



LUND  
UNIVERSITY

# Differential Evolution and Chewie, a genetic optimization algorithm and computation cluster

**Mark Max-Hansen**

Department of Chemical Engineering, Lund University, P.O. Box 124, SE-221 00 Lund, SWEDEN  
www.chemeng.lth.se

## Introduction

Modeling of physical systems such as Chromatography has been possible for some time now, thanks to more powerful computers, however optimization is in many cases difficult as the models in chromatography for an example are very stiff and multimodal, which lead to problems for the classic indirect methods, also known as Newton solvers. One way to work around this is to use direct methods, such as the Nelder-Mead simplex method, or Differential Evolution.

## Protein Chromatography Simulator

The Protein Chromatography Simulator was developed by Ph.D. student N. Borg at the Department of Chemical Engineering, Lund University. It simulates Ion Exchange and Reversed Phase Chromatography. The equations below are an example of a homogenous Langmuir model.

### Column description

$$\frac{\partial c}{\partial t} = D_{ax} \frac{\partial^2 c}{\partial z^2} - v_{int} \frac{\partial c}{\partial z} - \frac{1 - \epsilon_c}{\epsilon_c} \frac{6}{d_p} k_f (c - c_p)$$

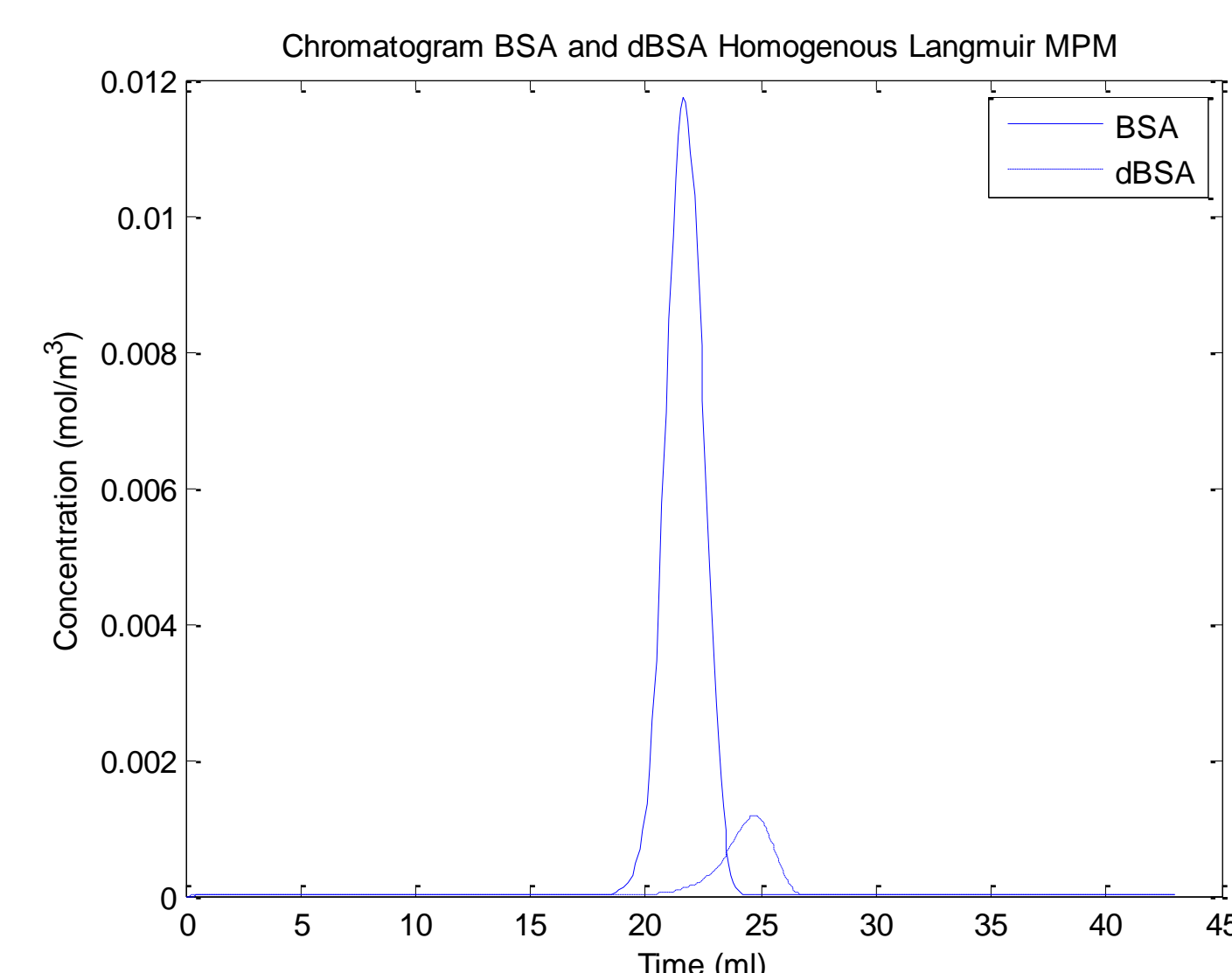
### Particle description

$$\frac{\partial c_p}{\partial t} = \frac{D_e}{\epsilon_p} \left( \frac{\partial^2 c_p}{\partial r^2} + \frac{2}{r} \frac{\partial c_p}{\partial r} \right) - \frac{1}{\epsilon_p} \frac{\partial q}{\partial t}$$

### Adsorption/desorption description

$$\frac{\partial q}{\partial t} = k_{ads} \cdot c \cdot q_{max} \left( 1 - \sum_i \frac{q_i}{q_{max,i}} \right) - k_{des} \cdot q$$

The simulator calculates concentration profiles of the entire column for each given timestep, but most often only the outlet concentration is of interest, and can be used to produce a chromatogram. The system of differential equations is very stiff, and thereby requires, depending on selected model, between 5 seconds and 10 minutes.



## Differential Evolution

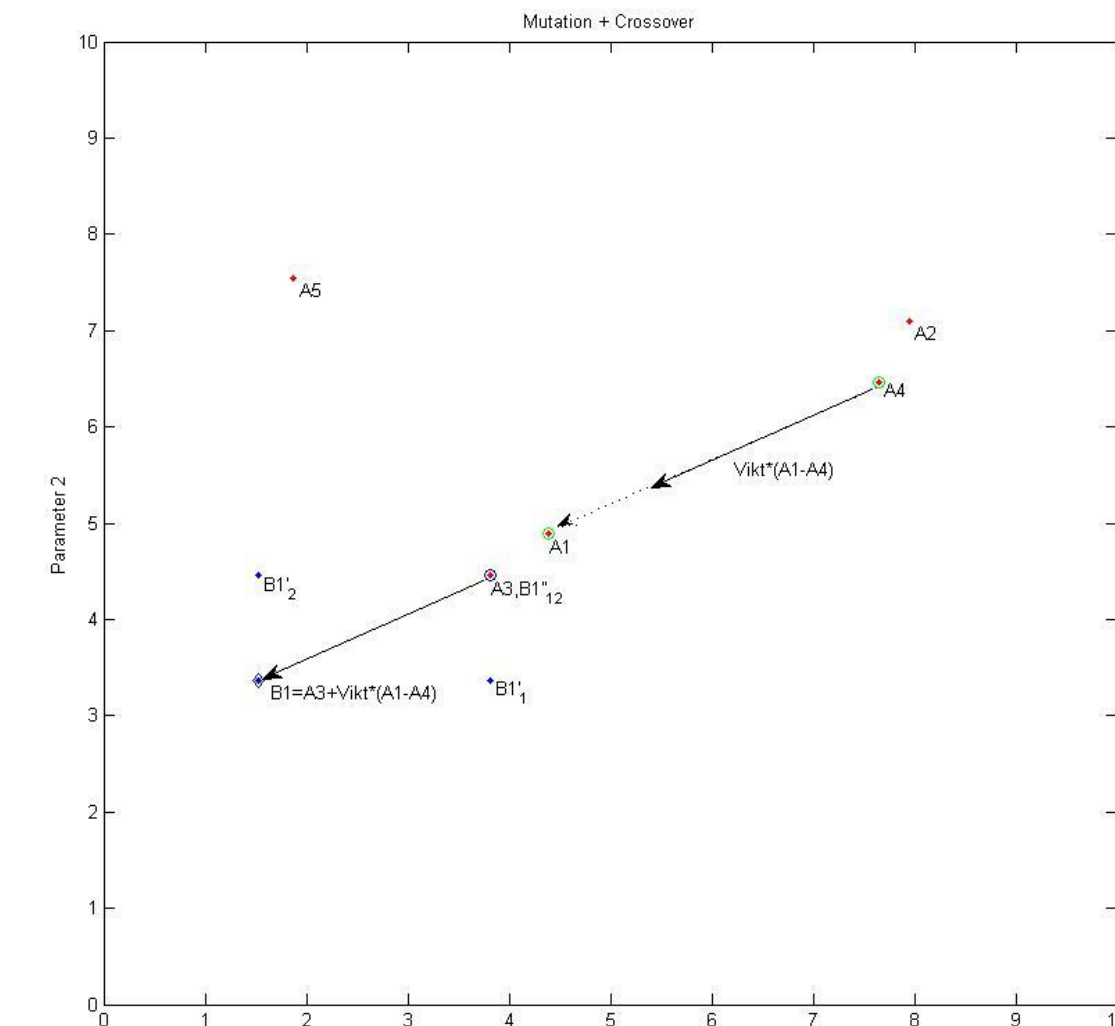
Differential Evolution is a genetic algorithm, which is a refined version of random walk that takes the values of the cost function into consideration. The process of Differential Evolution is:

