Simulation and Optimization of Citric Acid Production with SuperPro Designer using a Client-Server Interface

ALEXANDRU WOINAROSCHY*, SANDU TARAS
Politehnica University of Bucharest, Dept. of Chemical Engineering, 1-5 Poliz Str., 011061, Bucharest, Romania

SuperPro Designer is a very useful simulator due to its large data bases of specific chemical compounds and unit operations, and for the evaluation of economical and ecological process performance indexes. For sensitivity analysis and optimization purposes, hundreds or more simulations are necessary. Consequently, an off-line use of the simulator isn't possible. A client-server interface based on Component Object Module (COM) technology using Excel-Visual Basic for Applications (VBA) scripts was developed in order to call SuperPro Designer simulator repetitively, inside loops, for various sets of input variables. The potential of the client-server application was demonstrated in the frame of the sensitivity analysis and optimization of the bioprocesses of citric acid production from starch fermentation using Aspergillus niger. The corresponding results are useful both for technological, and economical reasons.

Keywords: biochemical processes, simulation, optimization

Modeling, simulation, and optimization play today a crucial role in achieving a full exploitation of the bioprocesses potential. SuperPro Designer is the most useful and used simulator [1] for technological processes of fine chemical and biochemical products. The main advantages of this simulator consist in a large data base of specific chemical compounds and unit operations, and the evaluation of economical and ecological process performance indexes.

For the sensitivity analysis purposes a simulator can be used with a little bit trouble in the usual, classical mode, by manual assigning of each set of the input variables. If the number of input data sets is large, this procedure is clumsy.

Most of all chemical and biochemical engineering simulators are not equipped with optimization tools. In very few simulators (e.g. Aspen Plus) there are some optimization tools, but the formulation of optimization problems and techniques are severely imposed. SuperPro Designer doesn't contain any optimization tools. For optimization purpose, hundreds or more evaluations of the objective function, and consequently the same number of complete simulations, are necessary. Therefore, for optimization applications, an off-line, classical use of the simulator isn't possible. Due to this fact, the application of chemical and biochemical engineering simulators were not involved up today in the search of the optimum solutions. In the present work a useful client-server application was developed in order to call SuperPro Designer simulator repetitively, inside loops, for various sets of input variables.

Sensitivity nalysis

For the first step, respectively for the sensitivity analysis purposes, a client-server interface based on Component Object Module (COM) technology was realized. This is a new feature that has become available since the release of version 6.0 of SuperPro Designer [2]. Using COM technology, it is possible to add code so that the applications behave as an Object Linking and Embedding (OLE) Automation Server. Using the SuperPro Designer COM Server it can explode Windows interoperability combining several other applications with SuperPro Designer. The SuperPro Designer COM Server can thus be used by client applications to perform various tasks including: data exchange between the SuperPro Designer simulation variables (input/output) and other applications, exporting of SuperPro Designer reports, charts, and pictures to specified files or to the clipboards, etc. The object exposed by the SuperPro Designer server is the Designer Type Library. The use of the methods of this library to interoperate with other Windows applications (such as Excel) requires the use of a common scripting language, and Visual Basic for Applications (VBA) was chosen to this.

An interface between Excel and SuperPro Designer, based on COM technology, using Excel-VBA scripts [3] was realized. This interface was used for a study of sensitivity analysis applied to the bioprocesses of citric acid production from starch fermentation using Aspergillus niger (fig.1 [4]).

For the variation of the starch flowrate between 28000 and 31000 kg/batch with a step of 100 kg/batch (input stream S-109) there were investigated the evolutions of the values for: effluent biomass concentration in stream S-118, CO_2 effluent flowrate in stream S-117, and from the Economic Evaluation Report the citric acid productivity, unit production cost, total capital investment, and total annual revenues. The corresponding VBA code contains the following functions: DocumentObjects (define and open the connection between Excel and SuperPro Designer), SetAndReadStarchFlowrate, ReadBiomassConcentration, ReadCO2 Flowrate, ReadCitricAcidProductivity, ReadUPC, ReadCapitalInvestment, Read AnnualRevenues, and a code for the button press event. The Excel interface sheet for starch flowrate analysis is presented in figure 2.

