.fh-koeln.de*

[+] *Department of Electronic Eng. National University of Ireland, Maynooth, Maynooth, Co Kildare, Ireland, E-mail: sean.mcloone@eeng.nuim.ie*

[*] *Institute of Automation and Industrial IT. Cologne University of Applied Sciences, Germany E-mail: bongards@gm.fh-koeln.de*

Abstract – **The optimization of agricultural biogas plants with respect to external influences and various process disturbances is essential for efficient plant operation. However, the optimization and control of such plants is a challenging problem due the underlying highly nonlinear and complex digestion processes. One approach to addressing this challenge is to exploit the flexibility and power of computational intelligence methods such as Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO). In this paper these methods are used in conjunction with a validated plant simulation model to optimize substrate feed mix, a key factor in stable and efficient biogas production. Results show that an improvement of up to 20% in biogas production and substrate reduction can be achieved when compared to conventional manual operation. In addition, a comparison of the performance of GAs and PSO reveals that while both methods can achieve comparable results PSO has faster convergence and hence is preferred for this application.**

Keywords – **Intelligent process optimization, PSO, Genetic Algorithm, biogas plant**

I INTRODUCTION

In the past twenty years the rise in worldwide energy production using the conversion of biomass materials to methane in biogas plants has its origin in large-scale government aid. Renewable energy laws guaranteeing lucrative electricity remuneration rates and funding for biogas plant construction, support a booming biogas sector [1].

This new market for renewable energy from energy crops and municipal organic waste is struggling in Germany due to reducing governmental support and rising prices for biomass. Advanced control and optimization of biogas plants is therefore becoming more and more important as inefficient plant operation can no longer be afforded. This need goes along with increasing utilisation of newly developed online-measurement and process monitoring systems.

The difficulties with biogas plant operation are primarily due to the complexity of anaerobic digestion processes. Physical, chemical and biological processes run simultaneously and are furthermore affected by external influences such as local weather conditions, environmental changes and changes in daily influent load. The combination of multiple complex processes and their dependencies on external influences make it difficult to develop an automated control and optimization strategy which is both reliable and effective. Reliability is of particular importance for agricultural biogas plants, where permanent attention and supervision by an operator is not practical.

To develop such an optimal control strategy, it is critical to monitor anaerobic digestion processes as closely and accurately as possible to enable estimation of process states and to detect unstable process states in time and if possible in advance. This facilitates the implementation of more effective control measures. Online process monitoring is rare at most agricultural and even some industrial biogas plants because of high acquisition and maintenance costs and a lack of reliability during *in situ* measurements as proven by Kujawski *et al.* (2007) [2]. This necessitates the development of new methods for designing and optimizing advanced control strategies before testing them in practice.

Therefore, advanced control and optimization strategies like Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO) are developed and tested using a dynamic simulation model for anaerobic digestion, the Anaerobic Digestion Model No. 1 (ADM1) [3].

This paper introduces the use of GAs and PSO to optimize biogas plant operation. In particular, the substrate feed (total amount and mixture) is optimized, taking into account constraints such as amount of total solids and digester load. The flexibility of these computational intelligence (CI) methods makes them perfectly suited to the non-convex multi-objective nature of the optimisation problems posed by these complex systems.

Section 2 introduces the basic principles of GAs and PSO while section 3 describes the functionality of biogas plants and the main anaerobic digestion processes involved. Section 4 presents the optimization results obtained using a calibrated simulation model and the CI methods mentioned above, comparing and analyzing optimization time and repeatability of the obtained results. Furthermore, the applied fitness function is explained along with other optimization parameters. The conclusion sums up achieved results and gives a final evaluation of the optimization strategies.

