

Mat-2.4177 Seminar on Case Studies in Operations Research

**Designing Layouts for Specimens in Agricultural
Experiments Suffering from External Disturbances**

04.05.2008

Final Report

Client: Yara Suomi

Contact: TkT Jukka Sinisalo

Group:

Benjamin Hughes

Iiris Kolu

Jussi Lehtinen

Lauri Puro

Ville-Veikko Venojärvi (Project Leader)

Table of Contents

1.	Introduction.....	1
1.1.	The Client.....	1
1.2.	The Challenge	1
1.3.	The Objectives of the Project.....	2
1.4.	Structure of the report.....	3
1.5.	Terminology.....	4
2.	Literature review	5
2.1.	Introduction.....	5
2.2.	Reducing experimental error with plot design	6
2.3.	Theoretical methods for plot design	7
2.3.1.	Completely Randomized Design (CRD).....	7
2.3.2.	Randomized Block Design (RBD)	9
2.3.3.	Latin Square Design (LSD).....	12
2.3.4.	A Simple randomization procedure	14
2.4.	Conclusion	16
3.	Results	17
3.1.	Proposed Solution.....	17
3.1.1.	Ideology.....	17

3.1.2.	Implementation.....	17
3.1.3.	Definition of the criteria used.....	17
3.1.4.	Interpretation of the criterion used.....	21
3.2.	Alternative solutions.....	26
3.2.1.	Approach base on optimizing.....	26
3.2.2.	Approach based on block design.....	26
3.2.3.	Approach based on Latin Squares.....	27
3.2.4.	Approach based on distances between similar elements.....	28
4.	Discussion.....	30
4.1.	Analysis of Impact.....	30
4.2.	Success of the project.....	32
4.2.1.	Timetable.....	32
4.2.2.	Goals.....	33
4.2.3.	Quality of solution.....	34
4.3.	Future research.....	35
5.	References.....	36
	Appendices.....	37
	APPENDIX 1: WHAT WENT WELL VERSUS WHAT COULD HAVE GONE BETTER.....	37
	WHAT WE LEARNED.....	37

1. Introduction

This study is part of the course “Mat-2.1477 Seminar on Case Studies in Operations Research” that is lectured by Professor Ahti Salo from the Department of Mathematics and Systems Analysis in the Helsinki University of Technology. The study was concluded in Spring 2008. The project aimed to design a layout for agricultural experiments that suffer from disturbances. The project team consists of five undergraduate students of the Faculty of Information and Natural Sciences.

1.1. The Client

Yara Suomi is one of the leading fertilizer and fodder phosphate manufacturers in Europe. It operates in close connection with farmers, breeders, grocery- and fodder producers, chemistry industry and distributors. Yara Suomi pursues to develop partnerships that benefit all the parties involved. In its operations Yara Suomi is concentrated in creating advanced and tailored products for crop cultivation, animal feed industry and other industries.

Yara Suomi has 13 production facilities in eight countries as well as its own phosphate mine. The company’s products are sold in more than 100 countries and its net sales in 2006 were EUR 1,166 billion. Yara Suomi employs around 2,100 people worldwide. In 2007 Yara International ASA, the largest fertilizer producer in the world acquired Kemira GrowHow and so Yara Suomi was founded. Yara has over 8,200 employees and its revenues for 2007 were 7,266 billion euros.

1.2. The Challenge

One important research method for Yara Suomi is to conduct field experiments. These experiments are made to find out the effects, different kind of factors and their levels have on plant breeding. These experiments are conducted indoor as well as outdoor. Examples of different factors and their levels could be: whether the crop has ailment prevention or not and the level of fertilizer used.

Conducting these field experiments may involve disturbances that make the interpretation of the results more difficult or even impossible. The disturbances could be for example unequal sunlight for the plants, temperature alternations in the experiment or even wildlife. The problem these

disturbances make is that if a part of the experiment is ruined and it contains most of the specimens with a certain level of a factor, the analysis of the effect of this factor could become impossible.

Our task is to develop an algorithm or criteria with what better layouts for these field experiments can be made. Better layout here means that if something distracts the experiment, the layout is made so that the analysis of the experiment can be conducted in best possible way. The assumption here is that the possible areas that may encounter disturbances can be evaluated to some extent.

Our experiment planning algorithm should take the following aspects in consideration: number and the levels of factors in the experiment, the shape of the experiment area and other restrictions due to the arrangements of the experiment. It would also be useful if we could determine when the problems that disturbances cause can be avoided and when not, given certain experiment parameters.

1.3. *The Objectives of the Project*

Our research objectives have been formulated based on discussions with the representative of our client company as well as with our professor. Our main research objective can be stated as follows: This study aims to provide the theory and tools for designing more robust layouts for specimens in agricultural experiments.

In order to accomplish this main objective, we define the following sub-objectives for our study.

1. Literature survey. In order to better understand the restrictions and finer points of agricultural experiment design, we will conduct a review of the relevant literature. Although experiment design is an old subject and agriculture is a multi-billion dollar business, there are no existing models that could be directly applied to the problem in hand. However, studying the existing literature provides insight into possible solutions.

2. Identification of risks. Initially we planned to attempt to identify the different kinds of risk factors that can affect the quality of agricultural experiments for our client company. However, we

realized that apart from some well-known issues such as the fragility of the edges of the area, identifying the risks beforehand would not be practical, feasible or possible.

3. Mathematical model. We will describe the problem our client company is facing in mathematical terms by creating a model for the layout of their experiments.