For code exemplification are given the contents and some comments of the functions DocumentObjects and SetAndReadStarchFlowrate:

```
Function DocumentObject () As Object
    Dim spfFileOne As String
    spfFileOne = Range ("B2")
    Set DocumentObject = GetObject (spfFileOne)
End Function
```

email: a_woinaroschy@chim.upb.ro

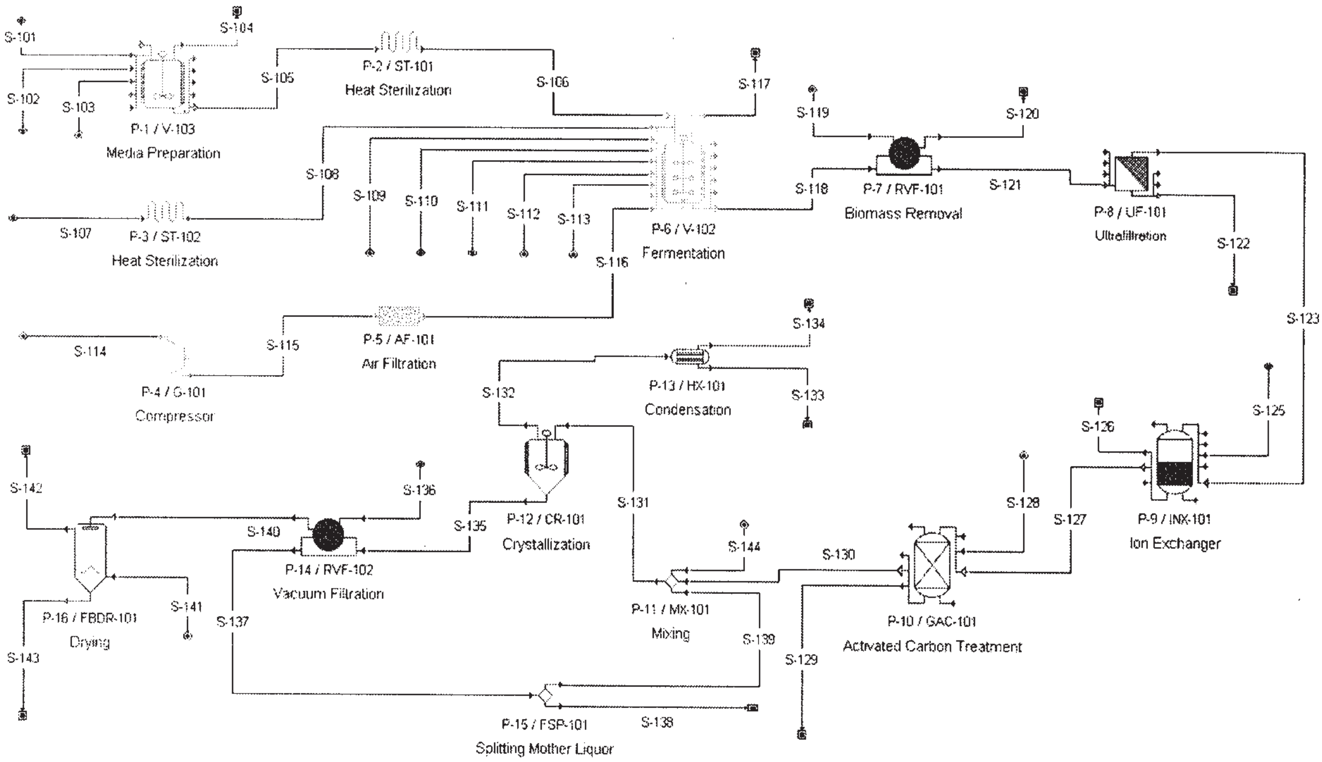
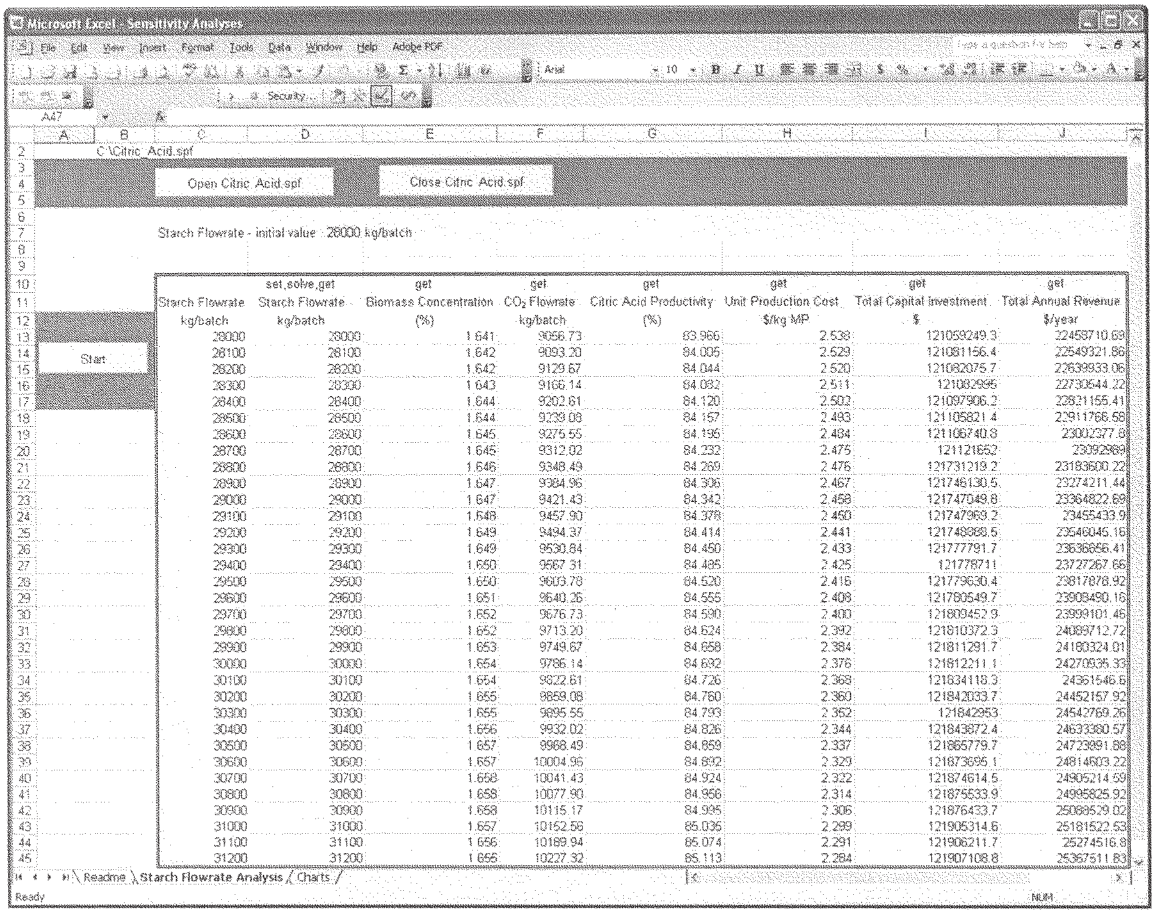


