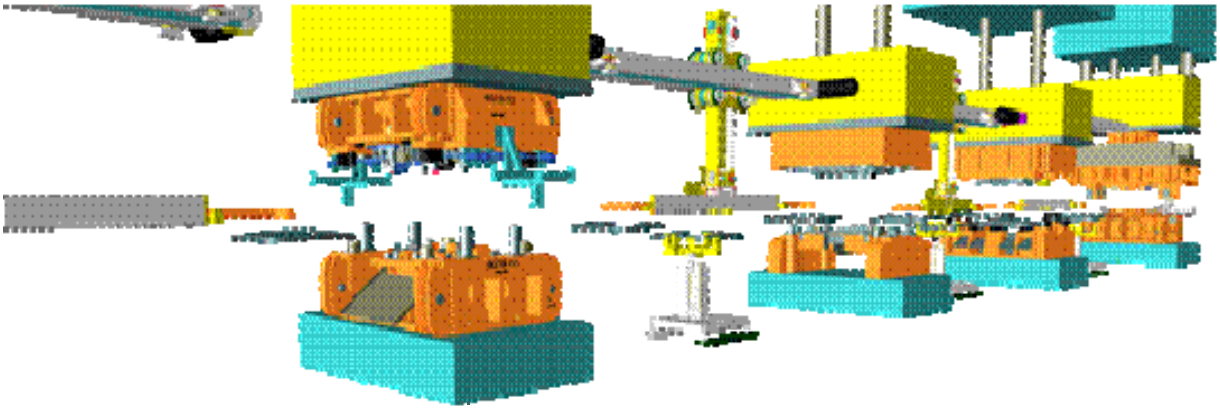
Appendix B

ENSIL – Mechatronic speciality
2nd year

report of technical studies

Study of an algorithm based on RSM



TORRES Stephane

Appendix B

Table of contents

Introduction 4
1 Algorithm parameters 5
2 Results with the simulation 6
2.1 Initialization points 6
2.1.1 100 initialization points 6
2.1.2 Conclusion for the initialization points 9
2.2 Prediction and real value 9
2.2.1 With 50 initialization points : 10
2.2.2 With 100 initialization points 11
2.3 Psi and sigma influence 13
2.3.1 Coefficient psi 13
2.3.2 Coefficient sigma 16
2.4 Correlation and regression functions 19
2.4.1 The different correlation and regression functions 19
2.4.2 Linear correlation 20
2.5 Results conclusion 23
Conclusion 24
Annex 1: 25
Annex 2: 26
Annex 3: 27
Annex 4: 31
Annex 5: 33

Appendix B

Table of pictures

Figure 1 : parameters for algorithm functions 5
Figure 2: parameter values for 100 initialization points 7
Figure 3 : algorithm behaviour for 100 initialization points 8
Figure 4: difference between prediction and real value for 50 initialization points 10
Figure 5: difference between prediction and real value for 100 initialization points 11
Figure 6: psi influence on the weight function 13
Figure 7: parameter values for psi=3 14
Figure 8: algorithm behaviour for psi=3 15
Figure 9: sigma influence on the weight function 16
Figure 10: parameter values for two sigma values added and psi=3 17
Figure 11: algorithm behaviour for two sigma value added 18
Figure 12: parameter values for linear correlation 21
Figure 13: algorithm behaviour for linear correlation 22
Figure 14: algorithm structure 25
Figure 15: algorithm behaviour for 50 initialization points 26
Figure 16: algorithm behaviour for 25 initialization points 26
Figure 17: production rate with a model using correxpg 27
Figure 18: production rate with a model using corrlin 27
Figure 19: production rate with a model using corrspherical 28
Figure 20: production rate with a model using correxp 28
Figure 21: production rate with a model using corrgauss 29
Figure 22: production rate with a model using corrspline 29
Figure 23: production rate with a model using regpoly1 30
Figure 24: production rate with a model using regpoly2 30

Appendix B

Introduction

The first part of my project was to implement an algorithm based on RSM in the Process Optimizer. Then, with the first results [ref 1], we have seen that the algorithm was not yet very efficient for a simulation based optimization problem in 10 dimensions, in particular because it ´s very difficult to build a surface in 10 dimensions with only few points. One of the goals for the second part of my project in Sweden is to test different parameters in the algorithm and see their effect on the algorithm behaviour.

Firstly, I will describe some parameters that we can tune in the algorithm to try to improve it. Then, I will show you all the results got with the algorithm and different values for all the algorithm parameters.

Appendix B

1 Algorithm parameters

The algorithm based on RSM (see Annex 1) used to optimize the objective function has several parameters which can be tuned to improve the algorithm behaviour for our simulation based optimization problem. All these parameters are written in all the result files in order to see which parameter change has done a modification in the algorithm behaviour.

Some of these parameters are (figure 1):

- number of initialization point
- type of regression and correlation function
- sigma list (tuning the balance between local and global search for the newpoint function)
- psi (weight function parameter, also tuning the balance between local and global search for the newpoint function)

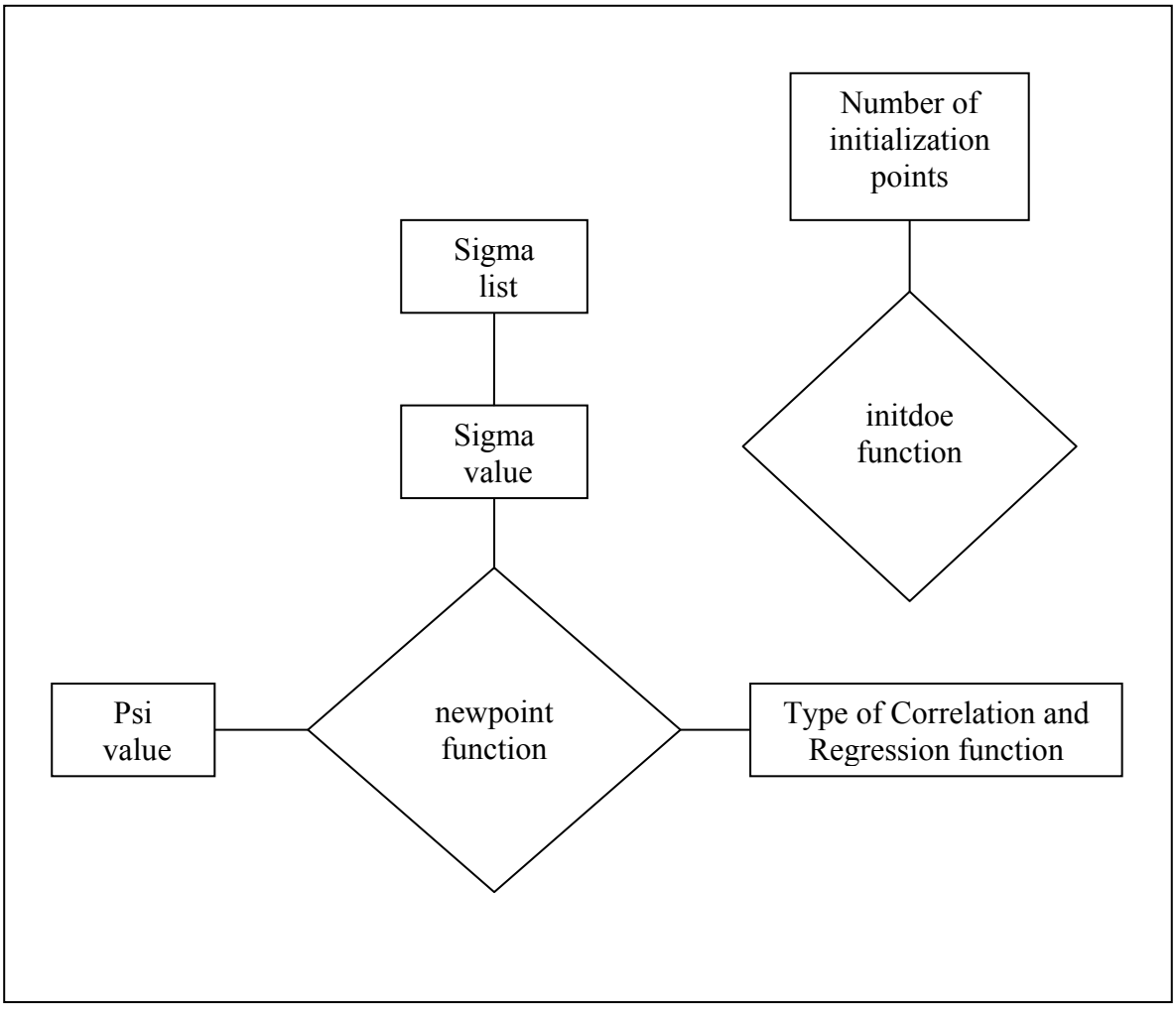


Figure 1 : parameters for algorithm functions

Appendix B

In figure 1, we can see some of the algorithm parameters associated with their algorithm function that we can tune to try to improve the algorithm behaviour. Tests for different number of initialization points are given in [ref 1] and will be kept on in this one.

The MATLAB package used in the algorithm give the possibility to test different regression and correlation functions [ref 2], these correlation and regression functions can have a real effect on the model construction. There is also the possibility to tune some special value as the psi value and the sigma list values.

The experiment in [ref 1] and in all the continuation of this report have been done with the basic parameter :

- Regression function: regpoly0
- Correlation function: correxpg
- Sigmalist : 3*10 ^ [0 7 1 6 2 5 3 4] (8 sigma values: [3 30000000 30 3000000 300 300000 3000 30000])
- ψ =1.5 (psi value)
- search method: structed grid with sampling 2 and zoom factor 0.75

In changing one by one these parameters, it ´s possible to see the real effect of each one on the algorithm behaviour.

2 Results with the simulation

2.1 Initialization points

In the first report [ref 1], 25 and 50 initialization points have been tested however just with few newpoints. In annex 2 are the results for 25 and 50 initialization points with respectively 200 and 250 newpoints and with the basic parameters. The comments in [ref 1] also work to describe the figures in Annex 2, indeed for 25 initialization points the best value found is higher than -12 and for 50 initialization points the algorithm finds a value smaller than -12 but cannot anymore improve it. One experiment with 100 initialization points has been done in order to compare it with the experiment taking 25 and 50 initialization points.

2.1.1 100 initialization points

The initialization points are important because the first model in the algorithm is constructed with these points. It ´s obvious that more points we have to build the model and better will be the model.

Appendix B

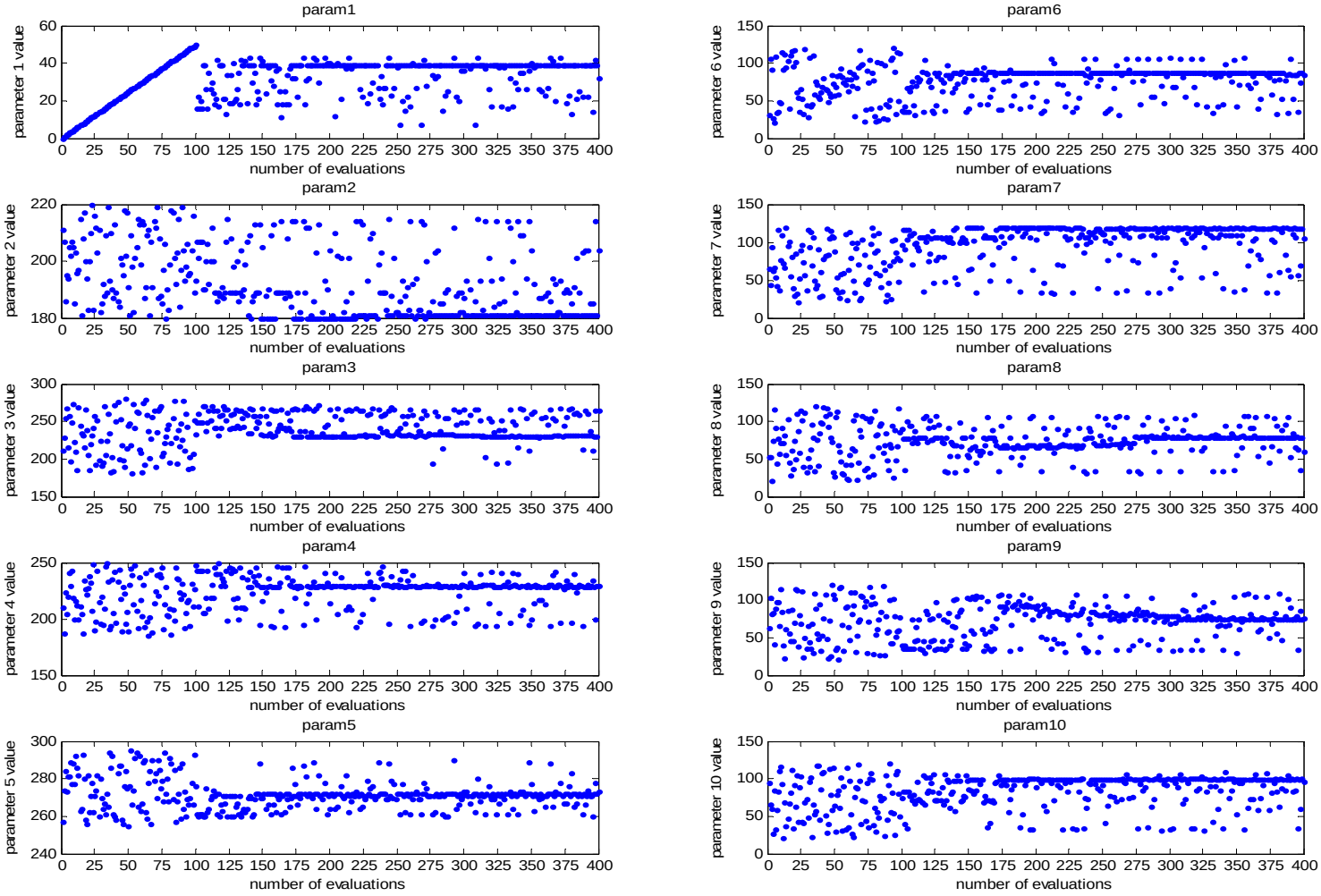


Figure 2: parameter values for 100 initialization points

Appendix B

In figure 2, we can see the ten parameter values for 100 initialization points and 300 newpoints. During the initialization (DOE), all the search space is explored and then the algorithm looks for finding the global optimum in a smaller search space. Indeed, for P3 for example, just few newpoints have been explored with a low P3 value (less than 240) and also for P1 whose few newpoints have been explored with a P1 value less than 10 whereas the best optimum found are for P1 value less than 15 [ref 2]. After the 172th point, for all the parameters, we can see a horizontal line which means that the parameter value doesn´t change anymore and stay almost constant (local search) which is not the case between the 100th and the 170th where the search seems more global. For some parameters, these lines are close to optimum parameter value [ref 2] like P2, P3, P4, P5, P7 and P10 but for others, these lines are far away the optimum parameter value. The algorithm behaviour is particularly bad for P1 but in looking in details the objective function in figure 3 for the DOE, we see that all the first points (with a low P1 value) are collision points. So the algorithm certainly doesn´t explore more the area with low P1 value caused of these collision points thinking this is a collision area. And we can see a lot of collision points during the initialization which are very difficult points for the model. Indeed, it increases the difficulty to build a model in more than the 10 dimensions difficulty.