4. MATLAB-program. Our main contribution was creating a piece of software for numerical calculation program MATLAB which will enable the user to automatically generate good layouts for specimens depending on variables the user enters. This represents a significant improvement from the current practice of layout design at Yara.

5. Quality. We define a measure for the quality of the layout of any given experiment. This will enable us to show the benefit of our solution compared to the current practice in our client company.

1.4. Structure of the report

The structure of the report is as follows. The report consists of four different sections. In section 2 we present the state of the discussion and the implications for our research as identified in our literature review. Section 3 first presents the proposed solution that we ended up with and the alternatives that we evaluated. It also presents in detail the MATLAB program that we created and give a brief description of the results. The last section compares the results with the current practice and analyzes the success our success in fulfilling our goals. The report ends with a short consideration of possible subjects for future research.

1.5. Terminology

Disturbances	Any type of external event or influence that has an unintended effect on the results of the experiment
Experiment	A test under controlled conditions that contains multiple specimens that are exposed to different levels of various factors. Designed to test the effect the factors have on their development
Factor	An intended influence on the experiment.
Layout	The plan according to which the specimens will be placed in the area designed for the experiment
Level	A discrete state of a single factor.

2. Literature review

2.1. Introduction

Agricultural field experiments have a significant historical role in the development of many modern scientific and statistical tools. Agricultural field experiments have been an important way of verifying the test results when people have crossbred plants or developed new cultivating methods in order to improve the overall harvest. Thus, agricultural field experiments are not a new research subject and there is plenty of literature available in the field. In addition, the problems related to agricultural experiments are quite common in other experiments too. Uneven environmental factors affect many kinds of scientific tests. Therefore much research has been conducted on experiment design. The innovator in the use of statistical methods in experimental design was Sir Ronald A. Fisher (1890-1962) who first used analysis of variance as a primary method of statistical analysis in experimental design (Montgomery, 1976).

Due to the historical nature of the problem most of the ground-breaking discoveries made in this area have been made decades ago and today some of them might seem quite self-explanatory. The applications developed in agriculture and biology have been afterwards applied to numerous research areas. Even though one could say that we take one step backwards from the latest research in our study, the case specific circumstances make this problem really interesting. The main difference between theoretical methods derived from the literature and our case is that usually every specimen is assumed to experience the same environmental effects or at least local changes in these effects are completely random. In our case we have some predetermined assumptions how the local disturbances might behave in the field experiment.

In many experiments, there are some case specific features which make it impossible to simply follow some literature based design method exactly. One has to apply the methods suggested by the literature as best as possible to the experiment under review. This applies to our problem too. Therefore it is really important to understand the different theoretical methods and their up- and downsides and then consider how well each one of them applies to this particular problem and

which ones to choose.

2.2. Reducing experimental error with plot design

Traditionally the main objective in agricultural field experiments has been the comparison made between the newly developed variety and the well-known standard variety. There are two issues to be considered:

1. Is the yield of the new variety superior to that of the old?
2. How large is the difference between the varieties?

As mentioned before researchers have developed many scientific and statistical methods in order to verify these questions. If the data was perfect this job would be very straightforward. But in real experiments there is always some experimental error that must be taken into account. Actually, one of the goals of modern experimental design is to provide a measure for experimental error. Another goal is to reduce experimental error as much as possible within the limitations of the resources that can be committed to the experiment. In this study we are particularly interested in how we can reduce the experimental error by different plot designs.

In his book “Agricultural Field Experiments” Roger Petersen lists few general considerations that should be considered when making the plot design. First of all, many studies with plots of different shape have indicated that long, narrow plots are to be preferred. Plots of this shape are usually more economical for such field operations as seeding, tillage, cultivation, and harvest. In general, if there is fertility or other gradient in the field, the plots should be oriented with the longest plot dimension in the direction of the greatest variability. This justifies our decision to focus on only rectangular plots. (Petersen, 1994)

The second important consideration concerns the borders of the plot. Plants along the edges of plots often perform differently than those in the center of the plot. Reasons for this are that outer boundaries have an advantage over interior plants as regards competition for air, sunlight nutrients and moisture. In addition when studying plants with different treatments there is certain spill-over that affects the plants on the borders. The machinery is usually not precise enough to keep the

treatment only on the specified area. Normal way of dealing with this is to exclude the borders from the study and look only at the plants in the center. This of course is usually applicable only in large scale outdoor studies. In our study we have give extra attention to assigning the specimens along the edges. (Petersen, 1994).

2.3. Theoretical methods for plot design

In this section we study some of the methods used when conducting agricultural field experiments. The aim of this review is to find support for our algorithm design. We are going to study closer the following experimental design methods: Completely randomized design, Randomized Block Design and Latin Square Design. The following results presented are mainly based on Petersen's book (Petersen, 1994).

2.3.1. Completely Randomized Design (CRD)

In CRD the treatments are assigned to the plots completely randomly and without any restrictions. This means that each plot is equally likely to be assigned to any of the treatments. The CRD has many advantages, disadvantages and uses considering experiment design.

Advantages

- *Flexibility.* Any number of treatments and replications of them may be used. The number of replications need not be the same for every treatment, although the best experimental results are obtained when treatments are equally replicated.
- *Simple statistical analysis.* Even with unequal replication of the treatments the analysis is not complicated. This is not the case when using more complex design methods.
- *Missing plots.* The analysis does not get more complicated if some of the plots are unusable.
- *Maximum error degrees of freedom (df).* This method provides the most error df than any other design method with equal number of treatments and replications.