Fig. 1. The citric acid process flow diagram [4]



	set,solve,get	get	get	get	get	get	get
Starch Flowrate kg/batch	Starch Flowrate kg/batch	Biomass Concentration (%)	CO_2 Flowrate kg/batch	Citric Acid Productivity (%)	Unit Production Cost $/kg MP	Total Capital Investment $	Total Annual Revenue $/year
28000	28000	1.641	9056.73	83.966	2.538	121059249.3	22458710.69
28100	28100	1.642	9093.20	84.005	2.529	121081156.4	22549321.86
28200	28200	1.642	9129.67	84.044	2.520	121082075.7	22639933.06
28300	28300	1.643	9166.14	84.082	2.511	121082995	22730544.22
28400	28400	1.644	9202.61	84.120	2.502	121097906.2	22821155.41
28500	28500	1.644	9239.08	84.157	2.493	121105821.4	22911766.58
28600	28600	1.645	9275.55	84.195	2.484	121106740.8	23002377.8
28700	28700	1.645	9312.02	84.232	2.475	121121652	23092989
28800	28800	1.646	9348.49	84.269	2.476	121731219.2	23183600.22
28900	28900	1.647	9384.96	84.306	2.467	121746130.5	23274211.44
29000	29000	1.647	9421.43	84.342	2.458	121747049.8	23364822.69
29100	29100	1.648	9457.90	84.378	2.450	121747969.2	23455433.9
29200	29200	1.649	9494.37	84.414	2.441	121748888.5	23546045.16
29300	29300	1.649	9530.84	84.450	2.433	121777791.7	23636656.41
29400	29400	1.650	9567.31	84.485	2.425	121778711	23727267.66
29500	29500	1.650	9603.78	84.520	2.416	121779630.4	23817878.92
29600	29600	1.651	9640.26	84.555	2.408	121780549.7	23908490.16
29700	29700	1.652	9676.73	84.590	2.400	121809452.9	23999101.46
29800	29800	1.652	9713.20	84.624	2.392	121810372.3	24089712.72
29900	29900	1.653	9749.67	84.658	2.384	121811291.7	24180324.01
30000	30000	1.654	9786.14	84.692	2.376	121812211.1	24270935.33
30100	30100	1.654	9822.61	84.726	2.368	121834118.3	24361546.6
30200	30200	1.655	9859.08	84.760	2.360	121842033.7	24452157.92
30300	30300	1.655	9895.55	84.793	2.352	121842953	24542769.26
30400	30400	1.656	9932.02	84.826	2.344	121843872.4	24633380.57
30500	30500	1.657	9968.49	84.859	2.337	121865779.7	24723991.88
30600	30600	1.657	10004.96	84.892	2.329	121873695.1	24814603.22
30700	30700	1.658	10041.43	84.924	2.322	121874614.5	24905214.59
30800	30800	1.658	10077.90	84.956	2.314	121875533.9	24995825.92
30900	30900	1.658	10115.17	84.995	2.306	121876433.7	25089529.02
31000	31000	1.657	10152.56	85.036	2.299	121905314.6	25181522.53
31100	31100	1.656	10189.94	85.074	2.291	121906211.7	25274516.8
31200	31200	1.656	10227.32	85.113	2.284	121907108.8	25367511.83

Fig. 2. The Excel interface sheet for starch flowrate analysis

```
Function SetAdReadStarchFlowrate (StarchFlowrate) As Double
    Dim str As String
    Dim var1, var2  As Variant
    Set DocObj1  =  DocumentObject ()
    var1  =  CDb1 (StarchFlowrate)
    str  =  "S-109"
    DocObj1.SetStreamVarVal str, VarID.massFlow_VID, var1, str
    DocObj1.DoMEBalances var1
    DocObj1.DoEconomicCalculations
    DocObj1.GetStreamVarVal str, VarID.massFlow_VID, var2, str
    SetAdReadStarchFlowrate  =  CDb1 (var2)
    End Function
```

In the DocumentObjects function it is indicated the address of the corresponding SuperPro Designer file (with extension spf) at the Excel cell B2. In the SetAndReadStarchFlowrate function, the fourth line makes the link with DocumentObject function. There are declared the stream S-109, and the set and get variables var1 and var2, corresponding to the starch flowrate in the stream S-109. The 8th and 9th lines command the execution of mass balances and economic calculations.

Optimization

An optimization procedure can be realized inside Excel, using its proper programming language or by the use of Excel Solvers. These ways are difficult and not certain. We consider to be more attractive the use for optimization the facilities offered by Matlab: the optimization procedures given by the functions from Matlab Optimization toolbox, many free optimization Matlab programs available on Internet (e.g. Gaot for genetic algorithms, Aneal for simulated annealing, etc), and the user optimization procedures written in Matlab language. For optimization purpose, it can be used Matlab as an Automation client and Excel as the server, which is a facility existing in Matlab COM technology [5].

The optimization application consists in simultaneous minimization of unit production cost UPC and total capital investment TCI for previous bioprocess, respectively citric acid production from starch fermentation using *Aspergillus niger*. The independent variables were: x_1 = starch flowrate in input stream S-109 and x_2 = fermentation time in fermentor V102. The two performance indexes were combined in a weighted objective function:

$$\mathrm{FOB} = \mathrm{UPC} + w \cdot \mathrm{TCI}$$

where w is a scale and weighting coefficient. This can be used to modify the weight, and consequently the interest in each of the two performance indexes.

