Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

JZUS

Ant colony system algorithm for the optimization of beer fermentation control

XIAO Jie (肖 杰)†, ZHOU Ze-kui (周泽魁), ZHANG Guang-xin (张光新)

(*National Key Lab of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China*)

†E-mail: seanxiaojie@zju.edu.cn

Received Sept. 23, 2003; revision accepted Oct. 12, 2003

Abstract: Beer fermentation is a dynamic process that must be guided along a temperature profile to obtain the desired results. Ant colony system algorithm was applied to optimize the kinetic model of this process. During a fixed period of fermentation time, a series of different temperature profiles of the mixture were constructed. An optimal one was chosen at last. Optimal temperature profile maximized the final ethanol production and minimized the byproducts concentration and spoilage risk. The satisfactory results obtained did not require much computation effort.

Keywords: Beer fermentation, Kinetic model, ACS algorithm, Optimization, Optimal temperature profile
doi:10.1631/jzus.2004.1597 **Document code:** A **CLC number:** TP20

INTRODUCTION

Long well known industrial beer fermentation is usually carried out in open loop conditions, using a predefined and empirically established temperature profile. The off-line measurement of the wort density, which is well correlated to the fermentable sugar concentration, is widely used on industrial scale for monitoring the fermentation. However, for optimization of operational conditions and process control purposes, we need a mathematical model for accurately simulating the industrial process under different operational conditions. There is no steady state in the process, and model equations are generally non-linear.

Further work based on the model is finding an optimal or pseudo-optimal operation regime, which calls for an efficient algorithm to fulfill this purpose. Ant algorithm is a new general-purpose heuristic algorithm, which was first applied in TSP (Traveling Salesman Problem) problems, inspired by the behavior of real ant colonies, in particular, by their foraging behavior. It is a population-based approach using positive feedback as well as greedy search. The effectiveness of its different versions as powerful tools for solving complicated optimization problems was demonstrated in a number of applications (Dorigo *et al.*, 2002), which gave us inspiration to apply it in dynamic process optimization problems.

In order to implement ant algorithm in optimization appropriately, two aspects should be taken into consideration:

1) Selecting a suitable version of ant algorithm (Ant Colony System-ACS algorithm was finally adopted in this paper);

2) Adapting algorithm to the specific optimization problem with some revisions to the original one. Enough attention should be paid to the stochastic greedy rule (the state transition rule) and the pheromone updating rule, which are the essences of the algorithm. The optimization of the

process (in forms of some mathematical equations) has been accomplished with the help of MATLAB, which is a high-performance language for technical computing.

PROCESS DESCRIPTION AND MATHEMATICAL MODEL

Beer fermentation exhibits complex characteristics. During the process, sugars are converted to ethanol. Without mechanical stirring, temperature is the most important manipulated intervention in the process. Low temperature makes the process time-consuming, and high temperature accelerates the process greatly. The later one (above 15 °C) will cause the loss of some ethanol efficiency, and multiply the risk of spoilage. Some byproducts (especially ethyl acetate and diacetyl), in addition to ethanol, are produced depending on temperature. They must be constrained within certain limits, in order to eliminate bad effects on flavor and aroma. The brewery uses a specific temperature profile (which is often confidential) to drive the process.

This work is aimed at finding an optimized temperature profile characterized by good ethanol efficiency, no spoilage risks, and being not over the concentration limits of ethyl acetate and diacetyl. For this purpose, a useful kinetic model (Andres-Toro *et al.*, 1998) was adopted.

It should be noted that there is a heating circuit in the experimental set-up, because the model parameter values were calculated from the experimental data obtained under isothermal runs (although the same model can be applied at the non-isothermal pilot plant scale). To keep a constant temperature, it is essential to have both cooling and heating circuits. However, in many breweries, fermentation tanks only have cooling circuits, so the increase of temperature relies on the heat released by the biochemical process itself. A conclusion can be drawn from this fact that if the optimized temperature profile could be used in real plants, it should obey the rule that once the temperature has been decreased, it cannot go up again (Increase Once Rule, in short). Because the temperature increase cannot be controlled in plants in the later parts of the fermentation process.