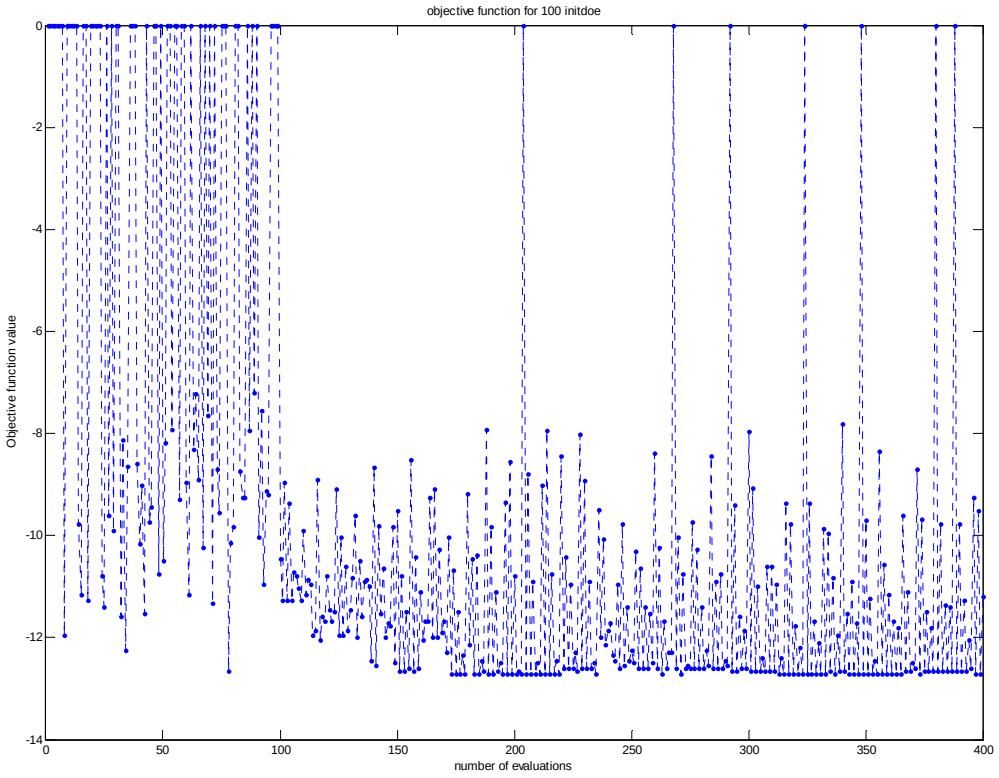


Figure 3 : algorithm behaviour for 100 initialization points

The points evaluated during the initialization can have different objective function values but most of them are above -12 so they are not close to the optimum but then after the initialization, the algorithm tries to find better points than in the DOE (the best point found in the DOE is -12.65 at the 77th point) in looking around these initialization points (local search) and also a little in all the search space (global search). The best point found by the algorithm is -12.71 at the 172nd point.

When the algorithm has found a better value than in the DOE, it seems that it cannot improve it again and set the parameter value close to the best point found at the172nd point and that is why there are lines in the parameters values (local search) plot and also in the objective function plot after the 172nd point. The reason is that local search in the neighbourhood of the best point so far gives approximately the same objective function value.

2.1.2 Conclusion for the initialization points

The results found with 25, 50 and 100 initialization points are (see Annex 2):

- 25 initialization points: the best point found is -11.72 at the 32nd point
- 50 initialization points: the best point found is -12.60 at the 174th point
- 100 initialization points: the best point found is -12.71 at the 172nd point

The best point has been found with the 100 initialization points knowing that during the initialization the algorithm has found a point with the value -12.65. It ´s not possible to know if the algorithm would find the same optimum if it doesn´t find a good point like -12.65 during the initialization. But with 100 initialization points, the model is expected better than with 25 or 50 initialization points and the results show that the best point is getting with 100 initialization points. In figure 3, figure 15 and 16, we can see that there always are a lot of collision points during the initialization. These collision points are interesting points to learn where are the dangerous area but as more than half of the initialization points are collision points, the construction of the surface for the model is very difficult with a lot of peaks. If we look the number of collision points after the initialization, we see on the figure 15 almost 17 collision points whereas in figure 3 almost 7 collision points and for the figure 16 almost 12 collision points. The algorithm with 50 initialization points is the most inefficient to predict the collision points for the model. For the algorithm behaviour, the 25 initialization experiment doesn´t provide very good results with an objective function value which stay almost constant around -11.71 and doesn´t improve itself. The algorithm behaviour for 50 or 100 initialization points is better than with 25 initialization points, indeed after the DOE it tries to improves the best value found but after almost the 175th point the algorithm becomes inefficient and doesn´t improve anymore the value. One of the problems for the algorithm with these two initialization points is the number of collision points for constructing the model which is really important. With few simulations in the DOE, the metamodel for the optimization method will be bad because it based on these few evaluations. The initialization points are really important for the algorithm, indeed they allow the construction of the first model but the collision points make the objective function nonlinear. For all the next experiment, the algorithm will be used either with 50 or with 100 initialization points.

2.2 Prediction and real value

The newpoint function (see Annex 1) in the algorithm uses a MATLAB package with a predictor function to build the metamodel. An important thing to know is: how the predictor function used in the algorithm looks the objective function when the algorithm knows just few points among 6 10^{18} possible points and when inside them are a lot of collision points? To see the behaviour of this predictor function with the model in 10 dimensions, the prediction values have been compared with the real objective values returned by the simulation for two cases, one with 50 initialization points and an other with 100 initialization points in order to see if there is a difference.

Appendix B

2.2.1 With 50 initialization points :

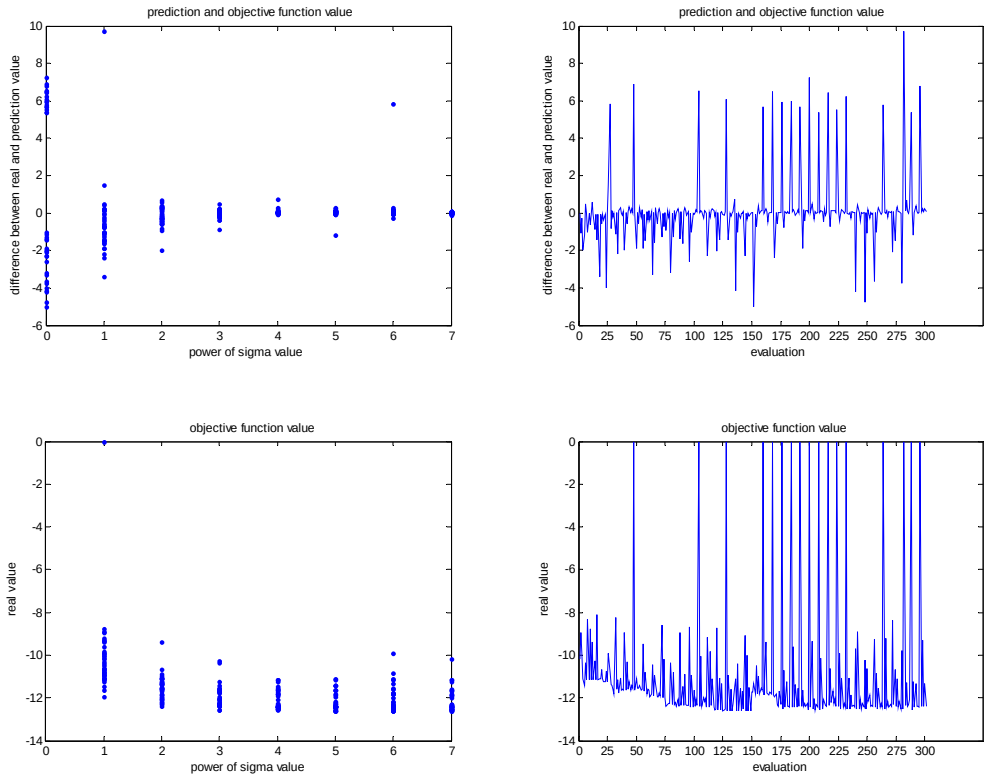


Figure 4: difference between prediction and real value for 50 initialization points

In figure 4, we can see 4 pictures with a vertical connection. The two first pictures (on the first line) show the difference between the real value returned by the simulation and the prediction value that is to say the value which is expected by the model. The first one on the left side depends on the power of sigma value (all the sigma value in the sigmalist are shaped as 3*10^(power of sigma value)) and the second one depends on the evaluation number (number of newpoints without the initialization points). In the first picture, we see that the difference is very large when the power of sigma value is equal to zero (when sigma is small) and improve itself when the power of sigma value increases until 4 and more. When the power of sigma value equals 4, the difference between the real production rate value and the prediction value becomes nearly zero. A small sigma value means a global search and a large sigma value means a local search. So when the algorithm is in a global search the model prediction is very bad but when it looks for the local search the prediction model become close to the true function. This result is what we expected, indeed when the algorithm is in a local search, the interesting points found by the algorithm are close to already calculated points, and close to them the predictor gives a good prediction. But the final goal is to find the global optimum and not the local optimum. As in the figure 12 in [ref 1], the search is too much concentrated close to the points found during the DOE. Indeed, there is a balance between the global and local search but with the actual sigma list values that the algorithm uses it seems more local than global search.

The second picture shows a lot of peaks which means a large difference between the model prediction and the real value but most of the positive peaks are collision points and it ´s very tricky for any model to predict these collision point. We can see easily the collision points with the peaks in the picture just below the second one. When there is a peak in the fourth picture (with the value 0), it ´s a collision point. We can also see some negative peaks, these peaks are the bad model prediction for low power of sigma value (global search). As results, we can say that the collision points are not well predicted by the model with 50 initialization points but it isn´t the alone problem because these points have a 0 value and the algorithm minimizes the objective function value. The other problem is the bad prediction for the global search, but by definition the global search is global so it hasn´t the same number of information than in a local search, that is to say when we look in a small space it is more easy to find the optimum than in looking for in a large space. The global search in 10 dimensions is a little based on luck (depend a lot of the initialization points). There are around 15 peaks (no collision peaks) with a difference inferior to 2 and around 12 peaks with a difference superior to 2. This isn´t so bad but the collision points are really annoying points for the model. The third picture shows the objective function value depending on the power of sigma value, we see that the best value has been found for a power of sigma value equal 4 (local search) and we also see that the low power of sigma value (global search) doesn´t provide good results for finding the global optimum. When the power of sigma value equal 0 or 1 the best point found is higher than -12 whereas with a power of sigma value superior to 3 the algorithm found some points under -12.