Disadvantages

- Low precision if the plots are not uniform. This is often the case with our client, since different external factors affect the experiment so that the plots cannot be treated as they have uniform substrates.

CRD is a very useful method for conducting agricultural field experiments if the circumstances for it are appropriate. The experimental area should be relatively uniform and thus the effect of external factors to the experiment minimal. If a large portion of the plots in the experiment may be lost CRD is a viable option since analysis of the data remains relatively simple. If the number of plots is limited CRD offers the maximum amount of df for error and thus makes it a good choice for experiment design.

Design construction

In CRD there is no restriction on the assignment of treatments to plots. The only stipulation is that any treatment is equally likely to be assigned to any plot. Let us assume that the experiment is made in such way that each treatment is equally replicated. This is the case in most agricultural experiments. If the number of different treatments is p and each treatment is repeated k times the total of plots in the experiment is $p * k$. Now that we know the number of plots in this particular experiment we can randomize the treatments to each plot. One might use for example random number generators of Matlab to do this. Figure 1 presents one experiment design generated with CRD. In this example there, are 3 treatments and each treatment is replicated 4 times. Here the letter indicates the treatment applied to the plot.

1 A	2 B	3 C	4 B
5 A	6 A	7 C	8 C
9 B	10 C	11 B	12 A

Figure 1: Field plan generated with CRD

Use of CRD with our problem setting

The biggest drawback of CRD is the fact that it does not perform very well if the circumstances of the experiment are not uniform. At the moment, Yara Suomi is using a somewhat modified procedure of CRD. The problem with CRD is that the placement of the treatments is totally random. If we for example know that in the experiment setting described in the Figure 1 the upper part of the experiment area (plots 1-4) is very likely to be lost or affected by certain external factors. This would mean that when using CRD this fact would be neglected and the upside of the experiment could contain all plots treated with some treatment. If it for example contained all the samples with treatment A the whole experiment could be useless. When the circumstances of the experiment are uniform CRD is a good choice of experiment design, but if external disturbances to the experiment are expected in certain areas it becomes less efficient. Our proposed solution is based on a replicated CRD procedure.

2.3.2. Randomized Block Design (RBD)

The randomized block design is the design most often used when performing agricultural research.

It is almost as simple to use as the CRD, yet it provides an opportunity for increased accuracy. In RBD the plots are first classified into groups, or blocks, of plots that are as nearly alike as possible. The treatments are then assigned to these plots so that each block contains the equal amount of any particular treatment. Usually the treatments are replicated once within a block. The object of RBD is to minimize the variation within plots in the same block and to maximize the variation between blocks. This is why it is essential that the plots within a block are as alike as possible and they are treated as equally as possible (techniques, equipment etc).

Blocking Considerations

There are a wide number of criteria on which plots may be grouped into blocks. It is studied that in field experiments neighboring plots tend to perform more nearly alike than do plots that are separated from each other. If other source of variation can not be detected the blocks may be formed from a square and compact group of adjacent plots. One study suggests that grouping plots into blocks in this manner increases the efficiency of the experiment design by 5% compared to the CRD. Thus, if a clear source of variation outside the experiment design can not be detected CRD is almost as good a choice for experiment design than RBD. If fertility, sunlight or any other gradient exists in the experiment the blocks should be long and narrow, with the long axis perpendicular to the gradient. This principle is illustrated in the Figure 2. In some cases it could be a good choice to have the blocks being completely separated from each other. This is the case for example if the experiment is conducted outdoors and the effects of some physical features (roads, trees, bridges etc.) should be avoided. To succeed in performing in RBD it is essential that each block gets as equal treatment within the plots in it as possible. This mean that it is preferred that the machines, labor etc. used within a block should be the same during the entire experiment. Once the blocks are formed they should be treated as individual units. The plots within blocks and treatments assigned to them can be done randomly, like in CRD.

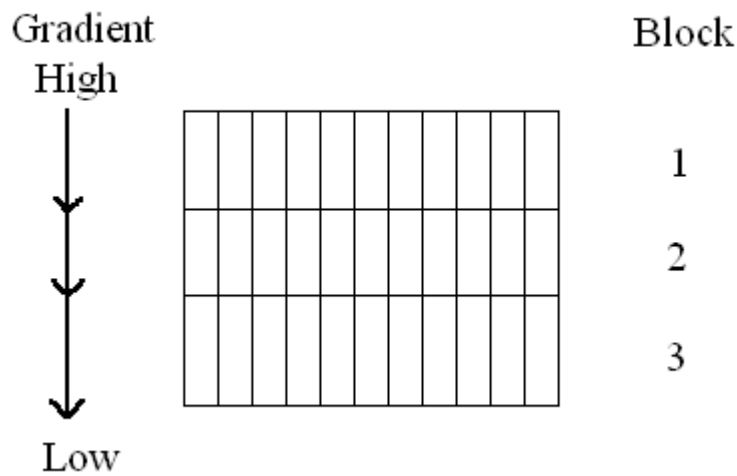


Figure 2: Placement of blocks with respect to gradients

Advantages

- RBD can remove one source of variation from experimental error and thus increase the precision of the experiment
- It can broaden the scope of the trial by placing blocks under different conditions
- Any number of treatments and blocks may be used. The only restriction is that each treatment is equally replicated within a block.
- Statistical analysis of an experiment designed with RBD is fairly simple.