The Matlab optimization procedure is based on a genetic algorithm [6], free distributed on Internet [7]. Float representation of chromosomes has been used. The selection of candidates chromosomes for crossover and mutation is made according with a ranking selection function based on the normalized geometric distribution. Three types of crossover are applied: simple, interpolated, and extrapolated crossover. In the simple crossover, the crossover point is randomly selected. The interpolated crossover performs an interpolation along the line formed by the two parents. The extrapolated crossover performs an extrapolation along the line formed by the two parents in the direction of better parent. Four types of mutation are applied: boundary, multi-nonuniform, nonuniform, and uniform mutation. Boundary mutation changes one gene of the selected chromosome randomly either to its upper or lower bound. Multi-nonuniform mutation changes all genes, whereas nonuniform mutation changes one of the genes in a chromosome on the base of a non-uniform probability distribution. This Gaussian distribution starts wide, and narrows to a point distribution as the current generation approaches to the maximum generation. Uniform mutation changes one of the genes based on a uniform probability distribution. The numbers of applications of the different crossover and mutation operators are imposed as parameters of the genetic algorithm. Their default values have been used, respectively for each generation: 2 simple, 2 interpolated, and 2 extrapolated crossover, 4 boundary, 6 multi-nonuniform, 4 nonuniform, and 4 uniform mutation. Due to the use of a maximization algorithm, the chromosome fitness corresponds to the negative value of the objective function.

During the evolution of the genetic algorithm, the values of independent variables x_1 and x_2 are transmitted to Excel using Matlab COM technology. In Excel these values are attributed to the corresponding process variables and sent to SuperPro Designer by a similar interface with that described in Sensitivity Analysis section. After simulation, SuperPro designer returns to Excel the corresponding values of UPC and TCI, and in Excel is calculated the value of FOB. Finally, Excel returns this value to Matlab for the genetic algorithm procedure. This variables flow is presented in figure 3.

The client server optimization application contains GAOT toolbox, the Excel interface with SuperPro Designer (file Model.xls), the SuperPro Designer file with simulation of citric acid production (file Citric_Acid.spf) and the following Matlab functions: main.m, fob.m, open_spf.m, and close_spf.m. These functions realize the following actions:

- function main.m: calls open_spf.m for opening the spf file; initializes bounds on independent variables, number of chromosomes of initial populations, and total number of generations; generates the initial population by calling the function initializega.m from GAOT toolbox; calls the genetic algorithm procedure, respectively function ga.m from GAOT toolbox; displays the optimal solution, computer execution time, and graphic representation of genetic algorithm evolution; calls close_spf.m for closing the spf file;
- function fob.m: transfers to Excel interface, respectively Model.xls, the values of independent variables x_1 and x_2 and receives the corresponding value of the objective function;
- functions open_spf.m, and close_spf.m: open and, respectively, close the spf file with bioprocess simulation (Citric_Acid.spf).

Here are presented the functions open_spf.m and fob.m:

```
function fval = open_spf(x)
excelapp = actxserver('Excel.Application');
wkbk = excelapp.Workbooks;
wdata = wkbk.Open('c:\MAEXSP\Model.xls');
sheets = wdata.Sheets;
sheet12 = sheets.Item('Sheet1');
if (x==1)
range = sheet12.set('Range', 'F4', x);
fval = 1;
end
wdata.Saved = 1;
wdata.Close;
excelapp.delete;
```