2.2.2 With 100 initialization points

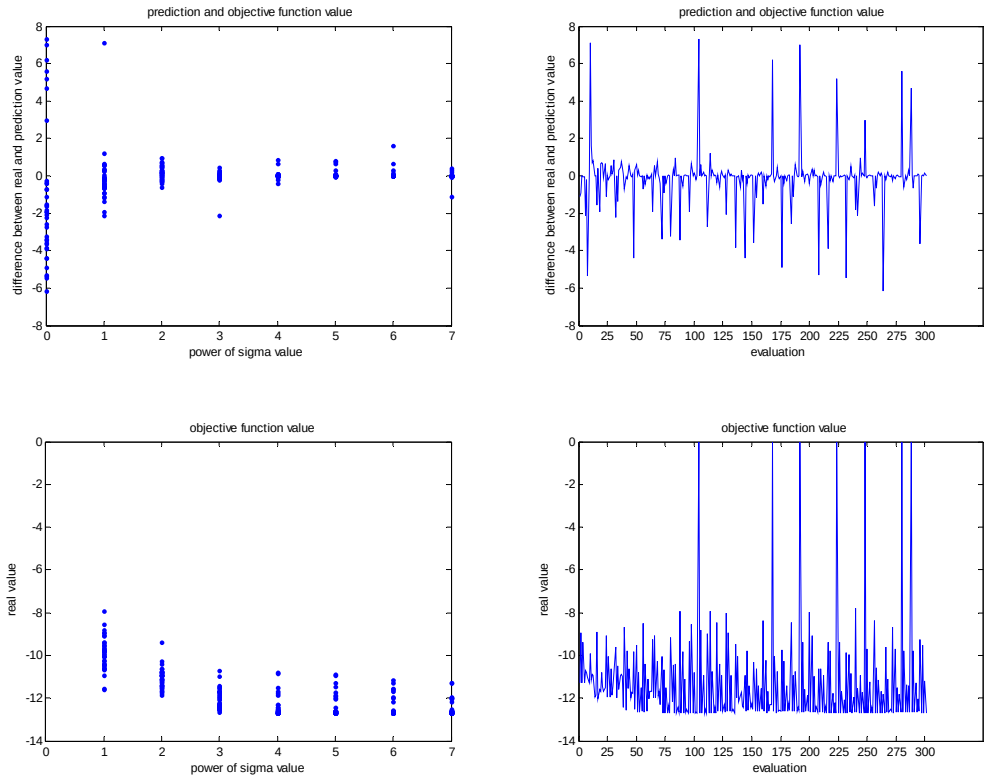


Figure 5: difference between prediction and real value for 100 initialization points

Appendix B

In figure 5, we see the same figure as in figure 4 but with 100 initialization points. One thing remarkable is that there are less collision points than with 50 initialization points either because the model is better and so predicts well some collision points and the algorithm doesn´t look around the collision points either it ´s only a stroke of luck. As there are some collision points, it ´s probably a stroke of luck to have only few collision points. The difference between the prediction value and the real value is still large in global search (for low power of sigma value) and nearly zero for local search (for high power of sigma value). More knowledge of the function gives a better metamodel for prediction. The best value found is still for the power of sigma value equal 4 (local search) but is better than with 50 initialization points. As already said, there is a compromise to find between the local and global search. For the difference between the prediction and the real value, there are 16 peaks with a difference superior to 2 and 10 peaks with a difference less than 2, it ´s quite identical with 50 initialization points. All these peaks show that the model isn´t yet efficient for the actual objective function and let us to think that it ´s maybe a bad idea to use a RSM method to optimize a function in 10 dimension with only few information about it and collision points everywhere.

2.3 Psi and sigma influence

The algorithm uses a weight function to build the auxiliary function which is used in the newpoint function (see Annex 1). This weight function is: $\omega(\tilde{x}) = \exp(-\sigma\tilde{f}(\tilde{x})^{\psi})$ where $\tilde{f}(\tilde{x})$ is the normalization of the predicted function values given by the metamodel. This weight function depends on ψ (psi) and σ (sigma).

2.3.1 Coefficient psi

In the figure 6, the weight function is shown for six different choices of ψ for the surrogate model function $\tilde{f}(\tilde{x}) = x^2$ when σ = 1. For small values of ψ more focus will be set on the local neighbourhood of the minimum point of the surrogate model.

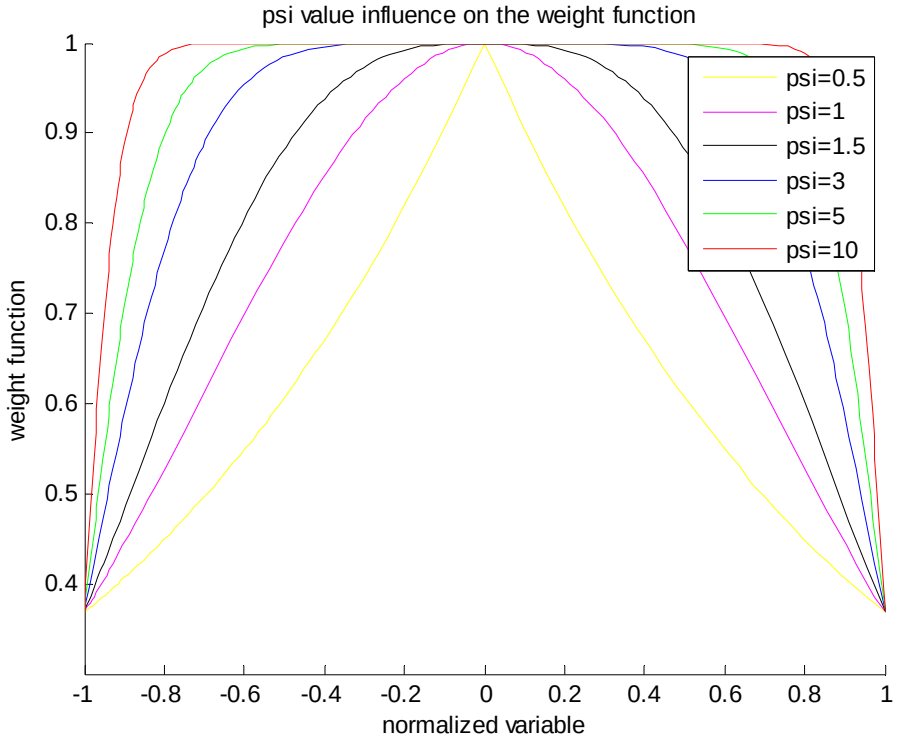


Figure 6: psi influence on the weight function

According to the figure 6, increase the ψ value will influence the search type of the algorithm in a more global search than with small ψ value. The confirmation is given with the experiment as shown below, not with $\tilde{f}(\tilde{x}) = x^2$ but with the normalization of the predicted objective function made with the model and ψ=3.

Appendix B

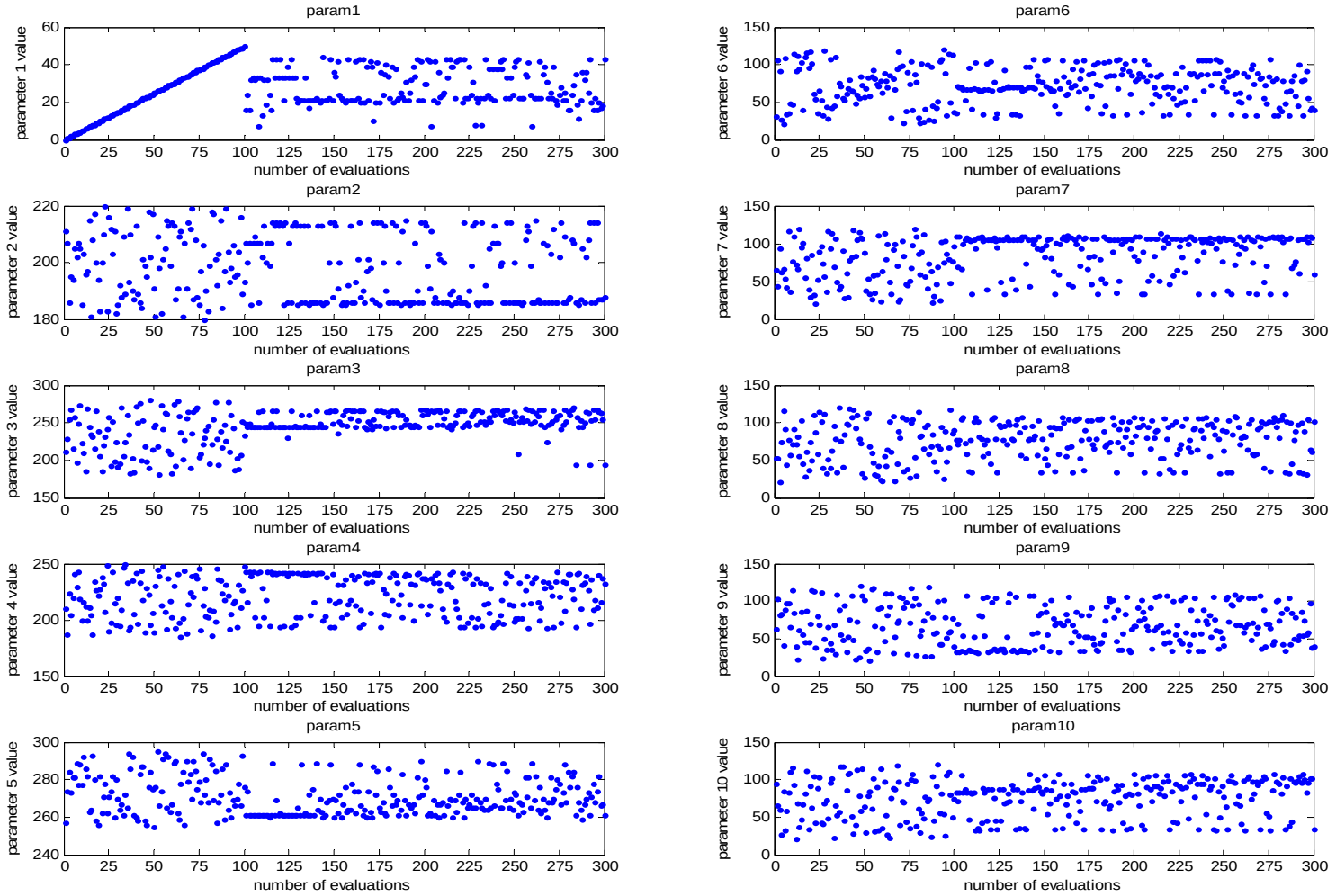


Figure 7: parameter values for psi=3

Appendix B

In figure 7, we can see the ten parameter values for 100 initialization points and 200 newpoints in setting ψ =3. There are no changes for the initialization points because it ´s always the same DOE. We can see some lines between the end of the initialization points and the 150th point, these lines means that the algorithm is more in local search than in global search and look around the initialization points which are interesting. The good thing is what we expected in tuning ψ =3, the lines have almost disappeared and the search is more global than local after the 150th point. So increase the ψ value has a real effect on the algorithm behaviour but the global search for example for P1 doesn´t explore a lot of low P1 value less than 15 and also for P3 doesn´t explore a lot of low P3 value less than 240. The strange thing is that the experiment with ψ =3 has almost erased the lines after the 150th point but new lines appear between the end of the DOE and the 150th point. Just after the DOE and particularly for the 5 first parameter, the parameter values are fixed and then the search is better than with ψ =1.5. For P2 and P7, the search after the 150th seems local but if we look more in details, we see that the P2 and P7 values are fixed but close to values which give an optimum for the objective function [ref 2]. For all the others the search is well global after the 150th point. The search has well been moved in a more global search than with ψ =1.5 and increase ψ has a real effect on the objective function results.

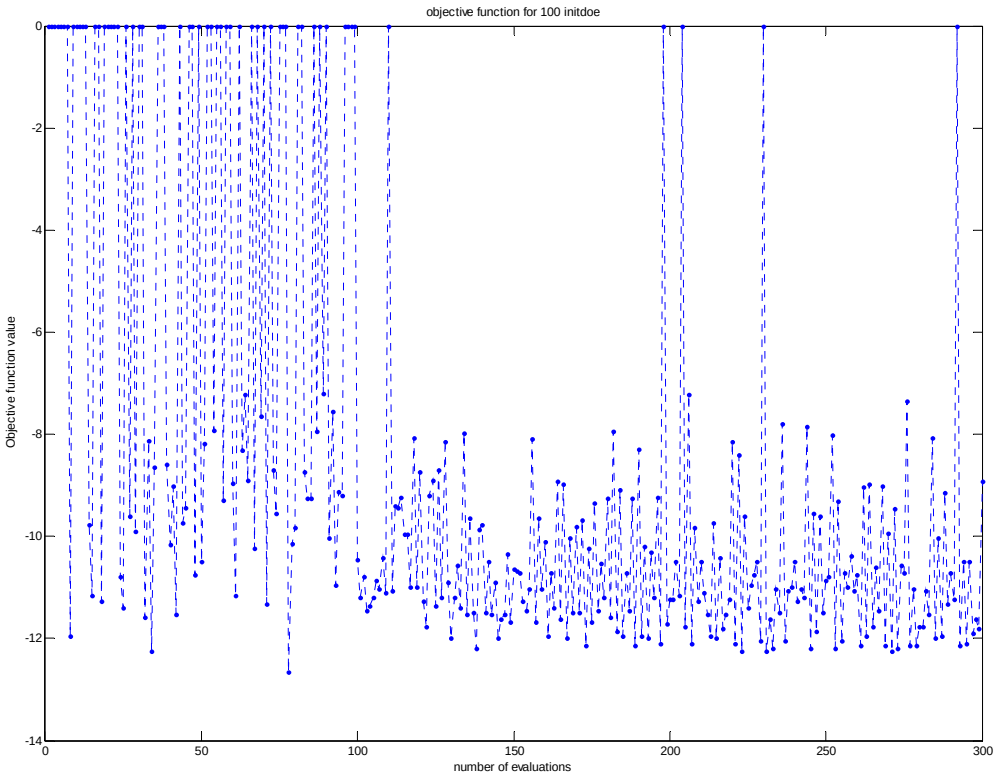


Figure 8: algorithm behaviour for psi=3

Indeed, as we can see in figure 8, the points evaluated after the initialization give worst results than with ψ =1.5. The best value found is -12.65 at the 77th point during the DOE. As the lines have disappeared for the parameter values after the 150th point, the objective function results have changed and there isn´t anymore a constant value for the objective function after the 150th point. This is what we expected because as the search has been moved to a more global search, two successive points can be away each other and this give what we can see on the figure 8.

The problem is that the algorithm doesn´t succeed to improve the objective function value, there is no negative slope which could show an improvement of this last and the best value found after the DOE is -12.3. So ψ =3 well give a better global search than with ψ =1.5 but the results are bad for optimum research. Here we see the global search problem like in all the algorithm, the global search looks in a bigger space than a local search but the results are less good because to find good results, it needs a local search. The balance between the local and global search is tricky and can also been changed with the sigma value.