Disadvantages

- Missing data can cause some difficulty in the analysis. One or two plots missing can be handled fairly easily, but several missing data can cause real problems.
- Assignment of treatments by mistake to plots in the wrong block can lead to problems in the analysis
- RBD is less efficient design method than the others if there are more than one source of unwanted variation
- If the plots are uniform CRD is more efficient than RBD

Uses of RBD

- It can be used to eliminate one source of variation (fertility of the soil, sunlight etc). Often in such situations The RBD provides satisfactory precision without more complex design.
- It provides unbiased estimates of the means of the blocking factors. Hence these means can be estimated using RBD.

Use of RBD with our problem setting

In our problem setting there are often sources of undesired variation such as unequal sunlight or differences within the soil. If it is known that there is primarily one source of variation and its effects to the experiment are known our algorithm could have used a modified RBD. The problem with RBD and our problem is that there can be several sources of unwanted variation. This makes the direct use of RBD in our solution impossible and hence we rejected the idea of implanting RBD directly in our solution proposal. However we decided to use some concepts of RBD in our final solution.

2.3.3. Latin Square Design (LSD)

Latin Square design is an experiment design method that is more restrictive than CRD or RBD. In LSD the number of plots in the experiment is the square of the number of treatments p . In this design the p^2 plots are first grouped into p groups of p plots, on the basis of one blocking factor (row grouping). The plots are then grouped into p groups of p plots (column grouping). Treatments are then assigned to the plots so that each row and column includes only one certain treatment. The basic idea of LSD is shown in figure 1.3, where the letters indicate the treatment applied to the plot. In this example $p = 4$. By blocking on two sources of variation LSD sometimes offers significant reductions in the experimental error compared to that of RBD.

A	B	C	D
B	C	D	A
C	D	A	B
D	A	B	C

Figure 3: Basic arrangement of a 4×4 Latin Square

Advantages

- It allows the experimenter to control two sources of variation

Disadvantages

- If there are only few treatment the error degree of freedom is small
- The experiment gets very large if the number of treatments increases (quadratic dependence on p). If $p = 10$ the number of plots used is 100.
- The statistical analysis gets complicated by missing plots and misassigned treatments.

Uses

- It is useful if two source of variance has to be controlled
- For practical reasons its use has usually been restricted to trials with $p \in [4,10]$

The Latin Square design can be used to remove the effect of two gradients. For example if fertility changes and the effect of sunlight are at right angles to each other we could use the experiments setting in Figure 3. Let us assume here that the Fertility increases when going from left to right in the experiment and the amount of sunlight increases when moving from up to the bottom of the experiment. In this kind of experiment environment Latin Square is a very viable option

for experiment design, since with it the effects of outside gradients (sunlight and fertility in this case) to the experiment can be significantly reduced.

2.3.4. A Simple randomization procedure

Let $p = 4$ for this experiment example. The randomization procedure proceeds the following way (here the alphabets represent different treatments):

1. First row in alphabetic order
2. Subsequent rows with letters shifted one position to the left of the one in the row immediately above
3. Randomize the order of the columns (for example 3, 1, 4, 2)
4. Randomize the order of the rows (for example 1, 4, 2, 3)

The procedure is presented in the Figure 4.

Phases 1, 2

A	B	C	D
B	C	D	A
C	D	A	B
D	A	B	C

Phase 3

C	A	D	B
D	B	A	C
A	C	B	D
B	D	C	A

Phase 4

C	A	D	B
B	D	C	A
D	B	A	C
A	C	B	D

Figure 4: Example of randomizing a Latin Square Design

Advanced Latin Square designs

Researchers have developed more advanced designs that utilize the principles of Latin Square design. One such method is Graeco-Latin Square Design. In this method one can combine two Latin Squares with each other. This design allows us to add another factor in the same study. Actually it is possible to combine multiple Latin Squares together and so add even more factors in the design. However, even the simple Graeco-Latin Square Design carries significant disadvantages. First, the design does not allow you to replicate any treatment combinations but each Graeco-Latin orthogonal pair is unique. Secondly, there are significant logical restrictions in the use of these methods. For example there just is not a Graeco-Latin Square available for a matrix of 6 x 6. For higher level designs these problems become even more severe. (Montgomery (1976), Cox (1958))

Use of Latin Squares with our problem setting

We can not use the Latin Square methodology directly in our algorithm design, since the nature of our problem is so diversified. There are some aspects we can consider when developing our criterion for more robust experiment designs. When for example we know that there are two outside factors disturbing the experiment then it might be a good idea to apply LSD, but we decided to make our solution such that it could perform well enough no matter what the number of the outside disturbance sources. Our most important lesson from this design method is concept of how to adjust if there are two outside factors affecting the experiment.

2.4. Conclusion

The literature review does not provide us a model that could be used as such to assign the specimens to a plot. However, we found a lot of interesting material on agricultural field experiments which we can utilize in solving the problem. First, we found some theoretical considerations about the plot shape and border effects. Secondly we found many methods which can be used as the basis for our algorithm. None of the methods fit perfectly to this case but combining the best features of each method, we can develop a relevant algorithm for assigning the specimens.

3. Results

3.1. Proposed Solution

3.1.1. Ideology

There would have been many different ways to try to find the solution for our problem and they are presented in section 3.2 Alternative solutions. After deliberation we decided to pursue the following methodology for our solution. Our approach was to use a brute force technique, which creates alternative, completely random and independent layout suggestions. For these suggestions we then calculate a criterion with what we can identify the most promising layout suggestion. To help us determine whether our criterion really is relevant we also decided to study the layout suggestions which our criterion suggested to be the worst.