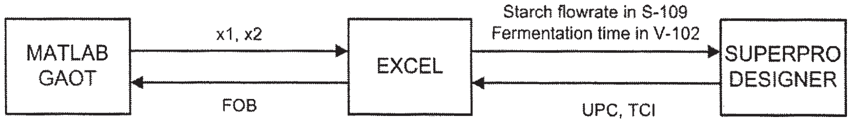


Fig. 3. Variables flow in the optimization application

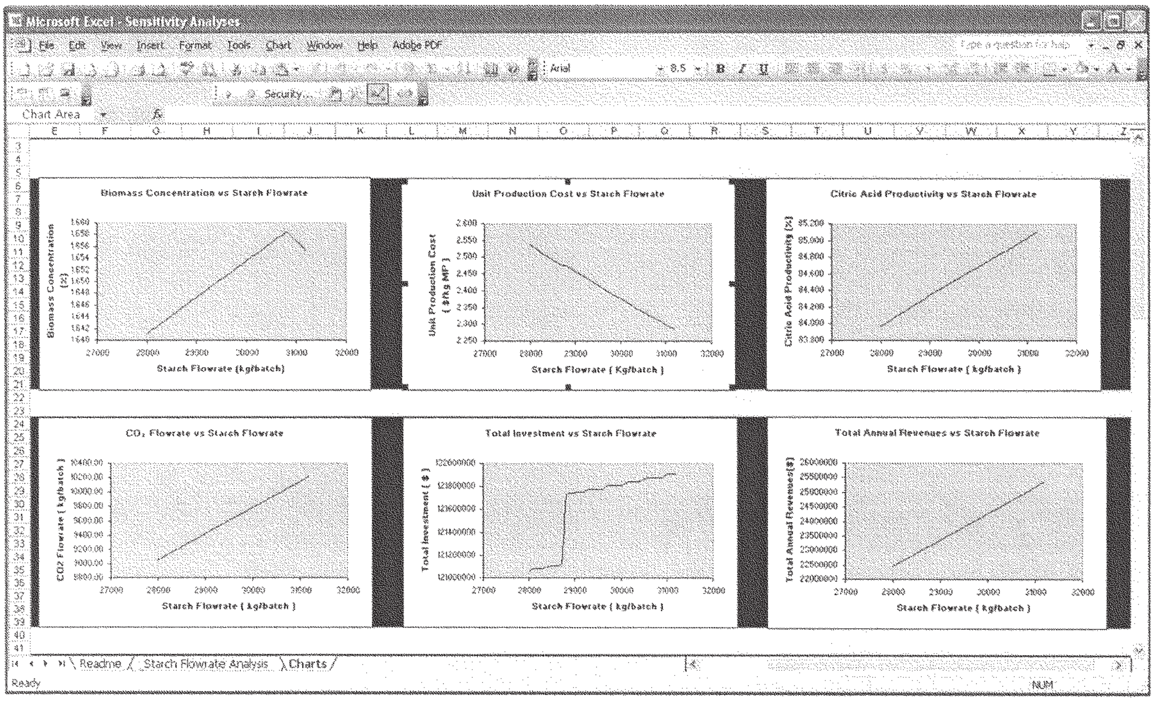


Fig. 4. Results of the sensitivity analysis corresponding to the variation of the starch flowrate

```
function [sol, val] = fob(sol,options)
excelapp = actxserver('Excel.Application');
wkbk = excelapp.Workbooks;
wdata = wkbk.Open('c:\MAEXSP\Model.xls');
sheets = wdata.Sheets;
sheet12 = sheets.Item('Sheet1');
range = sheet12.set('Range', 'B2', sol(1));
range = sheet12.set('Range', 'C2', sol(2));
range = sheet12.get('range', 'D2');
val = -range.value;
wdata.Saved = 1;
wdata.Close;
excelapp.delete;
```

It can be observed the use of Automation client-server from Matlab COM technology.

Results and discussions

The results of the sensitivity analysis corresponding to the sheet Charts of the interface are presented in figure 4.

It can be observed for starch (row material) flowrate increasing, an expected linear increasing of citric acid productivity, of CO_2 effluent flowrate, and of total annual revenues. The corresponding increasing of total capital investment presents a discontinuity due to equipments size standardization. The unit production cost decreases with starch flowrate increasing. An interesting evolution presents biomass concentration: increasing of biomass concentration with starch flowrate increasing is followed by a decreasing, which can be explained by biomass consumption by starch supplementary amounts.

In the optimization application, using an initial population of 20 chromosomes, after 50 generations the best solution is: x_1 = 28661 kg/batch, and x_2 = 120 h, corresponding to UPC = 2.25 \$/kg and TCI = 121.174 mil. \$. This solution corresponds to the value of the weighting coefficient $w = 4.1322 \cdot 10^{-7}$.