2.3.2 Coefficient sigma

In the figure 9, the weight function is shown for seven different choices of sigma value for the surrogate model function $\tilde{f}(\tilde{x}) = x^2$ when ψ = 1.5. The σ variation has the same behaviour than ψ variation but with opposite value as shown below. For low σ value, we see a surface on the top which means that the search will be more global than with high σ value where there is a peak and so a more local search.

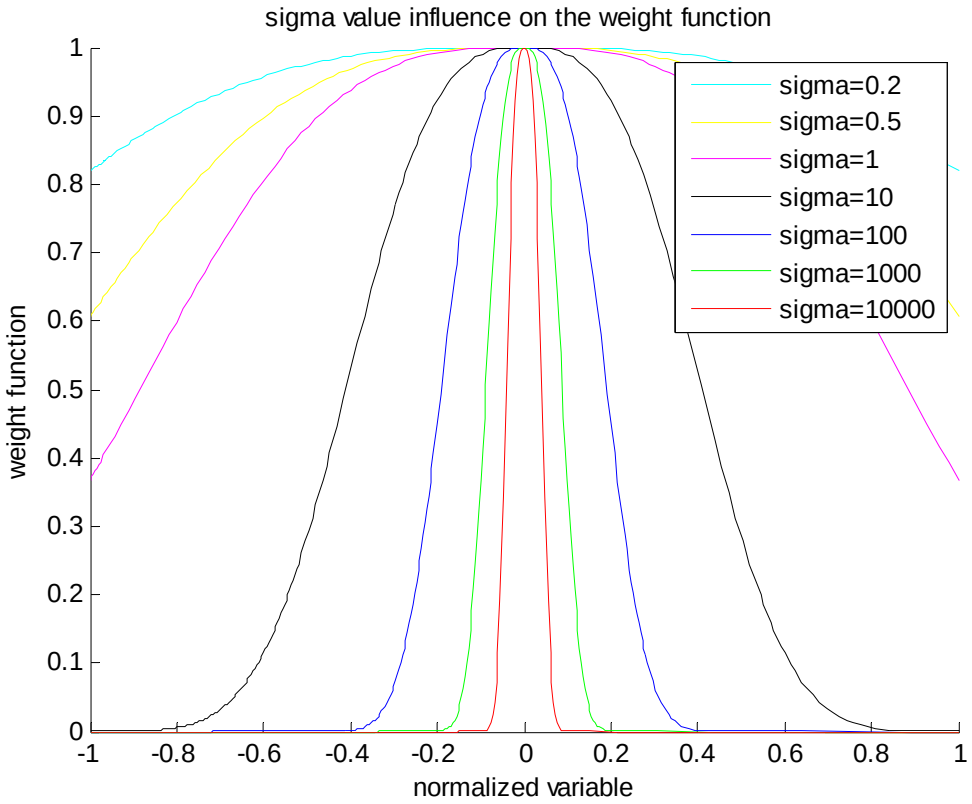


Figure 9: sigma influence on the weight function

As said above, the balance between the local and global search is tricky and an idea will be to add high sigma value in the sigma list in keeping ψ =3 to see the algorithm behaviour in hoping that the algorithm keeps a good global search and give better results for the local search with the high sigma values added.

Appendix B

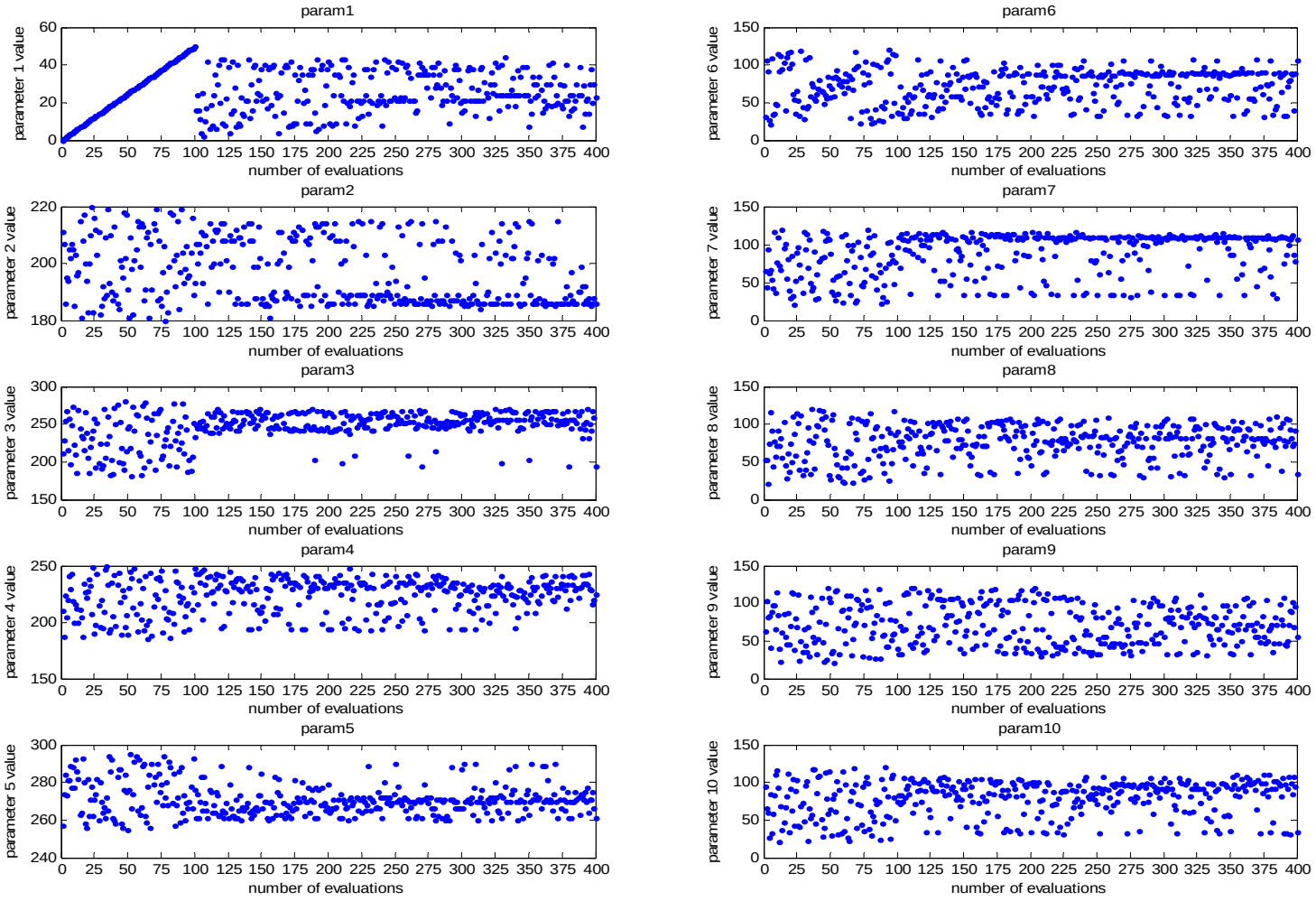


Figure 10: parameter values for two sigma values added and psi=3

Appendix B

In figure 10, we can see the ten parameter values for 100 initialization points and 300 newpoints in setting ψ =3 and the sigma list: 3*10 ^ [0 7 1 12 6 2 24 5 3 4]. We can see the global search with points in all the search space or almost, this is the results of ψ =3 but some new lines or points close to each other reappear. The search space for P1 is better explored than previously with points everywhere in the boundaries even if there is still only few points with low P1 value. As said in [ref 2], there is a collision area for P3 and the algorithm seems to well predicted this collision area and that is why the algorithm doesn´t look for low P3 value. So this is really important, the algorithm is able with some points of the DOE to predict a collision area (at least for the parameter3). The new lines or points close to each other are the results of the two high sigma value added. We can see these lines on P2, P5, P6, P7 and even P1, this local search behaviour is important to find the best point as possible but in our case it doesn´t find a better point than during the DOE as we can see on the figure 11. In fact the tune with ψ =3 and sigma list : 3*10 ^ [0 7 1 12 6 2 24 5 3 4] improve the search for the parameter value because the algorithm looks in a bigger part of the search space so a bigger search space is explored and the parameter don´t keep the same value and focus a little less on local search. But the figure 11 show that it doesn´t improve the value of the best point found.

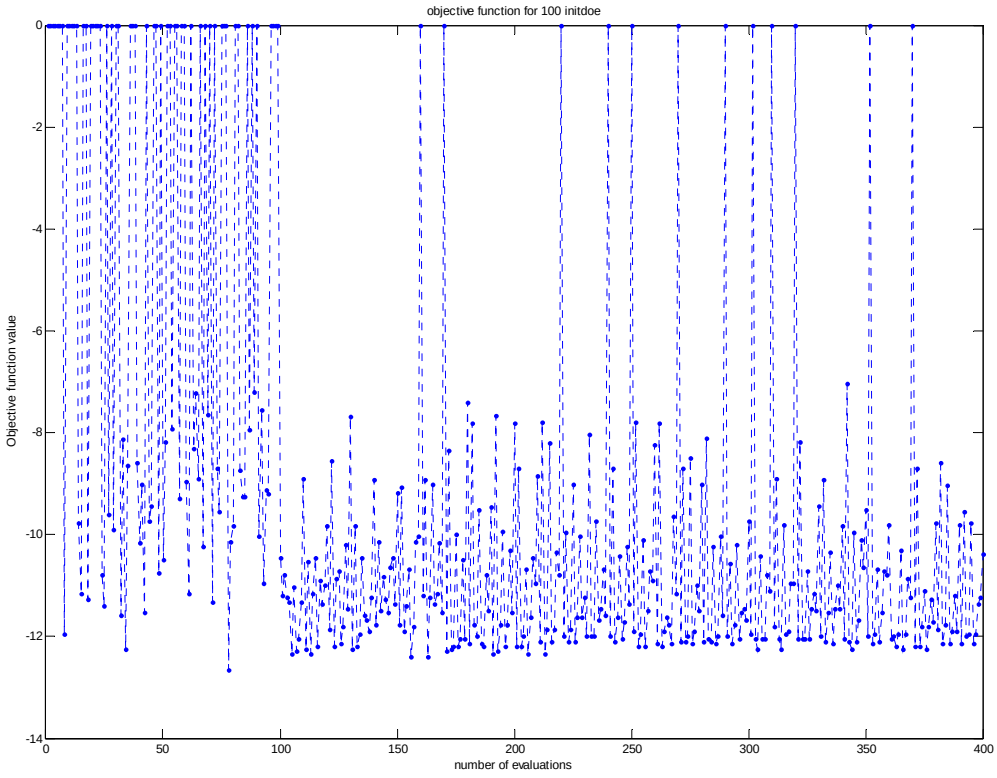


Figure 11: algorithm behaviour for two sigma value added

The best point is -12.65 and is found at the 77th point during the DOE. After the initialization, we can see a large difference between two successive points, one with a value close to -12 and an other one with a low value or even a collision point exactly as previously (figure 8 and figure 3). Unlikely at the first experiment with 100 initialization points where the objective function value remains almost constant, here we don´t see constant value but it ´s because the parameter values don´t remain constant (global search) and several successive points can be very away each other.

One thing remarkable is the number of collision points after the initialization, indeed the previous experiment only gives 7 and 5 collision points but here we get 12 collision points. We observe the same behaviour than previously, there is no improvement of the best point found during the initialization whereas the goal of the algorithm is to optimize that is to say to try to always improve the objective function value. The results are better with these two high sigma values than without, indeed the objective function value is often under or close to -12 whereas without the two high sigma values, there was just few points close to -12 and all the other were bad or collision points. So the two high sigma values improve a little the local search but not completely.

2.4 Correlation and regression functions

The algorithm used to optimize the objective function is based on RSM. This kind of algorithm builds a surface representing the function it wants to optimize. In our case, the model is built with the kriging package and this package provides different correlation and regression functions [ref 3].

2.4.1 The different correlation and regression functions

The toolbox provides regression models with polynomials of orders 0, 1 and 2 called regpoly0, regpoly1 and regpoly2.

It also contains 7 choices of correlation models :

- Exponential
- Generalized exponential
- Gaussian
- Linear
- Spherical
- Cubic
- Spline

These correlation functions can be separated into two groups, one containing functions that have a parabolic behaviour when the points are close to each other (Gauss, Cubic and Spline) and the other containing function with a linear behaviour when the points are close to each other (Exp, Lin and Spherical) [ref 3]. The general exponential Expg can have both shapes depending on one parameter and that is why this is this correlation function which is actually used in the algorithm.

To see the difference between these different correlation and regression functions, the MATLAB file used in [ref 2] to study the function in 10 dimensions has been used. It shows the production rate depending on P3 and P5 with P1=0 (see Annex 3).

The thing remarkable is that all the correlation functions with a linear behaviour when the points are close to each other (Exp, Lin and Spherical) look almost like the same (see figure 18, 19 and 20), that is to say an optimum area for P3=220:235 and P5=255:275, a collision area detected when P3 is close to 200 and P5 to 265 and a kind of break between the two areas.

An other interesting thing is the results for corrlin in figure 18 where we can see the optimum value which is the same than for the other correlation functions but the lower value 4 is closer to the collision value which is 0 with corrlin than with the other correlation functions where it ´s a little less than 6. So it could be possible that with the linear correlation function the collision points are well predicted. The plot for corrgauss in figure 21 is difficult to explain but it doesn´t give good results because the optimum area and the collision area are well known and are in a wrong place with hazardous production rate value. The plot with corrspline in figure 22 looks like quite the same to the plot with correxpg in figure 17, indeed the optimum area is well detected but not the collision area whereas with correxpg we can see a dark blue in left corner of the plot which seems to be the collision area. I didn´t succeed to make the plot with corrcubic so we cannot see if it will be an interesting correlation function for our case but as the two other correlation functions with parabolic behaviour when the points are close to each other are bad, it will be probably also a bad result.