3.1.2. Implementation

We implemented our solution with MATLAB. The process of creating a layout is following. The user must first decide the factors he/she wants to study. After that the number of levels of each factor has to be identified. Then the user has to decide how many replications of different combinations will be used. After this the user defines the shape of the experiment area (It was agreed with the client to be rectangular). Now all the data required has been defined and the user can create a layout for the experiment. Our implementation of this problem assumes that all the possible combinations of the factors and their levels will be used in the experiment. In reality the case always is not so but it was agreed with the client that we focus on the case where all the possible treatment combinations are studied. However the criterion we have developed is not depended on whether or not all the different combinations of the factors and their levels are used in the experiment or not.

3.1.3. Definition of the criteria used

The most important part of our study was to create a criterion which would find good enough layouts for the experiment. After discussions with our client, we decided to pursue a criterion which would generate more robust layout suggestions to outside disturbances than the client's present

methodology. We previously considered that the user could define the most likely areas to be disturbed by outside factors. Due to discussion with the client we decided to reject this idea, since the areas being affected by outside factors are not usually known. Our goal in this project has not been trying to find some kind of optimal layout design, we have settled to finding a good enough one.

We decided to use a penalty function to evaluate, whether or not a certain layout is a good candidate for the specific experiment. This penalty function studies all the different factors and their levels. It takes into consideration how close the similar levels of each factor are in the experiment, how many levels are in each factor and it also takes into consideration the location of different treatments in the experiment. It then calculates the criterion with what the layout can be rated. Forming of the criterion goes as follows:

m , the number of factors studied.

$C = [c_1 \ c_2 \ \dots \ c_{m-1} \ c_m]$, vector that shows the number of levels each factor has.

$A^{R \times C \times m}$, is a three-dimensional matrix that includes all the treatments of the specimens expressed as integers. It has R rows, C columns and the third dimension m represents the number of factors studied. So each factor has its own “layer” in the matrix. R and C are defined by the person conducting the experiment, they define the shape of the experiment area. It should be remembered here that the lower the value of the criterion is, the better the layout.

The value of the criterion Q is calculated with the following equation:

$$Q = \sum_{i=1}^m C(i) * \left(\sum_{j=1}^R \sum_{k=1}^C h_{jk} \right) + \left(\sum_{l=1}^m c_l \right) \left(\sum_{j=1}^R \sum_{k=1}^C d_{jk} \right), \quad (1)$$

where

$C(i)$ is the number of levels of factor i .

$\left(\sum_{l=1}^m c_i \right)$ represents the sum of all the factor levels.

h_{jk} represents the penalty calculated for every element of certain factor in A (here the matrix A can be treated as a 2-dimensional matrix since the calculations are performed only on one level of A at a time). h_{jk} is calculated as follows (pseudo code representation (2)-(6)):

$$h_{jk} = 0 \quad (2)$$

$$\text{if}(A(j,k) == A(j+1,k+1)) \Rightarrow h_{jk} = h_{jk} + 0,75 \quad (3)$$

$$\text{if}(A(j,k) == A(j-1,k+1)) \Rightarrow h_{jk} = h_{jk} + 0,75 \quad (4)$$

$$\text{for} : n = j+1 : j+C(i)-1 \quad (5)$$

$$\text{if} : A(j,k) = A(n,k)$$

$$\text{if} : k == 1 || k == C$$

$$h_{jk} = h_{jk} + \frac{1}{n-j} * d_1$$

$$\text{else} : h_{jk} = h_{jk} + \frac{1}{n-j}$$

$$\text{for} : l = k+1 : k+C(i)-1 \quad (6)$$

$$\text{if} : A(j,k) = A(j,l)$$

$$\text{if} : j == 1 || j == R$$

$$h_{jk} = h_{jk} + \frac{1}{n-j} * d_2$$

$$\text{else : } h_{jk} = h_{jk} + \frac{1}{n-j}$$

where

d_1, d_2 represent the weight put on the criterion if we are studying the borders of the experiment. Hence if the borders contain similar treatments close to each other it is penalized more than if they were in the middle of the experiment area. Naturally it has to hold that: $d_1, d_2 > 1$. In our proposed solution we set: $d_1, d_2 = 2,5$. We came to this value through simulations with various different values of d_1 and d_2 .

d_{jk} represents the closeness of every treatment combination. It is calculated similarly as h_{jk} except the input matrix is not a certain level of matrix A . The input here is a $B^{R \times C}$ matrix that contains all the different treatment combinations expressed as integers (the only thing that matters is that treatments can be separated from each other). Let $t = \prod_{i=1}^m c_i$ represent the number of all the different treatment combinations. Let B be the matrix that contains all the different treatment combinations. d_{jk} is calculated as follows (7)-(11):

$$d_{jk} = 0 \tag{7}$$

$$\text{if}(B(j, k) = B(j+1, k+1)) \Rightarrow d_{jk} = d_{jk} + 0,75 \tag{8}$$

$$\text{if}(B(j, k) = B(j-1, k+1)) \Rightarrow d_{jk} = d_{jk} + 0,75 \tag{9}$$

$$\text{for : } n = j+1 : j+t-1 \tag{10}$$

$$\text{if : } B(j, k) = B(n, k)$$

$$\text{if : } k = 1 \parallel k = C$$

$$d_{jk} = d_{jk} + \frac{1}{n-j} * d_1$$

$$\text{else : } d_{jk} = d_{jk} + \frac{1}{n-j}$$

$$\text{for : } l = k + 1 : k + C(i) - 1 \tag{11}$$

$$\text{if : } B(j, k) = B(j, l)$$

$$\text{if : } j = 1 \parallel j = R$$

$$d_{jk} = d_{jk} + \frac{1}{n-j} * d_2$$

$$\text{else : } d_{jk} = d_{jk} + \frac{1}{n-j}$$

Where


d_1, d_2 are defined similarly when calculating h_{jk} .