The differential in differential evolution comes from vector differences, and the trial population is generated by mutation and crossover, where mutation works by adding the vector difference of two individuals in the population multiplied by a weight factor to a third member of the population:

$$\text{Individual}_{trial}(i) = \text{Individual}_{z1}(i) + \text{Weight} \cdot (\text{Individual}_{z2}(i) - \text{Individual}_{z3}(i))$$

Individual L1	Individual L2	Individual L3	Individual New
A3	A1	A4	B1=A3+W*(A1-A4)
A5	A4	A2	B2=A5+W*(A4-A2)
A1	A2	A3	B3=A1+W*(A2-A3)
A2	A5	A1	B4=A2+W*(A5-A1)
A4	A3	A5	B5=A4+W*(A3-A5)

The lists L1..L3 are randomized lists of the population. Crossover works by comparing a randomized number for each parameter with a predefined Crossover weight, CW, and if the randomized number is greater than CW the parameter is inherited from the parent individual, instead of the mutation.



The figure above shows possible outcomes for the new individual B1, where B1<sub>1</sub>, B1<sub>2</sub>, B1<sub>1,2</sub> are the points where the crossover has occurred, for one or two parameters.

Generate Initial Population

Evaluate all members

If number of generations is less than preset maximum evals and achieved cost is still bigger than Break value, then continue search.

Generate Trial Population by Mutation and Crossover

If the cost of Trial Member *i* is less than the cost of Population Member *i*, the Trial member survives, otherwise the Population member survives

The Differential Evolution algorithm can be used for optimization, or parameter estimation, depending on the setup of the costs function. It can be used to generate Pareto plots, to help with design decisions for process parameters for an example.