OPTIMIZATION PROBLEM STATEMENT

The optimization of fermentation process is a multi-objective optimization problem. An objective function used to evaluate how good the fermentation is, should take time and efficiency criteria, together with some constrains, into consideration. The original one (Andres-Toro *et al.*, 1997a) is a group of five functions; each of them has a sub-objective. To make values of functions comparable, some modifications to the original sub-objective parameters were done:

Ethanol production must be as high as possible:

Function I:

$$J_1 = +10C_e(t) \tag{1}$$

in which $C_e(t)$ can be calculated from the following functions:

$$\frac{dC_e(t)}{dt} = f\mu_e X_{act}(t) \tag{2}$$

$$f = 1 - \frac{C_e(t)}{0.5C_{s0}} \text{ and } \mu_e = \frac{\mu_{e0}C_s(t)}{k_s + C_s(t)} \tag{3}$$

$$\frac{dX_{act}(t)}{dt} = \mu_x X_{act}(t) - \mu_{dt}X_{act}(t) + \mu_{lag}X_{lag}(t) \tag{4}$$

$$\mu_x = \frac{\mu_{x0}C_s(t)}{0.5C_{s0} + C_e(t)} \tag{5}$$

$$\frac{dX_{sd}(t)}{dt} = \mu_{dt}X_{act}(t) - \mu_{sd}X_{sd}(t) \tag{6}$$

$$\frac{dC_s(t)}{dt} = -\mu_s X_{act}(t) \tag{7}$$

$$\mu_s = \frac{\mu_{s0}C_s(t)}{k_s + C_s(t)} \tag{8}$$

Diacetyl and ethyl acetate concentration should be limited:

Functions II and III

$$J_3 = -5.73\times\exp(11\times C_{dy}(t) - 11.51) \tag{9}$$

$$_4 = -1.16\times\exp(6.7\times C_{ea}(t) - 66.77) \tag{10}$$

in which $C_{dy}(t)$ and $C_{ea}(t)$ can be calculated from the next two functions:

$$\frac{dC_{ea}(t)}{dt} = Y_{eas}(-\frac{dC_s(t)}{dt}) \quad (11)$$

$$\frac{dC_{dy}(t)}{dt} = \mu_{dy}C_s(t)X_{act}(t) - \mu_{ab}C_{dy}(t)C_e(t) \quad (12)$$

Spoilage risk along the process:

Function IV:

$$J_2 = -\int_0^t 9.91\times10^{-17}\times\exp(2.31\times T)dt \quad (13)$$

Brisk changes in temperature should be penalized:

Function V:

$$J_5 = -\sum_{i=1}^{[t/\Delta t]}\frac{|T_i - T_{i-1}|}{\Delta t} \quad (14)$$

Grouping all the five functions:

$$J_{max}=J_1+J_2+J_3+J_4+J_5 \quad (15)$$

The value obtained depends on the temperature profile applied by the brewery along the fermentation process. That means a dynamic optimization problem: the objective function should be maximized by the application of a certain temperature profile.

For the application of ant colony system algorithm, the format of this objective function has to be changed:

$$J_{min}=CONS-J_{max} \quad (16)$$

where $CONS$ ($CONS>J_{max}$) is a positive integer, and J_{min} should be minimized. In addition, the temperature profile obtained should obey the Increase Once Rule.

OPTIMIZATION WITH ANT COLONY SYSTEM ALGORITHM

Ant colony optimization (ACO) is a particular successful research direction in ant algorithm. Ant colony system (ACS) algorithm, one version of ACO, was introduced by Dorigo and Gambardella (1997) to improve the performance of Ant System (Dorigo *et al.*,1996), that was able to find good solutions for only small TSP problems within a reasonable time. The ACS differs from AS in three main aspects: 1) the state transition rule provides a direct way to achieve balance between exploration of new edges and exploitation of a prior and accumulated knowledge on the problem; 2) the global updating rule is applied only to edges which belong to the best tour; 3) while ants construct a solution, a local pheromone updating rule (local updating rule, for short) is applied to shuffle the tours. In this way ants will make better use of pheromone information. These three modifications enable ACS to solve large problems effectively. With about 160 nodes for ants to visit, ACS algorithm has been chosen to deal with this optimization problem.

To apply ACS algorithm, a piecewise approximation of the temperature profile by a series of breakpoints is made. The breakpoints in a temperature profile are the visited nodes in an ants' tour, and are regularly spaced, every one hour. Between each time interval, the temperature is constant. Integer numbers between 0 °C and 16 °C are used to represent temperature and fixed fermentation time is 160 hours. Until now, the construction graph is clear. There are 161 nodes for each ant to build a complete tour (a temperature profile). Each node will be the integer temperature value at the end of each time interval. At the beginning of each time interval, an ant will decide the temperature in the next time interval. After this kind of discretization, J_{min} and other parameter values can be easily calculated at each end of the time intervals according to 16 equations listed above.