3.1.4. Interpretation of the criterion used

The following section tries to justify the choices made when developing the algorithm. The relevance of the criterion is also being vindicated.

Let us take a look at part (1) of the criterion and especially the meaning of $C(i)$. It can be seen that the more levels the factor has (the higher the $C(i)$) the more significant the factor becomes in defining the goodness of the layout. We decided to emphasize our criterion this way because the more a factor has different levels, the fewer replications of each level there are in the experiment. So it is justified that factors with many levels are more randomized than factors with less levels and more replications.

Here the justification of the parts (3) and (4) is discussed. Due to our literature survey and discussions with our client it became clear that if agricultural experiments are being influenced by outer factors they usually tend to have a local impact. Hence we decided to penalize the studied layout if there are replications of certain factor level lying diagonally. For example from Figure 5, it can be seen that in the best layout and its first factor (upper left corner subfigure) the specimens in places (2,1) & (1,2) have different levels of factor 1. Hence it is not penalized for this. But when you look at the worst possible layout in this example and that respective specimens you can see that specimens in (2, 1) & (1, 2) do indeed have the same levels of factor 1 and hence this layout is penalized for it.

The justification of parts (5) & (6) is being presented here. In these parts of the criterion it is studied that how close to each other different replications of factor levels are in the experiment. The study is conducted for every specimen and it is conducted horizontally and vertically. If there are n different levels of certain factor the specimens lying at the distance $n - 1$ or closer are studied. If there are specimens that contain the same level of certain factor than the specimen studied then the layout is being punished according to the distance between the studied specimen and the one with the same factor level. As mentioned earlier these studies are conducted vertically and horizontally. For example if you look at the best and worst layout suggestions and their first factor. It has 4 different levels (4 colors). Take a look at the specimen in place (1,1) from both layouts. Then the studied specimens are (1,2...n = 2...4) & (2...n =2...4,1). Now if you look at the best layout you can see that all the specimen studied horizontally have different levels of factor 1 than the specimen in (1,1) (this picture illustrates that ). Hence there will be no penalty assigned to this layout from this study. If you look at the vertical study you can see that the specimen in (1,1) has the same level of factor 1 than the specimen in (4,1). Hence the penalty from this study is 1/3. If you conduct the same calculations for the specimen in the worst layout and place (1,1) the penalty is going to be 1 horizontally and 1 vertically and 2 total. From the definition of (5) & (6) it can be seen that the closer the specimen with the same factor level is, the bigger the penalty will be which is only natural.

The reasoning with parts (7)-(11) of the criterion is similar with the previous one. It should be

noticed that here we study the whole layout instead of a certain factor layer. Even though it is most important that the different factors would be as random as possible, we sought to penalize the layout if the replications of different treatment combinations were too close to each other. It should be evident that if all the replications of a certain treatment combination were to be lost it could affect the success of the experiment drastically. We decided to emphasize this part of the criterion with the sum of all factor levels. This is done to balance the effects of using $C(i)$ as the coefficient in the other section of Q . The effect of this part of the criterion usually isn't as high as the effect of the part that studies the different factor levels. This results from the fact that there are more treatment combinations than factor levels if there is more than one factor.

The validation of our criterion is not very simple. The best way to see if a certain layout is a good one is to look at a graphic illustration of it. The Figure 5 represent a part of the solution the user gets when using our solution. In this example the shape of the experiment area is a square (6*6). The number of different factors is 2, the first factor has 4 and the second one 3 different levels. There are the 12 different treatment combinations and 3 replications of each. Our solution prints out the best and the worst solution according to our criterion. If you look at Figure 5, the first factor (upper row of pictures) is more randomized in the best layout suggestion than in the worst. The difference between these layout suggestions is even clearer in the case of the second factor (second row of pictures). Here the second factor is more randomized in the best layout suggestion. The figures best layout and worst layout show how close the different treatment replications are to each other. In this case, it can also be seen that the treatments in the worst layout are closer to each other (for example the right side of the layout) than in the best one. So the relevance of our criterion can be rather seen than calculated.

In our solution, we have used a benchmark which defines how many different solutions our algorithm generates at most. Running our test we have seen that in most cases 200000 layout suggestions is enough. Our algorithm does not necessarily create all these. It stops if the best solution has not changed for $0,1 * \text{benchmark} = 20000$ iterations. This speeds up our algorithm a little. In Figure 5 the lowest subfigure is the distribution of our criterion Q for different layout suggestions. When the user creates a layout suggestion for an experiment, our solution checks

whether the distribution of Q is normal (this seems to be the case if the number of iterations is high enough). If Q is normally distributed then a value is calculated which tells the probability of a better solution to occur (according to our criterion) if a random solution is selected instead of our solution. In the case represented in Figure 5 the probability of a better solution to occur if created randomly is 0,024%. This indicates that because our criterion can be considered to be relevant, our algorithm creates meaningful and good enough layout suggestion for agricultural experiments.