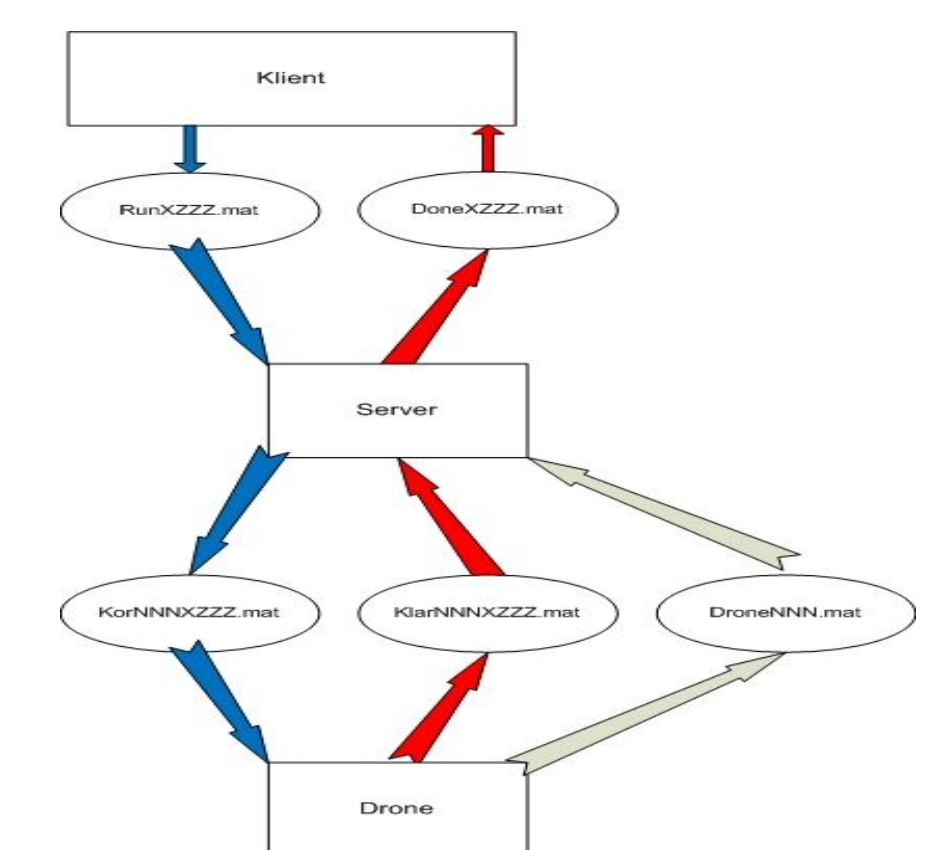
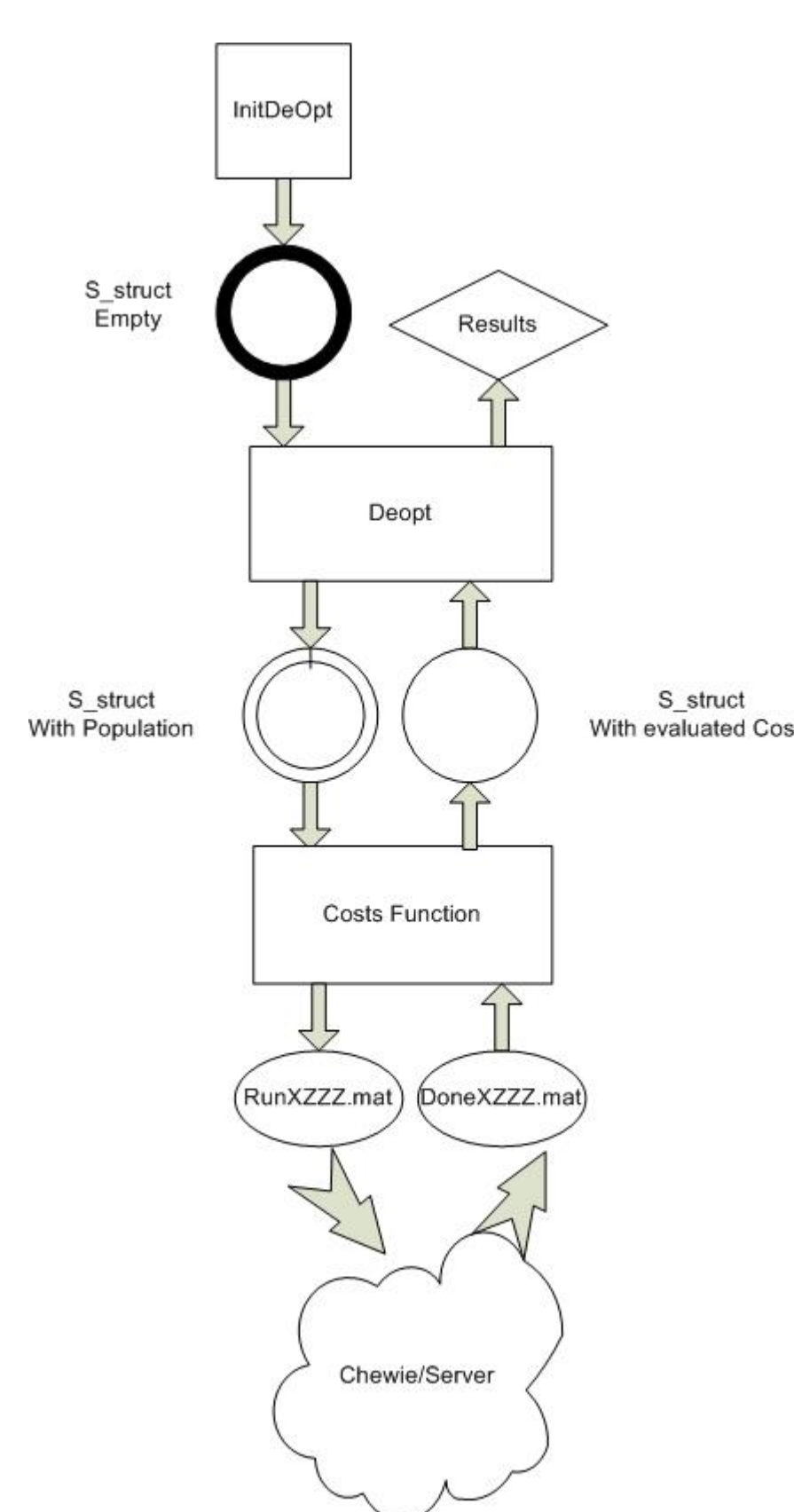
One of the advantages with Differential Evolution is that it handles only the function values, and does not try to calculate any Jacobian or Hessian matrices. Calculating Jacobian or Hessian matrices which is required with indirect solvers can be impossible with stiff ODEs. The drawback of this approach is that it requires far more function evaluations than a indirect method would. This requirement led to the development of Chewie, the computational platform that could reduce the time requirement by running the simulations in parallel.