II GENETIC ALGORITHMS AND PARTICLE SWARM OPTIMIZATION

Genetic Algorithms and Particle Swarm Optimization are both methods, designed to search among a collection of possible solutions for a designated solution. The characteristic of these CI methods, compared to analytic optimization methods, is the emulation of a natural phenomenon. In this case the inspiring natural examples are genetic evolution and the emergent complex patterns observed in the collective movement of many species (e.g. bird flocking, animal herding and fish schooling). The strength of these CI methods comes from their computational parallelism and their capacity to adapt to their environment. This combination allows them to effectively search for solutions of complex optimization problems in complex and large search spaces [4] [5].

a) Genetic Algorithms

GAs are described using biological terminology which differentiates them from other evolutionary computation methods. The most important terms are as follows:

- *Chromosome:* A representation of a possible solution to an optimization problem where parameter values are encoded using either binary, real-valued or tree encoding.
- *Genes:* Groups of bits or real values which encode one particular element of a possible solution (chromosome).
- *Crossover*: An operation in which genetic material is exchanged between two different chromosomes (parents) to generate new chromosomes (children).
- *Mutation*: An operation which randomly changes parts of a gene in a chromosome at randomly chosen places.
- *Population*: The set of chromosomes used to explore the optimization space. This can either be fixed or vary as optimization progresses.
- *Generation*: One optimization cycle of a GA.

As described in Mitchell [6] a basic genetic algorithm works as follows:

1. Generate a random initial population of n chromosomes.
2. Calculate the fitness $f(c)$ of each chromosome c in the population.
3. The following steps are repeated to obtain a new population with n offspring:
   a. Selection of two parent chromosomes according to the calculated fitness.
   b. Cross over the parents at a chosen point with a crossover probability p_c and create two offspring.
   c. Mutation of the offspring at every position with mutation probability p_m and addition of the mutated chromosomes to the new population.
4. Exchange the current population against the newly generated population.
5. Start back at step 2.

A single iteration of the described steps represents one generation.

The most critical parameter in a GA is the fitness function which evaluates all possible solutions to the optimization problem. If fitness function performance is poor, optimization results will also be poor. Finding the appropriate fitness function for a complex optimization problem is often the most difficult task. Furthermore, algorithm performance is sensitive to the choice of generation and population size, crossover and mutation functions and chromosome encoding; hence these parameters must also be carefully chosen for optimum results.

b) Particle Swarm Optimization

PSO is a population-based evolutionary computation algorithm for problem solving which simulates social behaviour in swarms. Individuals of a swarm (particles) exchange information with their neighbours which has an influence on their behaviour and finally on the movement of the whole swarm. This information exchange allows a swarm to move towards the most interesting site in a search space, as information about interesting sites is slowly propagated to the whole swarm. Thus, in PSO, the behaviour of each particle in a swarm is governed by two basic principles: *particle communication* and *particle movement*.

Particle communication is controlled by the parameter K_N, defined as the number of neighbouring particles a particle is exchanging information with. To guarantee sufficient particle communication K_N has to be carefully selected. If it is too small the propagation of important information to all particles might take too long and if too large, particles might get stuck in a local optimum. The probability $P_r(t)$ for a particle to be reached at least once after the t^{th} run is described by the following formula [7], where N is the number of particles and K_N is the number of neighbours for information exchange.

$$P_r(t) = 1 - \left(1 - \frac{1}{N}\right)^{K_N{}^t} \tag{1}$$

As the probability increases quickly with t, even with a small number of neighbours, K_N, information propagation throughout the whole swarm can be rapid.

The movement of a PSO particle in search space is defined in terms of its position vector $\mathbf{x}(t)$ and three parameter vectors:

- *Velocity* ($\mathbf{v}$): The speed at which the particle moves through the search space.
- *Personal best position* ($\mathbf{p}$): The best position a particle has currently found.
- *Global best position* ($\mathbf{g}$): The best position found by informants of a particle.

Using these parameters a particle's position and velocity are updated at the t^{th} iteration as follows:

$$\begin{aligned} \mathbf{v}(t+1) &= c_1\mathbf{v}(t) + c_2(\mathbf{p}(t) - \mathbf{x}(t)) + c_3(\mathbf{g}(t) - \mathbf{x}(t)) \\ \mathbf{x}(t+1) &= \mathbf{x}(t) + \mathbf{v}(t) \end{aligned} \tag{2}$$

Weights c_1 to c_3 are constants that determine the importance of the different vectors:

- c_1 represents the confidence of a particle in its direction of movement.
- c_2 and c_3 represent the confidence of a particle in its personal best position and its best reported global position, respectively.