The figure 18 with correxpg well show that it is between the two different kinds of correlation functions, indeed we can see the optimum area, the collision area and even the break but it also closes to the plot with corrspline.

To conclude for the correlation models, correxpg well seems a compromise between the two different correlation functions but it will be interesting to test the linear correlation function which can maybe better predict the collision points than correxpg.

For the regression functions, the figure 24 and 25 look like hazardous but it ´s because to use regpoly 1 or 2 it ´s necessary to tune special coefficient for the regression function.

2.4.2 Linear correlation

As seen in figure 18, the linear correlation seems to better predicts the collision points than the other correlation functions so it could be a real improvement for the model.

Appendix B

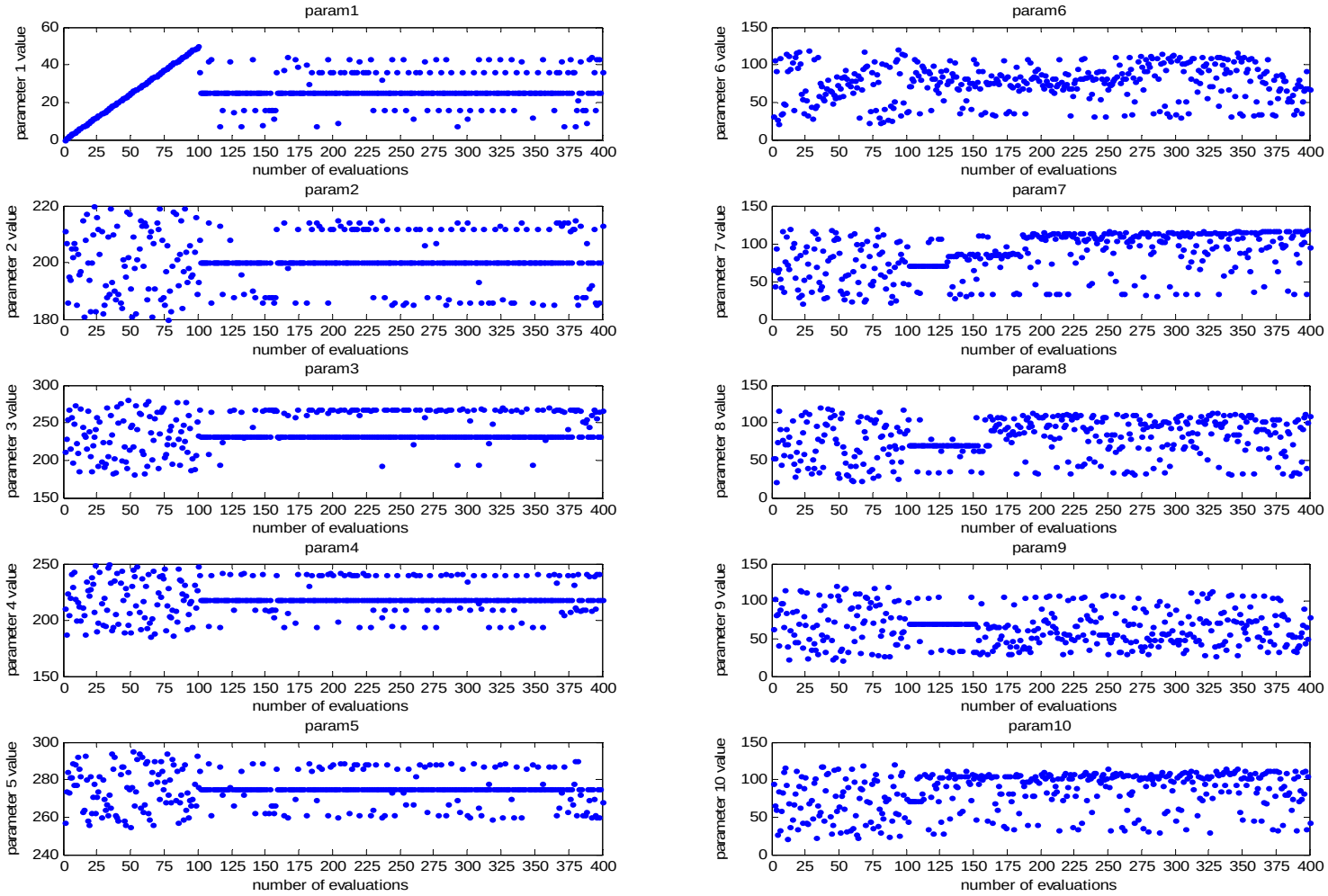


Figure 12: parameter values for linear correlation

Appendix B

In figure 12, we can see the parameter values for the 100 initialization points and 300 newpoints in setting the linear correlation function. The first thing remarkable is the constant value for the 5 first parameters (from P1 to P5, Position parameter) during the 300 newpoints which means that the search for this five parameters is clearly local and not global at all. In addition, the constant parameter value are far away the optimum parameter value. For the last 5 parameters (Velocity parameter) it ´s a little bit better but there are some lines just after the DOE (for example for P8 and P9) which means that directly after the initialization points the algorithm fix the parameter constant so it ´s not convenient to find the global optimum. After almost the 150th point, we see a better global search for the last five parameters but all the search space is not explored. So the linear correlation function makes the algorithm searching more in a local search than in a global search. But the results are not as bad as with the the case psi=3.

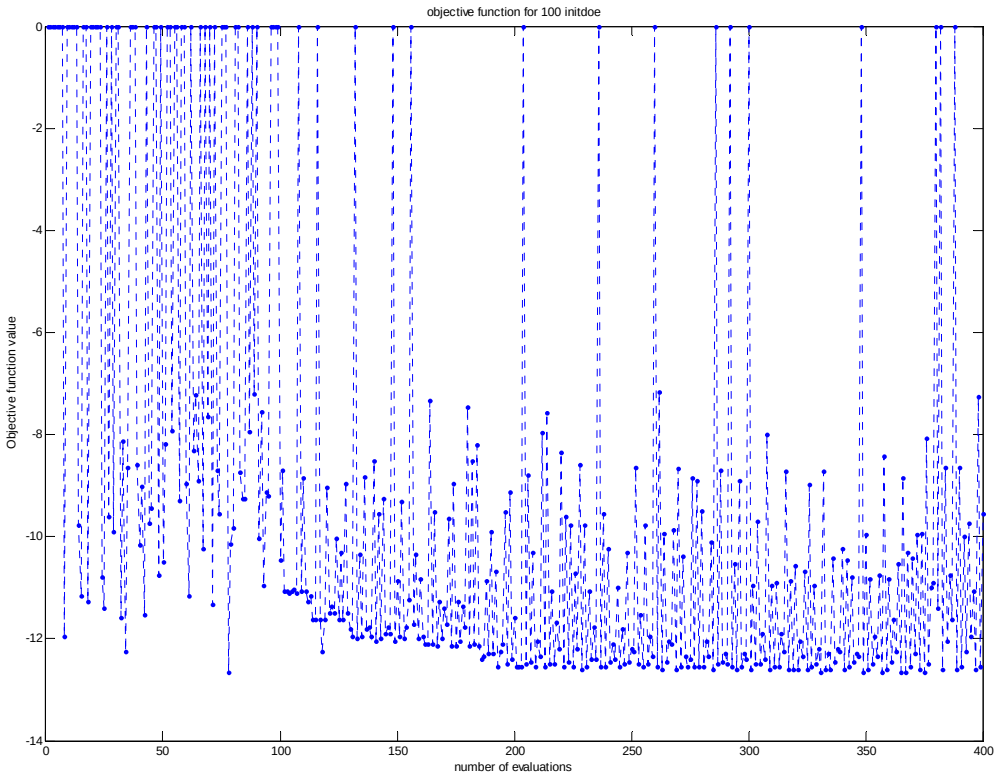


Figure 13: algorithm behaviour for linear correlation

Indeed, in figure 13 we can see the algorithm behaviour with the linear correlation and we see the negative slope just after the initialization which means that the algorithm try to improve the value found during the DOE even if the 5 first parameter doesn´t change. Then the objective function value is almost always under -12. This constant value for the objective function is getting because the algorithm looks too much in local search and stay around the best point found during the initialization. The best value found is still -12.65 at the 77th point during the initialization. We can also see a lot of collision points which show that the linear correlation doesn´t predict as expected the collision points. Compared to the figure 3, there is not a lot of difference. There are a little more collision points but otherwise the generalized exponential correlation function and the linear correlation function have the same effect on the objective value. But in figure 2, the parameters are more constants than in figure 12 so the search seems a little more global with the linear correlation.

2.5 Results conclusion

The best objective function value found by the algorithm is -12.71 with 100 initialization points and all the basic algorithm parameters after 172 simulated points. To compare with other algorithm, for example Direct has found -12.71 after 205 evaluations whereas our algorithm based on RSM has found it with 172 evaluations. One of the problems is that this algorithm cannot be launched for a lot of points caused of its calculation time (chapter 3.3, [ref 1]) and when it has found the -12.71 point, it cannot improve it with the number of evaluated points tested contrary to Direct. And as seen figure 3, the algorithm after having found this value doesn´t change anymore the parameter value.

The results of the algorithm with 25 and 50 initialization points are worst than with 100 initialization points so at least 100 initialization points should be used to build the first model in the algorithm to optimize a function in 10 dimensions. All the tests made with the different algorithm parameters don´t improve the best result found but improve the search, indeed set psi=3 move the search in a more global search and add two high sigma value improve the local search. There is a compromise to find between the global and local search, exactly as in the 'mix method' [ref 4].

Indeed, the 'mix method' uses an algorithm for the global search and then switches to an other algorithm for the local search. In the algorithm based on RSM this balance between global and local search is done with the sigmalist and the psi value but isn´t yet efficient in 10 dimensions. The problem is the same as in the 'mix method', we need a global search to find the interesting areas but then we need the local search to find the global optimum. When swith from global search to local search is a difficult issue.

An other problem of the algorithm based on RSM is that it depends a lot of the initialization points. As seen figure3, for the objective function we actually want to optimize, more than half of the initialization points are collision points so more than 50% of the initial information just notice where are the collision points but don´t help in the global optimum search. These collision points are a real problem for constructing the surface. The 'mix method' also depends on the points found by the global search algorithm but as it doesn´t build a surface the collision points don´t affect the algorithm behaviour as when the algorithm build a surface.

The algorithm based on RSM hasn´t been launch for high number of newpoint because I expect that the algorithm will have the same behaviour as in figure 3. But it will be interesting to do it just to see if the model with 1000 points can well predict the objective function results. The test with the linear correlation function is similar to the test with the Generalized exponential correlation function and doesn´t improve the model as expected.

Conclusion

An algorithm based on RSM has been tested to optimize a function in 10 dimensions. All the results show that the algorithm is not yet efficient for the simulation based optimization problem. The idea to use a model in 10 dimensions to optimize a function with only little information about this function is maybe not a good idea. In addition, this function is not a common function because it can have collision points everywhere and the optimum are close to the collision points. So two points close to each other can have a completely different value and it ´s a real problem for the construction and the prediction of the model.

It ´s necessary to improve the algorithm in order to use it for the simulation based optimization problem in 10 dimensions.

Bibliography:

ref 1. Implementation of a Response Surface Model based global optimization method for Virtual Manufacturing, TORRES Stephane, University West, 2009

ref 2. Study of a function in 10 dimensions, TORRES Stephane, University West, 2009

ref 3. DACE, A MATLAB Kriging Toolbox, Soren N. Lophaven, Hans Bruun Nielsen, Jacob Sondergaard, 2002 (http://www2.imm.dtu.dk/~hbn/dace/).

ref 4. Study of the ‘mix method’, Damien RINGENBACH, University West, 2009

Annex 1:

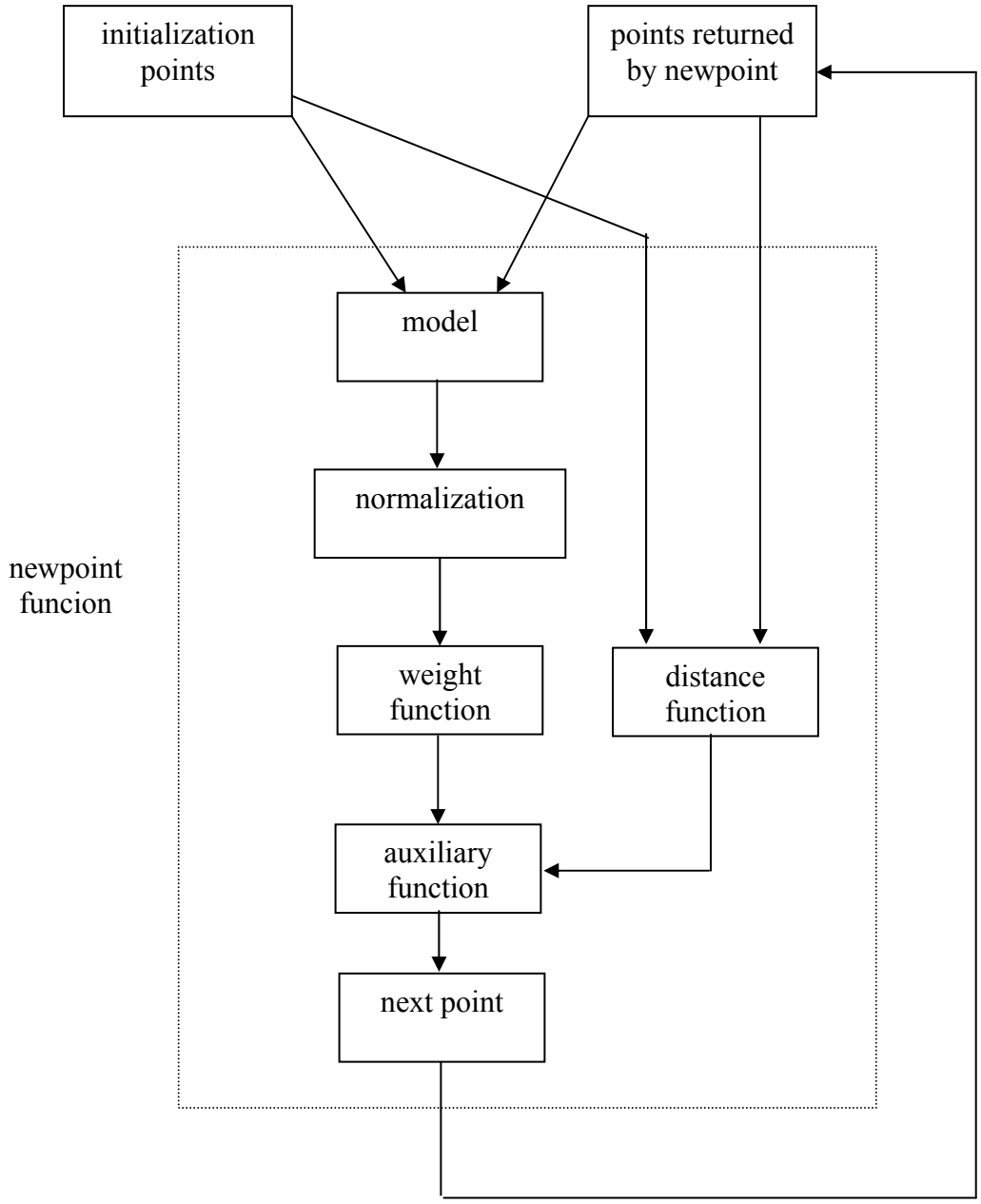


Figure 14: algorithm structure

Figure 14 completes the description of the way the algorithm works [ref 1]. The initialization points are got with the initdoe function. With these points and with the points returned by the newpoint function (at the first point there are only the initialization points), a model is built. With the normalization of this model, a special function is created: the weight function. Another function is created with all these points, it ´s the distance function which is in fact, just a measure of the uncertainty in the model. It depends on both the shortest Euclidean distance to the evaluated design variables and also the shortest distance to the boundary of the domain. In combining these two functions, the auxiliary function is built and the next point is found in maximizing this auxiliary function.

Annex 2:

This Annex 2 presents the algorithm behaviour with 25 and 50 initialization points.

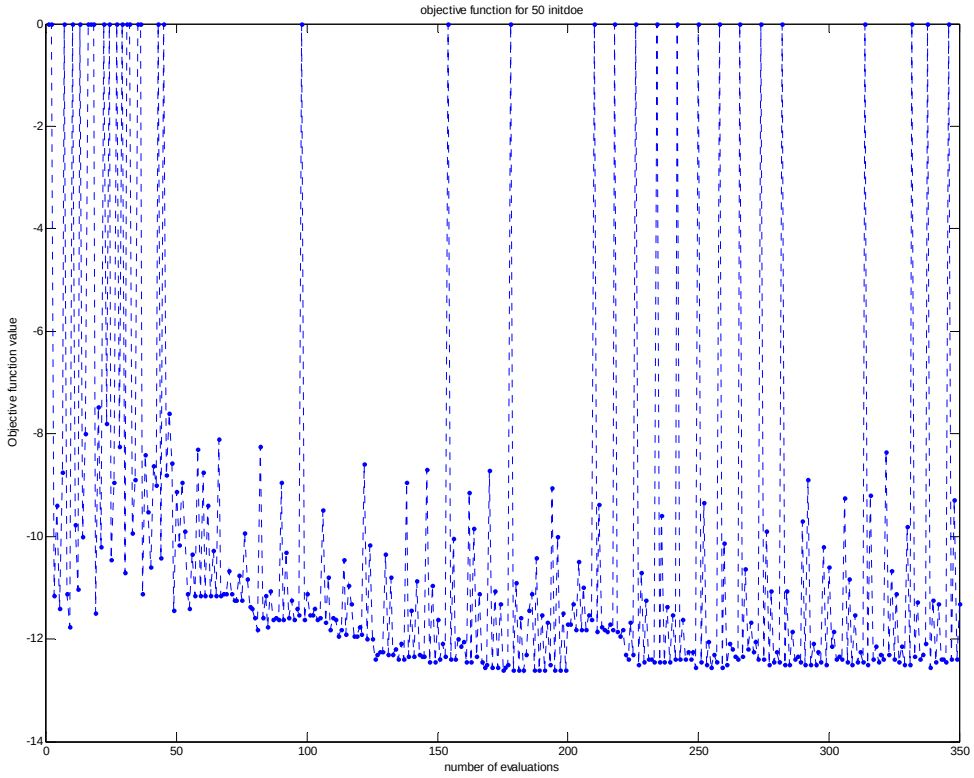


Figure 15: algorithm behaviour for 50 initialization points

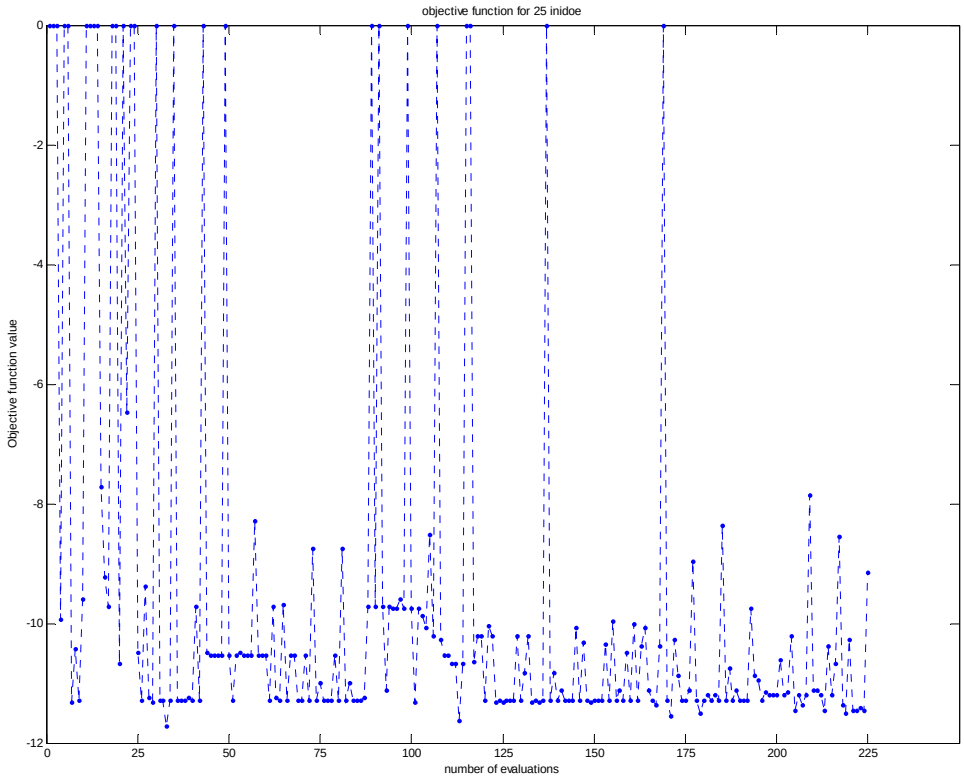


Figure 16: algorithm behaviour for 25 initialization points

Annex 3:

This Annex 3 presents the plot of the production rate depending on P3 and P5 with P1=0 for different correlation and regression functions.