## Chewie

Chewie is a parallelization platform, developed in Matlab, that keeps tabs on the computation drones, what users are sending what jobs, and sends back the results to the respective user.

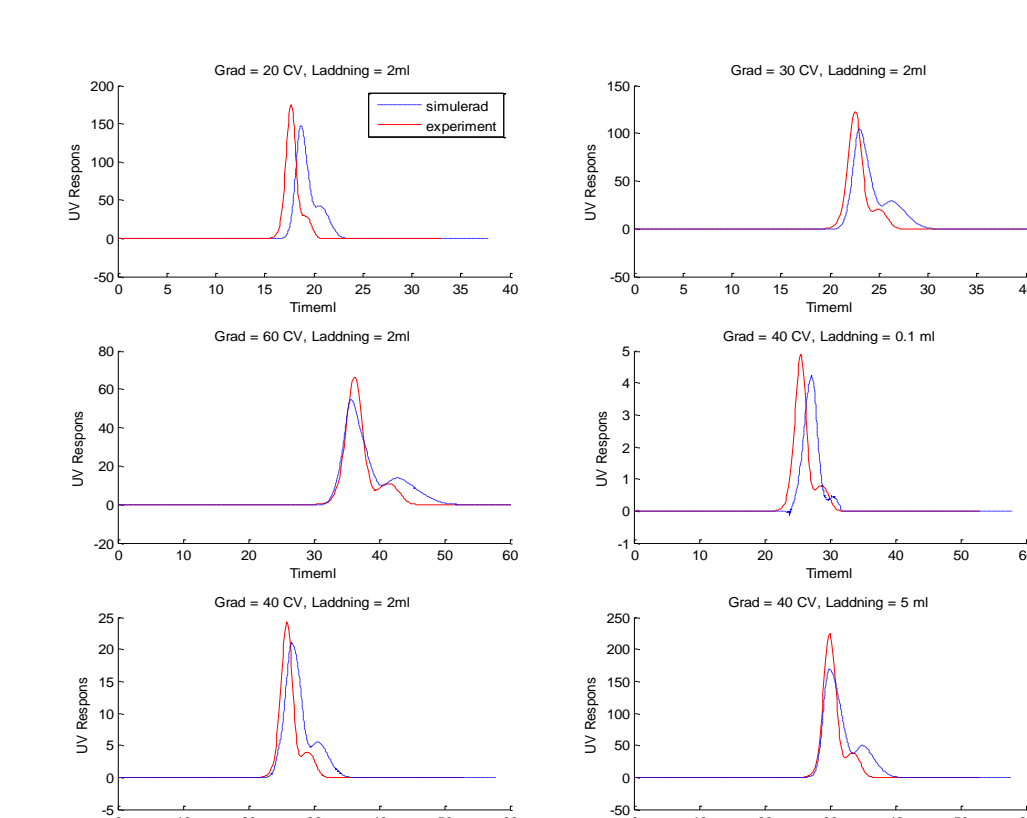
Chewie is the server, or the middleman, between the client and the drones. The client operates by sending Matlab data files named by RunXXXX.mat where X is the user number of the requester and YYY is the number of the simulation. These are renamed by the server to KorNNNXXXXY.mat, where NNN is the number of the drone to run the simulation. The drone then returns simulation results as KlarNNNXXXXY.mat, which makes Chewie mark it as available for work, and sends it back to the client as DoneXXXXY.mat.