Using these mechanisms a swarm of particles moves through a search space looking for an optimal solution to a defined optimization problem. In a similar fashion to GAs, at each iteration all particles are evaluated using a fitness function and this information is used to update the current position, personal best position and global best position of each particle.

III BIOGAS PLANT OPERATING PRINCIPLES

A biogas plant is designed to produce methane (CH_4) and carbon dioxide (CO_2) from organic material in the absence of oxygen. This conversion is called anaerobic digestion and its end-product is biogas. While there are big differences between the scale and operation of industrial and agricultural biogas plants, the basic plant design and components are essentially the same. Each biogas plant consists of one or several storage tanks for organic material, a fermentation tank and a final storage tank for fully digested sludge as shown in Figure 1.

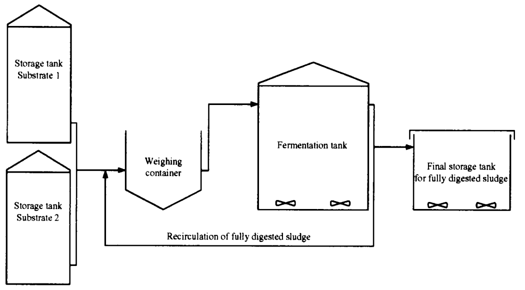

Figure 1. General layout of a Biogas plant

The fermentation tank has two phases, a gas and a liquid phase where organic material is digested by anaerobic bacteria during various complex bio-chemical processes. There are four processes involved in biogas production [8].

1. *Hydrolysis* breaks complex organic structures open to make them accessible to the following processes.
2. *Acidogenesis* produces organic acids as well as hydrogen, carbon dioxide, different alcohols and a small amount of acetic acid out of organic material.
3. *Acetogenesis* uses organic acids, hydrogen and carbon dioxide to produce acetic acid.
4. *Methanogenesis* produces methane from acetic acid and to a lesser extent from hydrogen and carbon dioxide.

All processes involved in anaerobic digestion make different demands on pH-value as well as relying on the full functionality of the other processes. This sensitive balance between fermentation processes is difficult to maintain.

In practice, as illustrated in Figures 2 and 3, variations in the throughput of substrate and concentration of organic total solids (oTS) in the substrate are the main factors that influence process stability and biogas production in agricultural biogas plants. The efficient optimization and control of these plants can therefore be realized by adapting the substrate feed according to the state of anaerobic digestion. Currently, two main strategies exist for biogas plant operation [9]:

Low substrate feed. The total amount of substrate fed to a biogas plant is reduced to assure enough buffer capacity against disturbances in the fermentation tank. Hence, biogas production and plant efficiency decrease.

High substrate feed. The total amount of substrate fed to a biogas plant is increased to achieve maximum biogas production and plant efficiency. A sophisticated control system with expensive online measurement systems is crucial to maintain process stability.

Agricultural biogas plants, in particular, are generally operated at low substrate feeds, as advanced control and measurement systems are not feasible. However, the advent of new cheaper online-measurement technologies coupled with the recent development of dynamic biogas simulation models, makes agricultural biogas plant optimization possible.