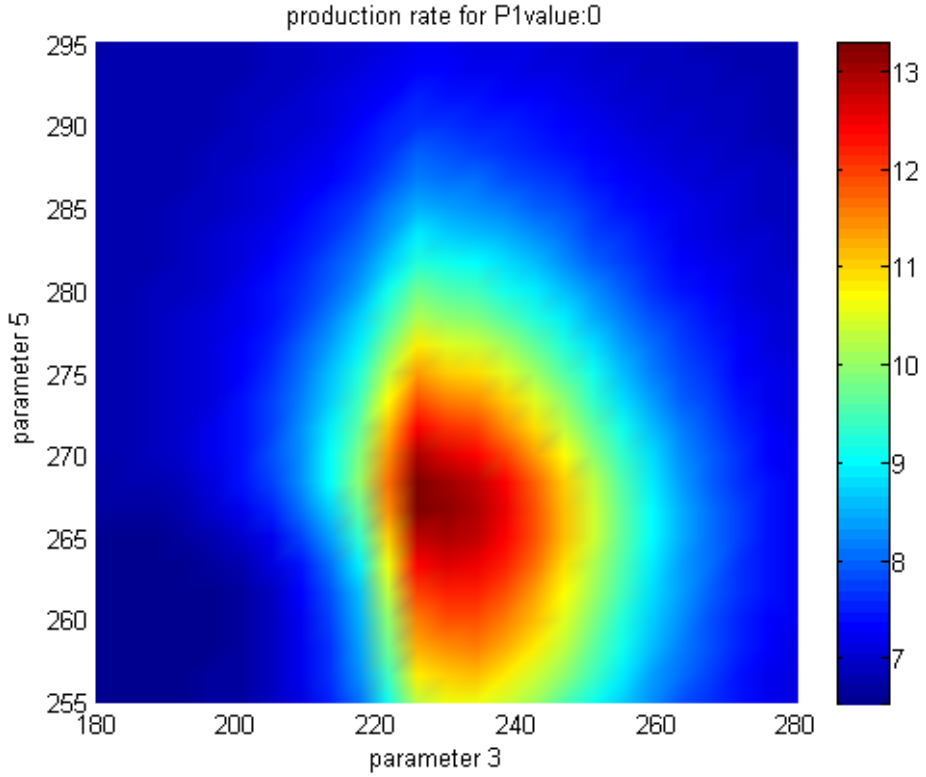


Figure 17: production rate with a model using correxpg

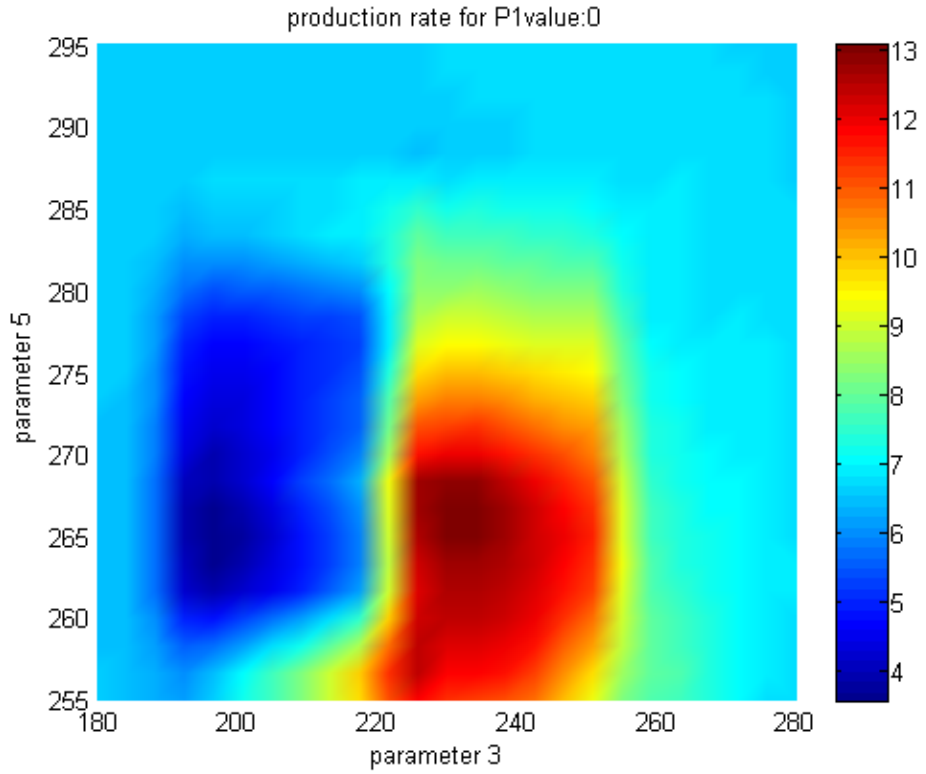


Figure 18: production rate with a model using corrlin

Appendix B

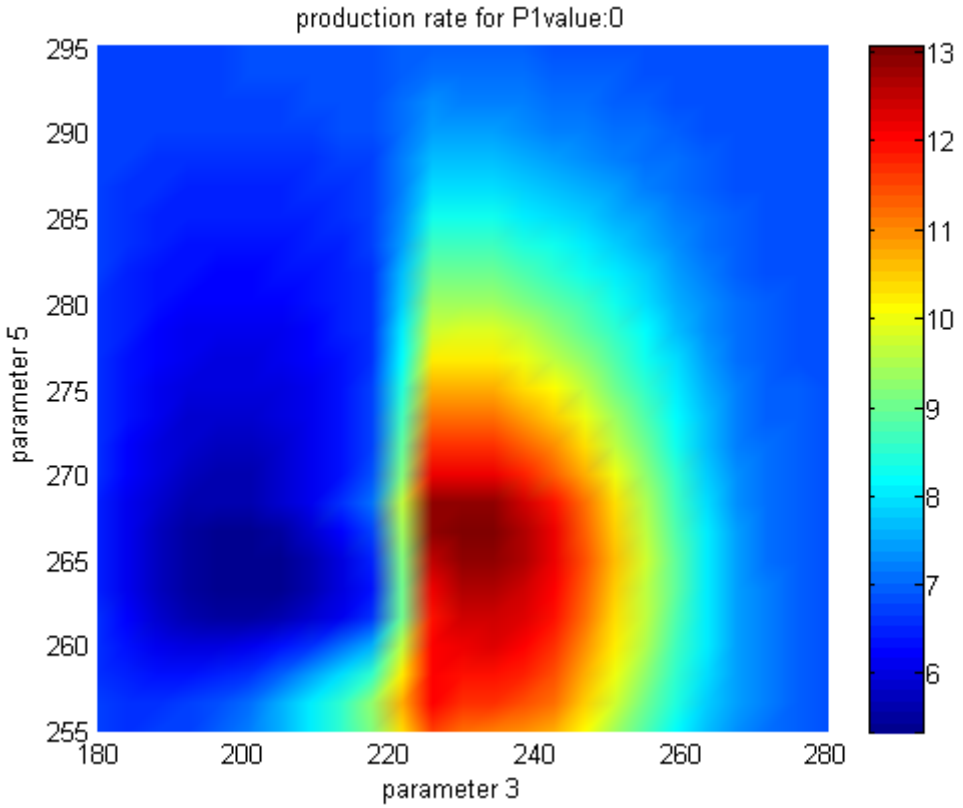


Figure 19: production rate with a model using corrspherical

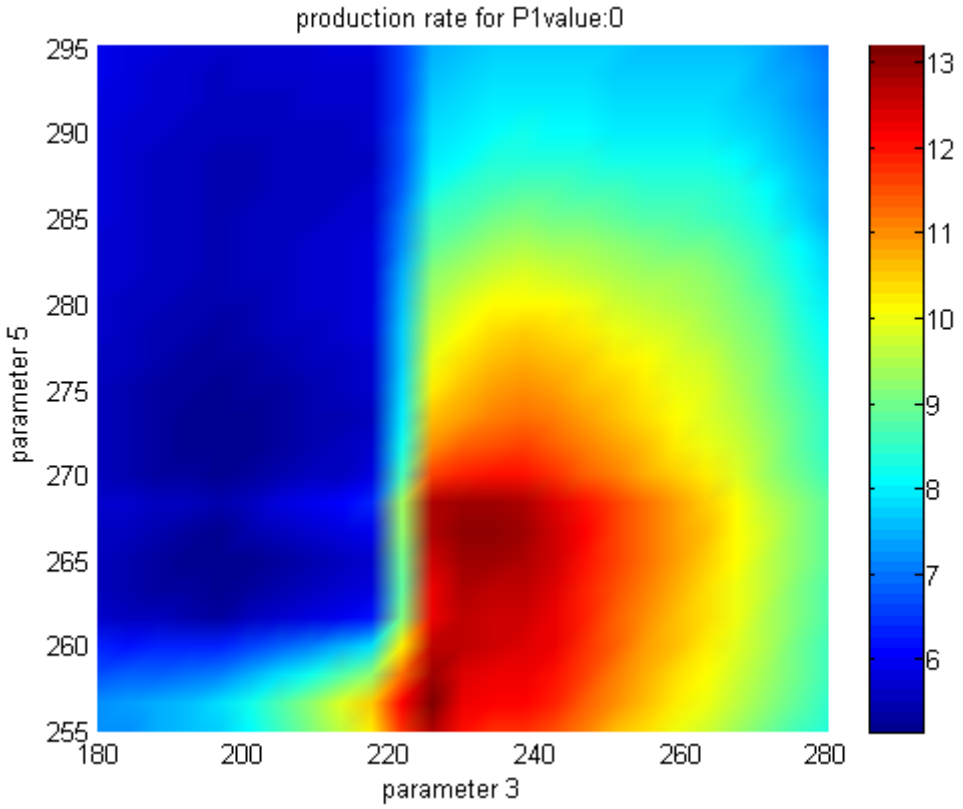


Figure 20: production rate with a model using correxp

Appendix B

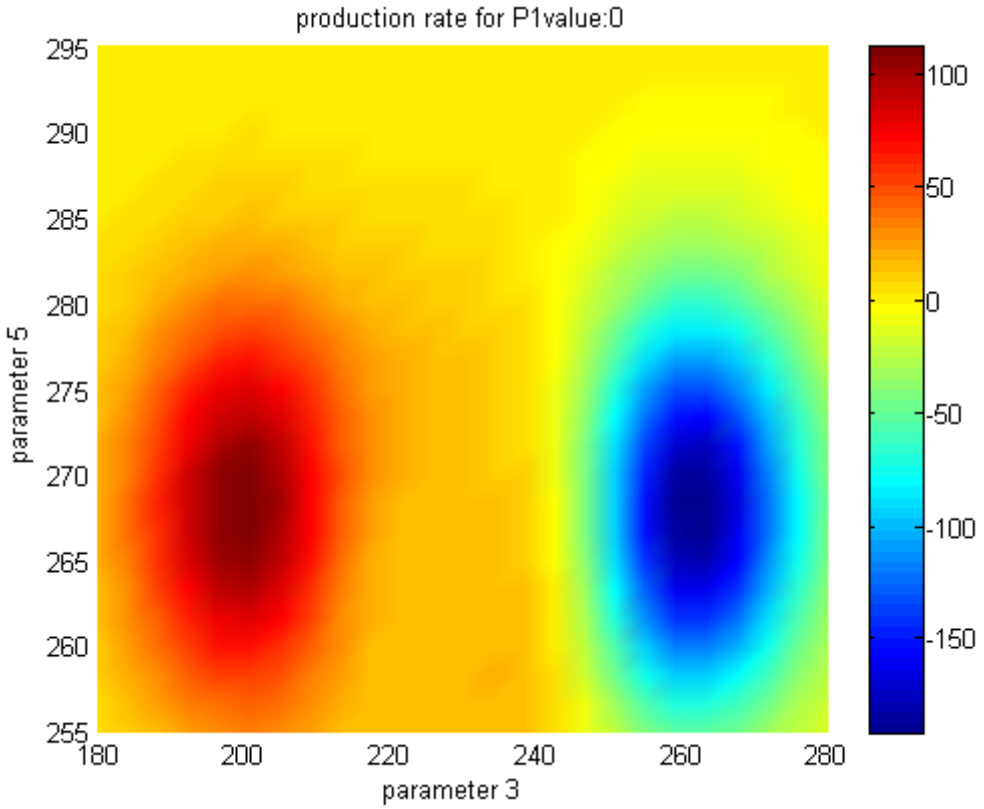


Figure 21: production rate with a model using corrgauss

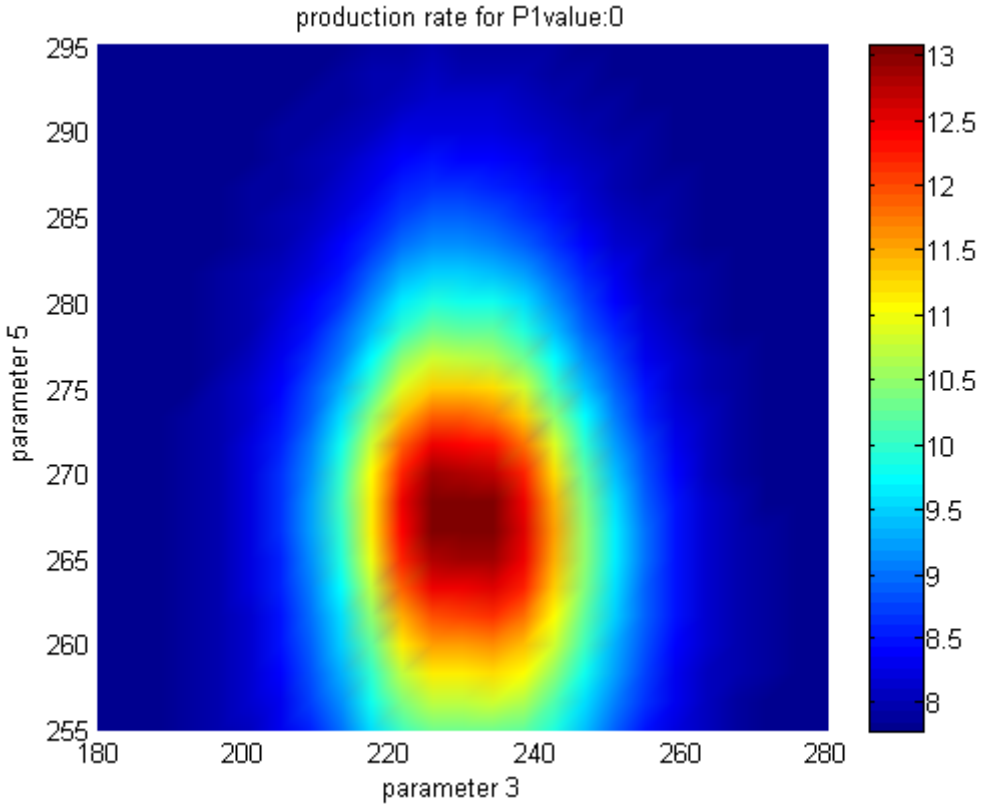


Figure 22: production rate with a model using corrspline

Appendix B

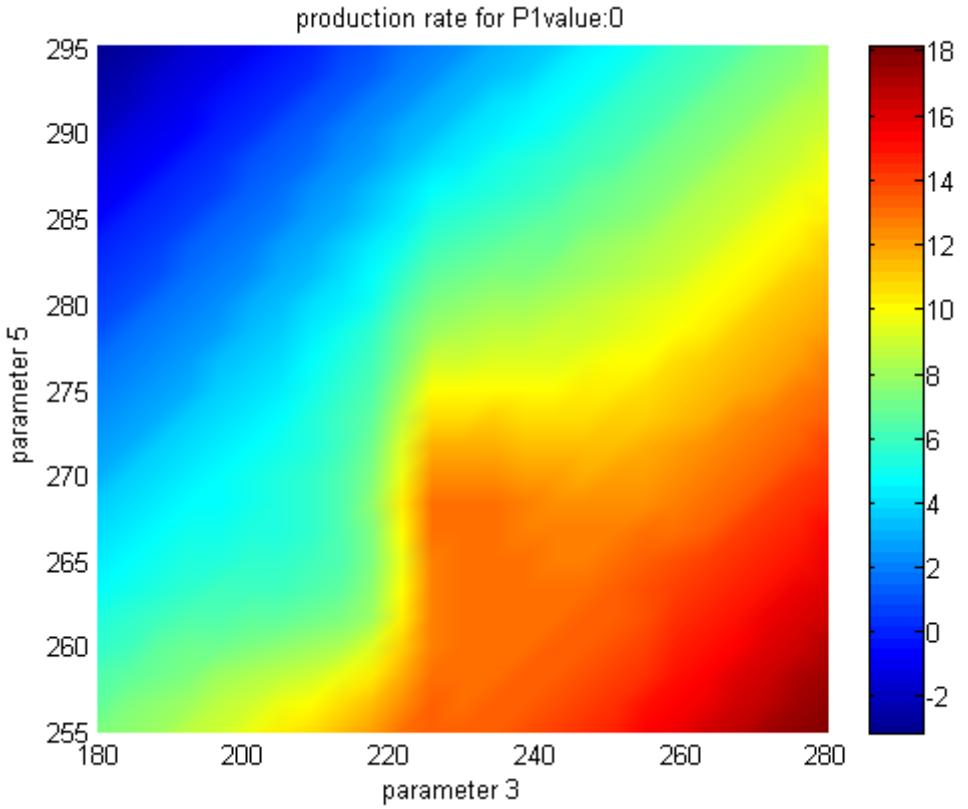


Figure 23: production rate with a model using regpoly1

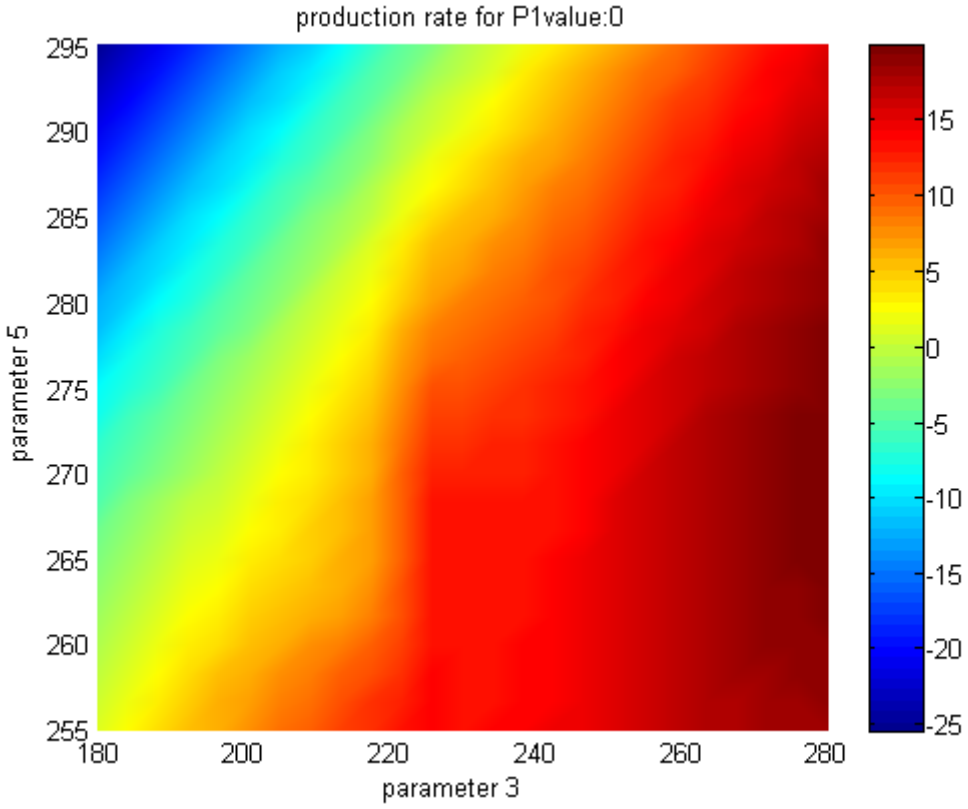


Figure 24: production rate with a model using regpoly2

Annex 4:

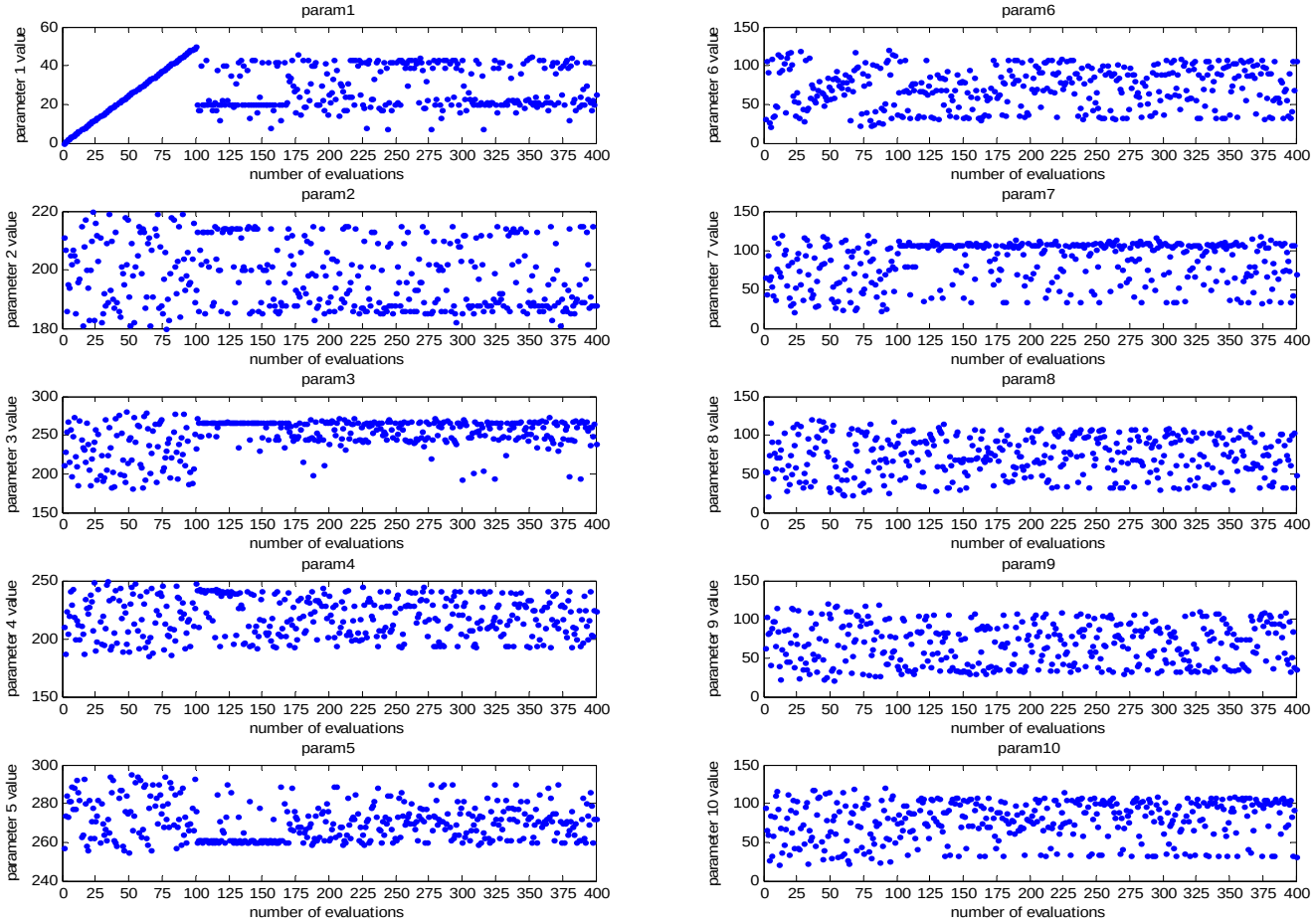


Figure 25: parameter values for search method: structed grid with sampling 1.8959 and zoom factor 0.79118 (structgrid=false)

Appendix B

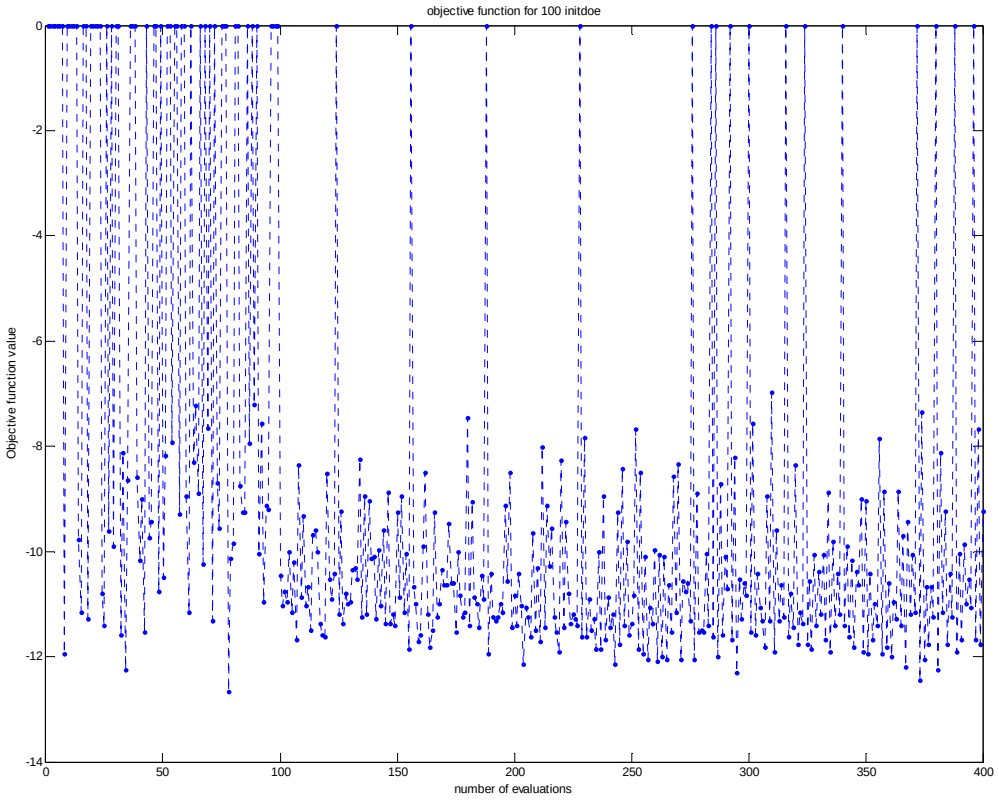


*Figure 26***:** *algorithm behaviour for search method: structed grid with sampling 1.8959 and zoom factor 0.79118 (structgrid=false)*

Annex 5:

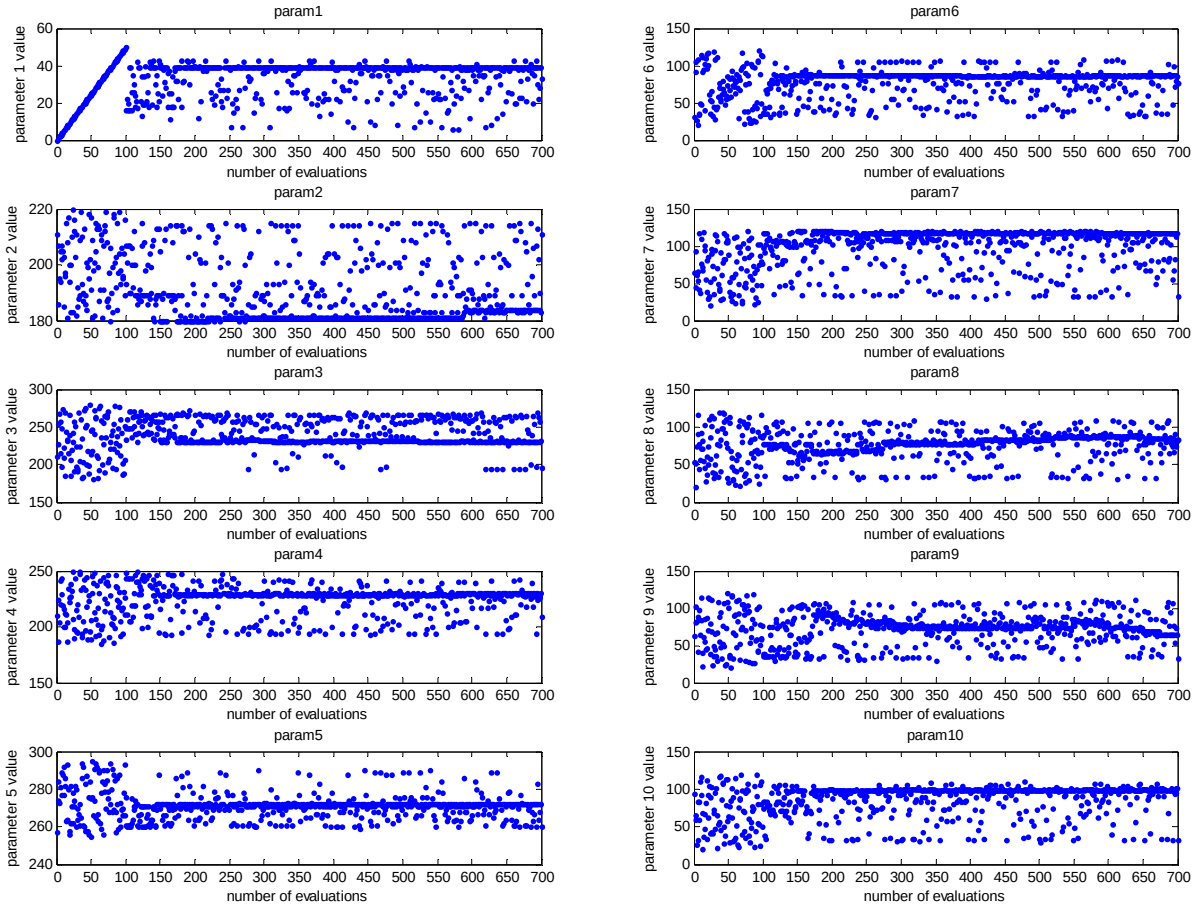


Figure 27: parameter values for all the basic parameters but with 600 newpoints

Appendix B

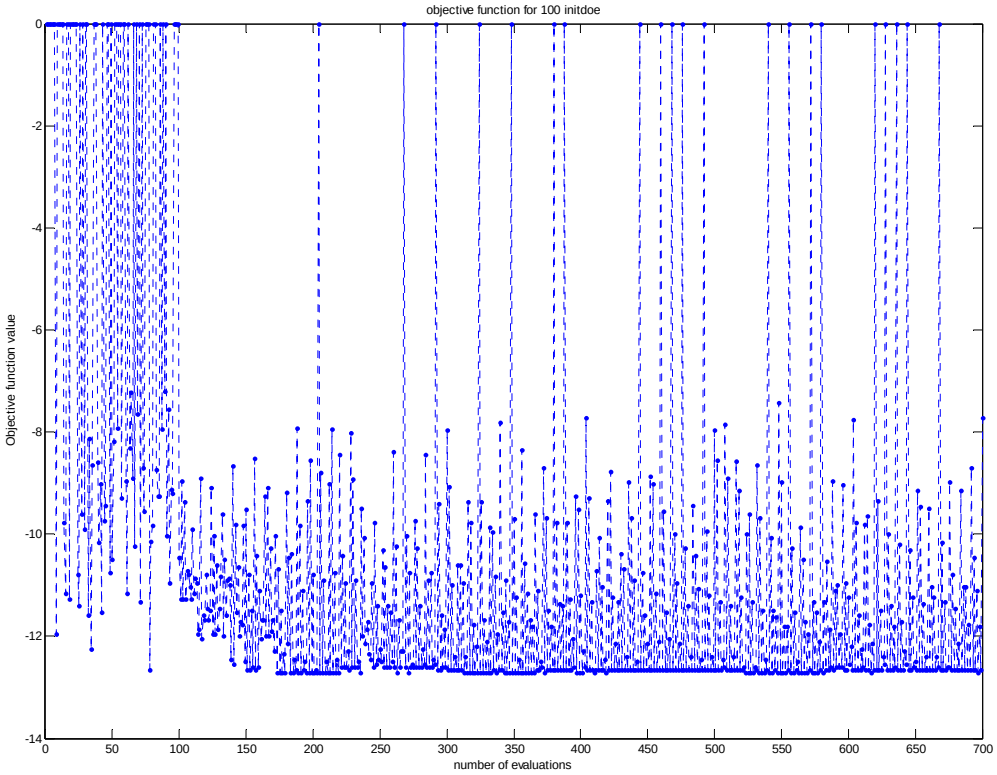


Figure 28: algorithm behaviour for all the basic parameters but with 600 newpoints