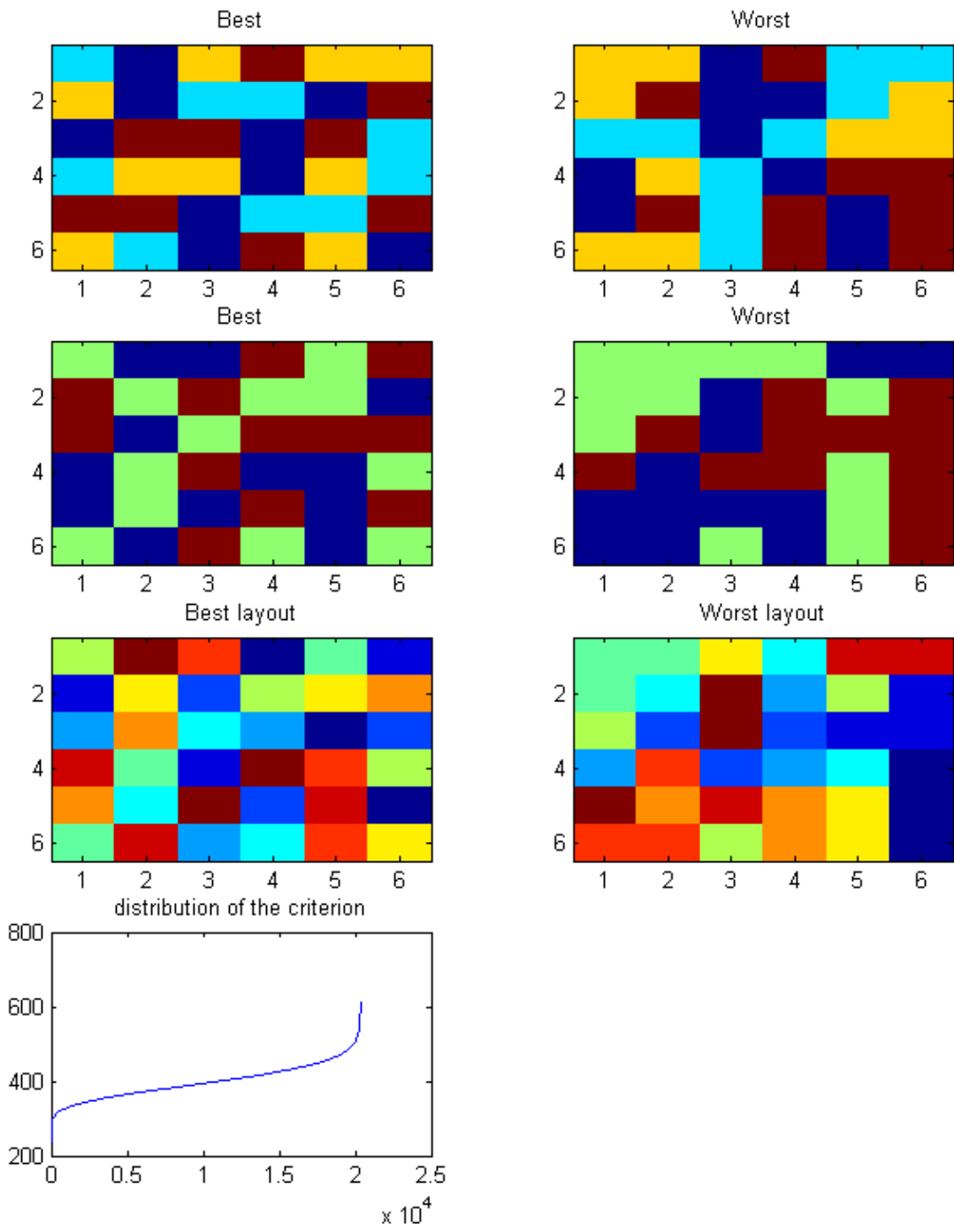


Figure 5: Distribution of the criteria.

3.2. *Alternative solutions*

During the process of developing our final algorithm, we thought about many other alternatives. Here we have analyzed some of these alternatives and why we chose the previously presented algorithm as the final one. Each of these alternatives has some features that are better and some that are worse than our final algorithm. However, when looking at the whole picture the final algorithm was considered the best.

3.2.1. Approach base on optimizing

Perhaps the most fundamental decision concerning the algorithm was that we did not try to find the optimal plot design but looked for a good enough one. The fundamental difference is that this decision enabled us to randomly generate plot designs and work with them instead of changing the matrix one element at the time and looking for the optimal design. We discussed this approach with the client and the guiding professor and we all agreed that in this problem there is no need to find the global optimum since a design that is good enough will do just fine. The alternative would have been to express the measure of goodness as a target function and let the algorithm change the matrix one element at the time when looking for the optimal value. The negative side of this approach would have been that it is very time consuming. When we add treatments, factors and replications the size of the plot area increases heavily which extends calculation time. At least with more complicated design the running the algorithm would have been painfully slow. This combined with the fact that there is little practical difference between good enough and optimal layout, we decided to abandon this. Perhaps if the design is kept simple enough with relatively low level of different factors this approach could be useful.

3.2.2. Approach based on block design

One alternative that we considered was an approach that based on the block design discussed in the literature review. In this approach the algorithm would have divided the plot area into to smaller plots. Then these blocks are mixed properly and combined together, and as a result the whole area would be mixed well enough. The advantage of this approach would be that we could easily divide the problem into sub-problems and so ease the calculation process. The drawback is that when we

combine the mixed blocks there might be some unwanted concentrations of specimens on the edges of the blocks. We tried to overcome this problem by choosing the blocking randomly and blocking the same matrix many times. However, soon we realized that if we do the blocking procedure too many times we might as well calculate the criteria for the whole plot in the first place. Again we decided to leave the blocking method as such out of the model since in the end it did not deliver the advantages we thought it could. However, the final algorithm does give the borders a special attention and this is an application of block design in some sense. The blocking method could prove to be really time efficient in cases where the plot size would be large and the number of treatments relatively low.