Fig.1 is an industrial temperature profile (Andres-Toro *et al.*, 1998). If the process follows this trajectory, and $CONS$ is set as 1000 (this value is decided according to some literature and it should be bigger than the suggested maximal value of J_{max}), values of J_{min} along the process are shown in Fig.2 showing that the value of J_{min} decreases as time goes on. So it is feasible to minimize J_{min} by ACS

approach. Here J_{min} is like tour length in TSP problems, and the only difference is that J_{min} increases negative values along the tour construction.

Informally, ants search the optimal temperature profile as follows: m ants are initially positioned on ($t(0)$, $T(0)$): it represents coordinate in construction graph, in which X axis represents time and Y axis represents temperature. Here, $t(0)$ is the beginning of the first time interval, and next time interval begins at $t(1)$. That means there are 161 X axis coordinates: $t(i)$, i=0,1,..., 160. Using 10 °C ($T(0)$ set as this value) as the beginning of the solution is recommended by the industrial temperature profile shown in Fig.1. When the program goes into the inner loop, each ant begins to build a tour (a feasible temperature profile). Take an ant k (k=1,2,...,m) for example, if it is now in node r: ($t(i)$,$T(i)$), i=0,1,...,160, it should choose the next node s: ($t(i+1)$, $T(i+1)$) to move to. There are 17 choices available: ($t(i+1)$, 0), ($t(i+1)$, 1), ($t(i+1)$, 2), ..., ($t(i+1)$, 16). $N(r)$ is the set consisting of these choices. $J_{min}(i+1, j)$, j=0,1,...,16, can be easily calculated. Each of them corresponds to a choice of node ($t(i+1)$, j). Then the state transition rule is applied:

$$s=\begin{cases}\arg\max_{u\in N(r)}\{\tau(r,u)[\eta(r,u)]^{\beta}\} & \text{if } q\le q_0\\ S & \text{otherwise}\end{cases} \quad (17)$$

where $\tau(r, u)$ is the amount of pheromone trail on edge. There are 160 time intervals, and in each of it, a matrix (17×17) is used to record the intensity of the trail. $\eta(r, u)$ is a heuristic function given by the next Eq.(18):

$$\eta(r,u)=\begin{cases}\dfrac{1}{J-\min_{j\in[0,\cdots,16]}(J+1)} & \text{if } \min_{j\in[0,1,\cdots,16]}(J)\le 0;\\ \dfrac{1}{J} & \text{otherwise}\end{cases} \quad (18)$$

where $J=J_{min}(i+1,j)-J_{min}[i,T(i)]$.

β is a parameter which weighs the relative importance of pheromone versus heuristic value. q is a value chosen randomly with uniform probability in [0,1], q_0 ($0\le q_0\le 1$) is a parameter, and S is a random variable selected according to the distribution given by Eq.(19), which gives the probability with which an ant in node r choose the node s to move to.

$$p(r,s)=\frac{[\tau(r,s)][\eta(r,s)]^{\beta}}{\sum_{u\in N(r)}[\tau(r,u)][\eta(r,u)]^{\beta}} \quad (19)$$

While building a solution, ants visit nodes and change their amount of pheromone trail by applying the following local updating rule:

$$\tau(r,s)\leftarrow(1-\rho)\tau(r,s)+\rho\Delta\tau(r,s) \quad (20)$$

where ρ ($0<\rho<1$) is the pheromone decay parameter. $\Delta\tau(r,s)=\tau_0$ is the initial pheromone level.