On an IBM Intel Pentium IV computer (2.66 GHz processor, 1 GB memory, Microsoft Windows XP Professional SP2 operating system, Matlab 6.5, Excel 2003, SuperPro Designer 7.0) the execution time was 6.73 h, involving 1220 evaluations of the objective function, respectively SuperPro Designer simulations. Almost total time needed for process simulations is used for the convergence of material balances corresponding to recycled loop S-135, S-137, S-139, and S-131. The execution time can be substantially reduced both by programming improvements, and by the use of a more powerful computer.

The evolution of the genetic algorithm, respectively the mean value of the fitness at each generation, is presented in figure 5.

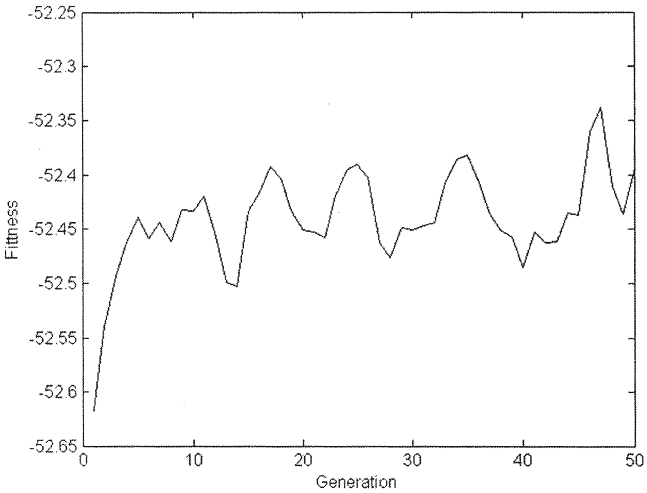


Fig. 5. The evolution of the genetic algorithm

If the value of the weighting coefficient w is ten times higher, the total capital investment participation in the objective function is correspondingly increased. In this case the best solution obtained is: x_1 = 25000 kg/batch, and x_2 = 120 h, corresponding to UPC = 2.59 \$/kg and TCI = 120.885 mil. \$. It can be remarked the expected UPC increasing, and TCI decreasing in comparison with the previous best solution.

The unit operation models in SuperPro Designer are not very complex. If more accurate results are desired, the solutions obtained with SuperPro Designer at the large scale of the entire process can be refined at the small scale of unit operations. For the investigated unit operations, detailed and more sophisticated simulation models can be developed.

Conclusions

In the present work it was developed a useful client-server application in order to call the SuperPro Designer simulator repetitively, inside loops, for various sets of input variables. The application was realized by VBA scripts inside EXCEL. For optimization purposes, this was linked with Matlab using Automation client-server of Matlab COM technology. Unfortunately, not all the process variables are given in the Designer Type Library of SuperPro Designer server. Consequently, the selection of the independent variables is slightly restricted. Up today, the use of SuperPro Designer simulator for optimization problems on the base of a client-server application was not reported in literature.

The potential of the client-server application was demonstrated in the frame of the sensitivity analysis and optimization of the bioprocesses of citric acid production from starch fermentation using *Aspergillus niger*. The corresponding results are useful both for technological, and economical reasons.

Acknowledgments: Financial support from National University Research Council, Ministry of Education and Research, Grant ID 1727, Multilevel Optimization of Sustainable Bioprocesses is gratefully acknowledged.

References

1.*** INTELLIGEN, INC., SuperPro Designer User's Guide, 2007
2.*** INTELLIGEN, INC., SuperPro Component Object Module (COM) Help, 2007
3.BIRNBAUM D., Excel VBA Programming for the Absolute Beginner, Ed. 2, Thomson Course Technology PTR, Boston, 2005
4.HEINZLE E., BIWER A.P., COONEY C.L., Development of Sustainable Bioprocesses: Modelling and Assessment, Wiley & Sons, New York, 2006
5.*** MATHWORKS, INC., Matlab Release 13 Help, 2002
6.HAUPT R.L., HAUPT S.E., Practical Genetic Algorithms, Wiley & Sons, New York, 1998
7.HOUCK C., JOINES J., KAY M., The Genetic Algorithm Optimization Toolbox (GAOT) for Matlab 5, http://www.ise.ncsu.edu/mirage/GAToolBox/gaot/

Manuscript received: 16.05.2009