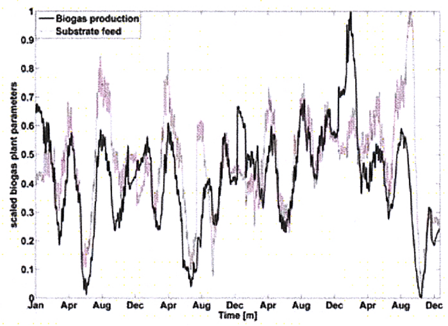

Figure 2. Development of biogas production against total substrate feed from 2004 to 2007 of a full-scale biogas plant

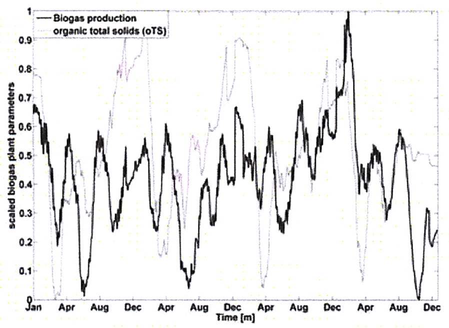

Figure 3. Development of biogas production against concentration of organic total solids from 2004 to 2007 of a full-scale biogas plant

IV OPTIMIZATION RESULTS

Testing different substrate feeds at full-scale biogas plants is difficult and often not practical. In particular, substrate feed variations can affect process stability and cause extreme situations that cannot be handled easily. A dynamic simulation model, which includes the inhibitions caused by fast changes in substrate quantity and quality, is, thus, an appropriate tool for testing different optimization strategies. Here, a validated simulation model of a full-scale agricultural biogas plant is used to investigate the optimization of substrate mix using GAs and PSO.

a) Dynamic Simulation Model

The dynamic simulation model used for substrate feed optimization is built according to the design of an agricultural reference biogas plant near Frankfurt (Germany). Using basic online-measurements, the simulation model is calibrated to match biogas production and quality corresponding to different substrates. The substrate feed of the reference plant consists of Cob Corn Mix (CCM), Rye and pig manure.

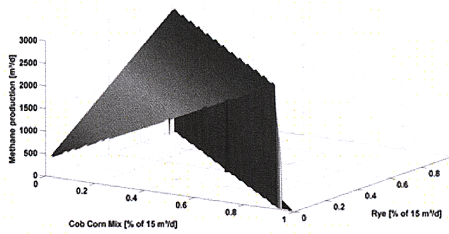

Figure 4. Simulated methane production for increasing substrate feed

The substrate feed can be increased up to a certain level until methane production collapses, which depends on the composition of the substrate feed (Figure 4). Different substrates might increase or even decrease methane production, due to their ingredients. The simulated methane production shown in Figure 4 matches the behavior of the full-scale reference biogas plant.

The breakdown of methane production can be attributed to many factors but the most common are:

- Critical concentration of organic acids
- High ammonia concentration
- High hydrogen sulfide concentration
- Critical concentration of heavy metals

Critical process states caused by acid and ammonia inhibition are captured by the simulation model.

b) GA and PSO Fitness Function

The key performance parameters of the fitness function are how far a substrate has been digested (D_s), energy consumption for pumps and heating of a fermentation tank (E), gas quality (G_q) and amount (G_a), digester load (L_d) and penalties for exceeding pH (P_{pH}), substrate (P_s), or total solids limits (P_{TS}). A weighted sum of these parameters, where each is scaled and multiplied with constant factors c_1 - c_8, constitutes the fitness function f.

$$f = \begin{cases} c_1 D_s + c_2 E + c_3 G_q + c_4 G_a + \\ c_5 L_d + c_6 P_{pH} + c_7 P_s + c_8 P_{TS} \end{cases} \quad (3)$$

The weights for parameters G_a and L_d are selected to be greater than the others, because the first priority of substrate feed optimization is to tap the full potential of a biogas plant.

c) Genetic Algorithm Design

Design parameters such as population, number of generations, crossover function and mutation rate require careful selection in order to obtain good optimization results. Table 1 shows the parameters used in the GA for substrate optimisation. The GA has been created using the standard GA Matlab toolbox [10].