After all ants had completed their tours, global updating is performed. The pheromone amount is

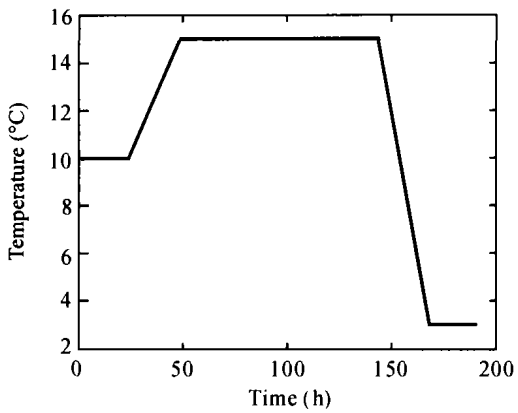

Fig.1 Industrial temperature profile

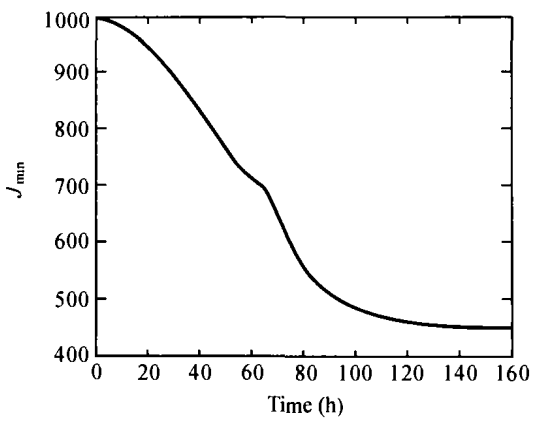

Fig.2 Objective function value along the process

updated by applying the follow global updating rule. To simplify the problem, global updating rule only apply to those tours that once belonged to the global-best-tour:

$$\tau(r, s)\leftarrow(1-\alpha)\tau(r, s)+\alpha\cdot(J_{\min gb})^{-1} \quad \text{if } (r, s)\in \textit{global-best-tour} \qquad (21)$$

$0<\alpha<1$ is the pheromone decay parameter, and $J_{\min gb}$ is the $J_{\min}$ value of the global best temperature profile from the beginning of the trail.

A version of ant colony system algorithm for this case is rapidly developed by MATLAB language.

Initial tests were done with recommended parameter settings (Dorigo and Gambardella, 1997): m=10; β=2; q_0=0.9; $\alpha=\rho$=0.1; *CONS*=1000; $\tau_0=(161\times(1000-541.5504))^{-1}$; iteration number=5. After several runs, the parameter β changes to 4; q_0 changes to 0; iteration number changes to 10. With these parameters, a promising temperature profile has been achieved.

RESULTS AND DISCUSSION

The temperature profile obtained using ACS algorithm for the selected parameters in this paper is shown in Fig.3. This jagged profile calls for a smoothing process. This process can be done by means of average calculation for every 40 hours of the obtained temperature profile. Four new temperature values are obtained and placed in the center point of every time range (40 hours). From 0 to 20 hours (expressed as [0,20]), the temperature (T) is the first mean value (expressed as *mean*1); [20,40], T changes from *mean*1 to *mean*2 (straight line links this two values); [40,60], T=*mean*2; [60,80], T changes from *mean*2 to *mean*2+(*mean*3−*mean*2)/2; [80,100], T changes from *mean*2+(*mean*3−*mean*2)/2 to *mean*3; [100,120], T=*mean*3; [120,140], T changes from *mean*3 to *mean*4; [140,160], T=*mean*4. Fig.4 shows the final temperature profile, which is suitable for implementation, obtained after smoothing process.

Using this smoothed temperature profile, the behavior of the kinetic variables of the fermentation process model can be shown in Figs.5, 6, 7 and 8. Fig.5 illustrates time courses of the total suspended biomass and its components: active, latent, and dead cells. Fig.6 is the sugar concentration curve, Fig.7 is the ethanol concentration curve and Fig.8 illustrates diacetyl concentration.

Several runs have been made changing the initial parameters related to the ACS algorithm. It is not recommended to use a large number of iterations to get "better results", because of the following reasons: (1) After 20 iterations, $J_{\min}$ seems to be constant in the experiment; (2) More iterations means much more time to get satisfactory result; (3) Most importantly, the trend of the industrial temperature profile together with Increase Once Rule will keep some results out of feasible solutions. So it is better to have more runs with moderate iterations in each run than the contrary approach. Actually, in these runs $J_{\min}$ are always satisfactory, but