Beneath is an example of the workflow when using the Differential Evolution algorithm with the Chewie cluster. It should be noted though, that the Chewie cluster also works well with other methods, such as Latin Hypercube Sampling.

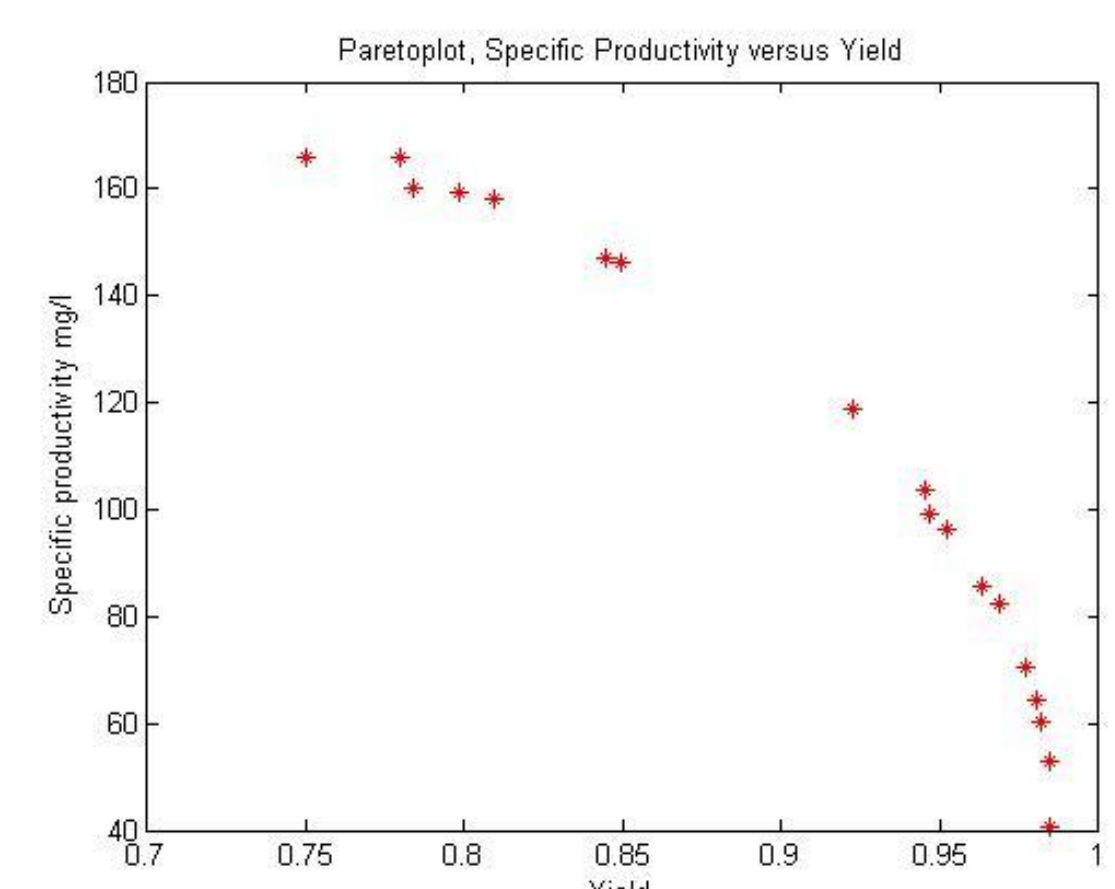


## Conclusions

Doing the parameter estimation and the Pareto plot was not possible or very difficult with the indirect methods in Matlab, and due to the nature of the numeric and physical system, a simplex method was not really applicable, and would require serial computation instead of parallel, with roughly estimated the same number of function evaluations. The Differential Evolution algorithm and the Chewie cluster has proved their worth.



Parameter estimation done with Differential Evolution and Chewie, on the BSA – BSA-dimer system.



Pareto plot of specific productivity versus yield for BSA – BSA-dimer system.