Table 1. Parameters for the GA

GA parameters	Value
Number of generations	200
Population size	60
Probability of crossover	0.7
Crossover strategy	Intermediate
Mutation strategy	Adaptive feasible
Selection strategy	Elitism + Stochastic uniform

The crossover strategy *Intermediate* creates children as a weighted average of the parent solutions according to the following equations,

$$C_1 = P_1 + r\,(P_2 - P_1)$$
$$C_2 = P_2 + r\,(P_1 - P_2) \quad (4)$$

where C_1 and C_2 are children of parent solutions P_1 and P_2, r is a uniform random crossover factor in the range [0, 1]. The mutation strategy *adaptive feasible* randomly generates directions, which are adaptive according to the last successful or unsuccessful generation. The step length along each direction is chosen automatically matching existing ancillary conditions.

d) Particle Swarm Optimization Design

PSO was implemented using the parameters shown in Table 2 and is based on a free Matlab toolbox [11]. The values for personal and global best influence represent how much confidence a particle has in its personal best position and in the best position it has ever heard of, while the initial and final inertia weights reflect how much confidence a particle has in its current position.

Table 2. Parameters for the PSO

GA parameters	Value
Number of runs	400
Number of particles	30
Personal best influence	2
Global best influence	2
Initial inertia weight	0.9
Final inertia weight	0.6

e) Optimization Results

Since the same fitness function is employed with both optimization strategies direct comparison of the results is possible. Figure 5 shows the results for substrate feed optimized for a daily biogas production, which is equivalent to a power output of 300 kW. The corresponding evolution of the GA and PSO fitness function averaged over 10 optimization runs are depicted in Figure 6.

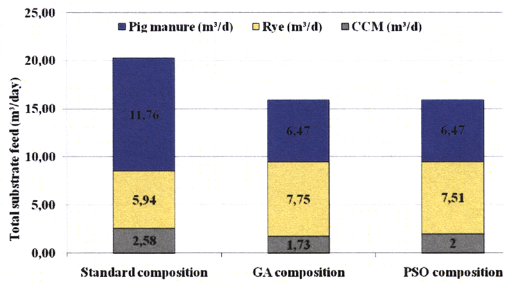

Figure 5. Comparison of standard and optimized substrate feed for an energy production of 300kW

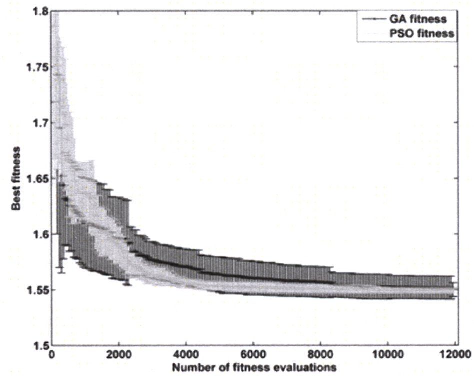

Figure 6. Improvement in the best fitness as a function of the number of fitness function evaluations for (a) GAs; and (b) PSO (average and standard deviations shown are based on 10 optimization runs)

As can be seen GA and PSO achieve comparable results with the average final fitness of the PSO composition marginally superior to the GA (1.549 compared to 1.550). In addition, the variability of results obtained with PSO is much less than obtained with GAs (40%), hence PSO has a much greater likelihood of generating good optimization results in a given run.

When comparing the computational performance of PSO and GAs it has to be considered that each method requires a different number of fitness function evaluations per generation and takes a different number of generations to converge. For example the best GA result required 196 generations with 60 simulations for each generation to reach the best fitness, whereas the best PSO run needed 337 iterations with 30 simulations each. This resulted in a total of 11760 biogas-plant simulations for the GA and 10110 simulations for PSO, which clearly highlights that PSO is approximately 14% faster than the GA in this instance. Compared to the gain in final fitness this saving in simulations is more important (each simulation takes approximately 10 seconds on a 2.4 GHz Quad-Core Pentium Processor). Thus PSO is the preferred optimization strategy for this application.

The overall results show that there is lots of potential for improvement in biogas plant operation if the substrate feed is optimized. Figure 5 shows that the overall substrate feed can be reduced by 21% achieving the same energy production, no matter, which optimization strategy is used. Furthermore, energy consumption for heating the fermentation tank and pumping substrate as well as the retention time of the substrate in the fermentation tank are reduced.