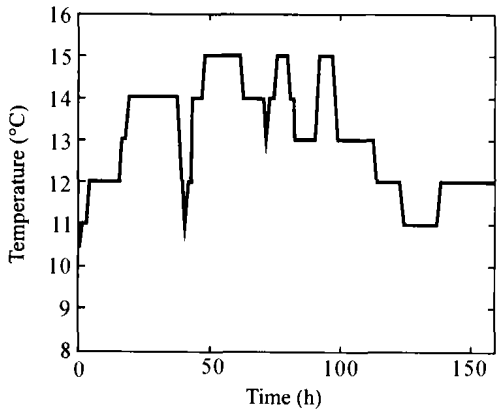

Fig.3 Optimal temperature profile obtained by ACS

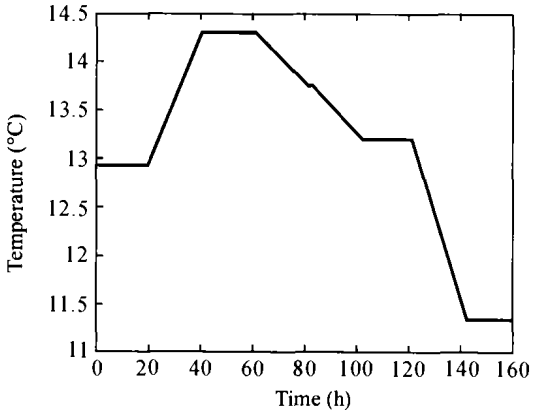

Fig.4 Smoothed temperature profile

sometimes results are infeasible solutions. After reviewing all the results, the best temperature profile with minimum J_{min} was selected and smoothed for feasible implementation. This new temperature profile is obtained not only by minimizing the value of J_{min}, which ensures getting the required ethanol level without sacrificing the quality of beer, but also by taking application factors into consideration.

Table 1 shows the results and comparison with previous results found in the literature.

CONCLUSION

In this paper, with the help of MATLAB language, ACS algorithm is first applied to the beer fermentation process optimization problem. The promising results of this work show that the ACS approach can compete with the original GA approach (Andres-Toro *et al.*, 1997a; 1997b; Carrillo-Ureta *et al.*, 2001) in terms of solution quality and computation speed in this problem and indicate the potential of ACS as a reliable and useful tool for solving dynamic process optimization problems.

Further work to be done includes: using ACS algorithm in continuous spaces without discretization approximation; getting a smooth profile without an additional smoothing process; and considering other industrial implementation aspects: economic factors, energy management, etc.

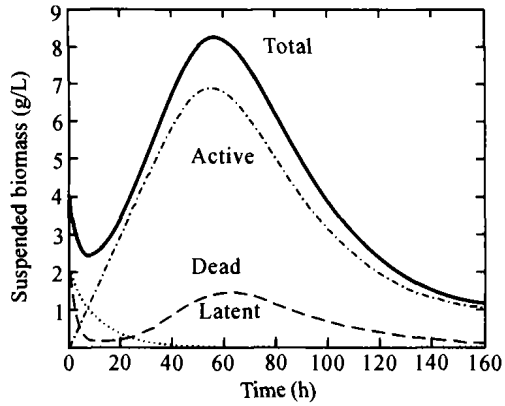

Fig.5 Suspended biomass behavior

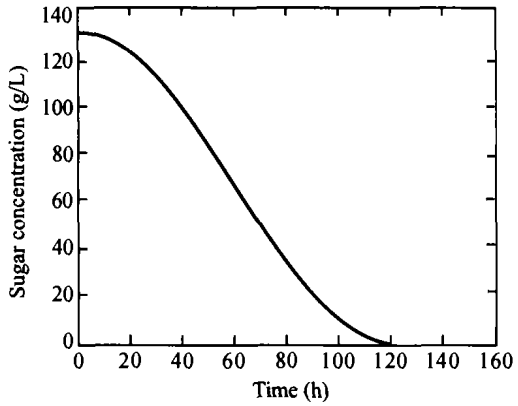

Fig.6 Time course of sugar concentration

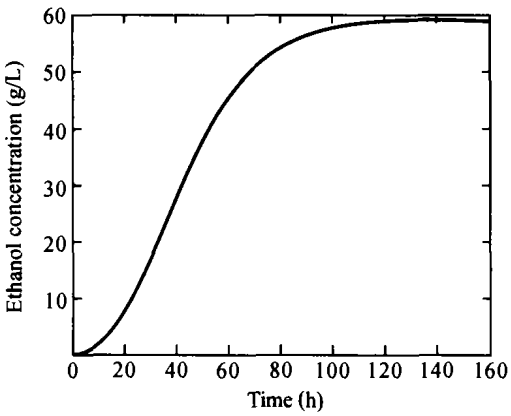

Fig.7 Time course of ethanol concentration

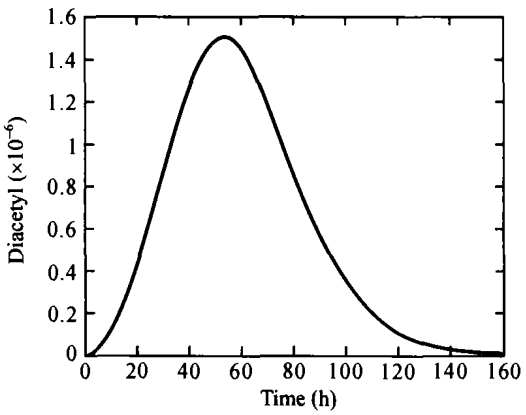

Fig.8 Time course of diacetyl concentration

Table 1 Comparison table

	Industrial temperature profile	Temperature profile obtained by GA	Temperature profile obtained by ACS
J_{max}	546.3	585.5214	585.5865
Computation effort (s)		NA	30.87

References

Andres-Toro, B., Giron-Sierra, J.M., Lopez-Orozco, J.A., Fernandez-Conde, C., 1997a. Application of Genetic Algorithms and Simulations for the Optimization of Batch Fermentation Control. Proc. IEEE International Conference on System, Man, and Cybernetics, 'Computational Cybernetics and Simulation', p.392-397.

Andres-Toro, B., Giron-Sierra, J.M., Lopez-Orozco, J.A., Fernandez-Conde, C., 1997b. Using Genetic Algorithms for Dynamic Optimization: An Industrial Fermentation Case. Proceedings of the 36th Conference on Decision & Control. San Diego, California, USA,p.828-829.

Andres-Toro, B., Giron-Sierra, J.M., Lopez-Orozco, J.A., Fernandez-Conde, C., Peinado, J.M., Garcia-Ochoa, F., 1998. A kinetic model for beer production under industrial operational conditions. *Mathematics and Computers in Simulation*, **48**:65-74.

Carrillo-Ureta, G.E., Roberts, P.D., Becerra, V.M., 2001. Genetic Algorithms for Optimal Control of Beer Fermentation. Proceedings of the 2001 IEEE International Symposium on Intelligent Control, Mexico City, Mexico, p.391-396.

Dorigo, M., Gambardella, L.M., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, **1**(1):53-66.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, **26**(1):29-41.

Dorigo, M., Gambardella, L.M., Middendorf, M., Stutzle, T., 2002. Guest editorial special section on ant colony optimization. *IEEE Transactions on Evolutionary Computation*, **6**(4):317-319.

Appendix A

Table 2 Nomenclature used

Parameter	Description	Unit	Parameter	Description	Unit
X_{act}	Suspended active biomass	g/L	μ_{lag}	Specific rate of activation	h^{-1}
X_{hg}	Suspended lag biomass	g/L	μ_x	Specific yeast growth rate	h^{-1}
X_{sd}	Suspended dead biomass	g/L	μ_{dt}	Inhibition rate of yeast growth	h^{-1}
$X_{inc}(0)$	Inoculum value: 4	g/L	μ_s	Substrate consumption rate	h^{-1}
$C_e(t)$	Ethanol concentration	g/L	k_s	Sugar inhibition parameter	g/L
$C_s(t)$	Sugar concentration	g/L	f	Fermentation inhibitor factor	
$C_{ea}(t)$	Ethyl acetate concentration	10^{-6}	μ_e	Ethanol production rate	h^{-1}
$C_{dy}(t)$	Diacetyl concentration	10^{-6}	Y_{eas}	Ethyl acetate coefficient rate	
$C_{s0}(t)$	Initial sugar concentration value: 130	g/L	μ_{dy}	Diacetyl appearance rate	
t	Time	H	μ_{ab}	Diacetyl reduction rate	
T	Temperature	°C	μ_{sd}	Specific yeast settling down rate	h^{-1}

Appendix B

ACS algorithm:

Loop /* at this level each loop is called an iteration*/
  Each ant is positioned on a starting node
    Loop /* at this level each loop is called a step*/
      Each ant applies *a state transition rule* to incrementally build a solution
        and *a local pheromone updating rule*
    Until all ants have built a complete solution
    A global pheromone updating rule is applied
Until End_condition