The global search capability of GAs and PSO, finding a very good solution to a complex problem, becomes obvious in Figure 7. This shows the methane production resulting from various solutions generated by the PSO during 400 generations and highlights that both very low and very high substrate feeds are evaluated.

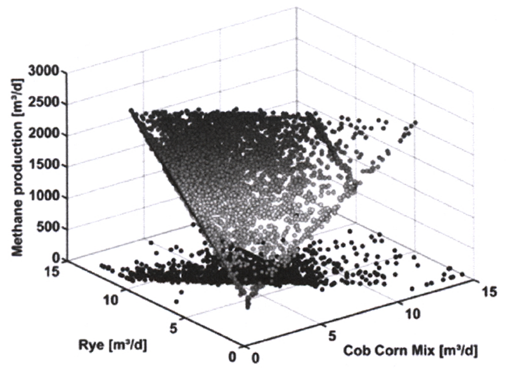

Figure 7. Methane production in dependency of PSO solutions

V CONCLUSIONS

The use of CI methods like GAs and PSO, in conjunction with a fully calibrated simulation model of a full-scale biogas plant, has been found to be an effective approach for optimizing substrate feeds. Optimization results show that an intelligent optimization strategy involving GA or PSO model-based optimization of substrate feed, can substantially improve the efficiency of biogas plants without compromising process stability. In particular, the reduction of substrate feed of 21% is very high, but this may differ from plant to plant as it very much depends on the optimization potential of individual biogas plants.

The direct comparison of a GA and PSO applied to the same application revealed interesting results, showing that both methods reach similar fitness values, but computation time to reach an optimum is significantly different. PSO has proven to be more efficient than GAs with the result that similar results to GAs can be achieved a lot faster using PSO.

VI REFERENCES

[1] BGB (Bürgerliches Gesetzbuch – German Civil Code) (2004); *Erneuerbare Energien Gesetz (Renewable Energy Sources Act) BGBI I*; 1999, p 2633, 07.12.1999, last time renewed, law from July 2004, BGBI I, page 1918.

[2] Kujawski O.; Wiese J.; König R.; Häck M. (2007); *Instrumentation, Control and Automation for Agricultural and Co-Digestion Biogas Plants – Yesterday, Today and Tomorrow*; 11th IWA World Congress on Anaerobic Digestion, Brisbane, Australia.

[3] Batstone D.J.; Keller J.; Angelidaki I.; Kalyuzhnyi S.V.; Pavlostathis S.G.; Rozzi A.; Sanders W.T.M.; Siegrist H.; Vavilin V.A. (2002); *Anaerobic Digestion Model No. 1, Scientific and technical report no.13*; IWA Task Group for Mathematic Modelling of Anaerobic Digestion Processes, IWA Publishing, London.

[4] Akjiratikarl C.; Yenradee P.; Drake P.R. (2007); *PSO-based algorithm for home care worker scheduling in the UK*; Computers & Industrial Engineering; Vol. 53 Issue 4, p559-583.

[5] Schwinning S.; Ehleringer J. R. (2001); *Water use trade-offs and optimal adaptations to pulse-driven arid ecosystems*; Journal of Ecology; Vol. 89, p464-480.

[6] Mitchell M. (1995); *An Introduction to Genetic Algorithms*; MIT Press, Cambridge.

[7] Clerc M. (2006); *Particle Swarm Optimization;* ISTE.

[8] Koppe, P., Stozek, A. (1993). *Municipal wastewater – its ingredients according to origin, Composition and reactions in waste water treatment processes including biosolid* (in German); Vulkan-Verlag 3. Auflage, Essen

[9] Zimmermann C. (2003): *Process control and management of agricultural biogas plants with dynamic plant operation* (in German); Ingenieurbüro für Entwicklung und Anwendung umweltfreundlicher Technik GmbH (UTEC); Bremen.

[10] Mathworks (2007); www.mathworks.com (accessed 27 January 2007)

[11] Birge B. (2006); Particle Swarm Optimization toolbox, www.mathworks.com/matlabcentral.