3.2.3. Approach based on Latin Squares

The two main problems with Latin Squares with our problems were that first Latin Square is shaped as a square by definition and the number of factors allowed is highly restricted (look for literature review for additional information). We wanted to enable the user to define the size and shape of the plot area quite freely. In addition number of replications, factors and factor levels should be chosen freely. Therefore the Latin Square does not work that well here. However, the same principle that every specimen should occur just once in one row and one column could be applied here. When the plot shape is changed or multiple factors included in the design (and experiment still done with full representation of specimen) it will be impossible to get exactly one specimen on corresponding rows and columns. But the idea could be applied so that each row and each column should include the expected number of each specimen. For example if you have 4x3 plot and you have 6 different specimens and two replications. Then for each row you should have $4/6 = 0.67$ of each specimen and on each column respectively $3/6 = 0.5$ specimens. By comparing the values of a certain plot design with these expected values, you can say how close to the optimal this particular design is in the spirit of Latin Squares. This approach has strong theoretical background and for small plots it works reasonably well. Of course, if you run the algorithm on a square plot (with correct amount of replications) the expected values are 1 and the optimal design is a Latin Square, which is usually considered quite optimal.

However, the negative aspect of this method is that it doesn't take into account how the specimens

lay on the row/column. This becomes a problem when the plot size is large and the expected value of specimens on a row is greater than one. For this algorithm, it is equally good if the two similar specimens on the same row are next to each other or if they are far away from each other. From our point of view, this is clearly not the same because the situation where the two similar specimens are next to each other is far worse than the other. The problem becomes even more severe when we start to look at the layout on a treatment level. We want to mix the layout in terms of different specimens but also in terms of all factors. If some factor has only two different treatments it is clear that there will be large number of similar treatments on each row and column. In such cases this approach will become blind to the differences between the layouts. However, this approach showed us that it has great potential. First of all, it is very simple and fast to calculate and thus the number of evaluated random designs could be increased substantially. In addition, it would seem to work quite nicely with smaller plots and on the upper level, where only different specimens are considered (enough different treatments). When the plot size increases and there is a need to look at the layout in terms of different factors this method proves to be too approximate.

3.2.4. Approach based on distances between similar elements

This approach is really close to the final method we chose to use. Here, instead of looking if a specimen has another similar specimen next to it, we would calculate its distance from other similar specimens. By doing this for all specimens we would acquire a measure of how far the similar specimens are from each other on average. By looking at this figure we could say if the layout is scattered enough. The positive side of this method is that it is quite intuitive and the goodness measure would have simple meaning: the average distance between two similar specimens. On the negative side this would be relative slow since one would have to calculate all the distances between matrix elements. By increasing the plot size and the number of measured layout the algorithm could prove to be too slow. However, the biggest problem with this approach is that as such it would not have enough emphasis on small distances. For example a “violation” of distance of only 1 could be neutralized to 3 by changing some other distance from 9 to 11. Surely from our point of view it doesn’t matter at all if the specimens are at a distance of 9 or 11, but a distance of 1 is unacceptable. As a solution to this problem the distances could be weighted so that close to zero would have large effect and for example higher than 5 wouldn’t matter at all. The weighting

process would of course add one more level of complexity in the model. Actually at this point we had the final idea for the model. Why do it so complicated if we are actually only interested if the nearby elements are not the same specimen. Thus, the final model is a kind of special version of this approach, where only the nearby elements are analyzed. However, this method could become handy if the possible disturbances would be really large and of all shapes (also long narrow disturbances possible). Then it could be wise to optimize the overall distances between similar specimens.

4. Discussion

In this section we elaborate on the benefits of our solution to our client as well as on our experience of working as team on a project of this scale.

4.1. *Analysis of Impact*

In order to examine the impact of our solution, we defined our own measure for the robustness of an experiment layout in the face of unexpected disturbances. As described earlier, this quality measure was a sum of penalties which a specific layout incurred if similar treatments were placed close to each other. Thus a low score would imply that a local disturbance will be less likely to affect similarly treated plants, whereas a high score indicates vulnerability.

An example of the benefits of our solution can be seen in Figure 6. There the task was to arrange 3 treatments with 2, 3 and 3 levels each onto an experiment area with 36 slots for specimens. Thus each unique specimen type was represented twice. The first six graphs represent the individual treatments in this scenario, with the top ones showing how the specimens were distributed in relation to the first treatment, the next two showing how they were distributed in relation to the second treatment and so on. The left hand figure always shows the best possible solution found through the simulation process, whereas the right hand figure shows the worst case scenario that was encountered during the simulation.

The plot at the bottom of the picture shows the distribution of quality encountered during the simulation run. In this case all the scores encountered are between 400 and 600, and our recommended solution is the layout with the very best score. The colorful pair of figures at the bottom of the picture shows the distribution of the specimens in relation to all treatments, i.e. how the unique specimens are distributed, again with the best one found being on the left and the worst one on the right. It is these figures that offer the most insight into the value of our solution, since at first glance both appear to be well scattered with no clear clusters of identical specimens. However, by looking at our measure of quality and by examining the individual treatment-pictures, it is

obvious that the layout on the right is far inferior. This is exactly the kind of situation where a poor, albeit initially robust-appearing layout may have been chosen. Thanks to our solution, the true robustness of the layout is revealed, thus enabling more informed decisions from our client.

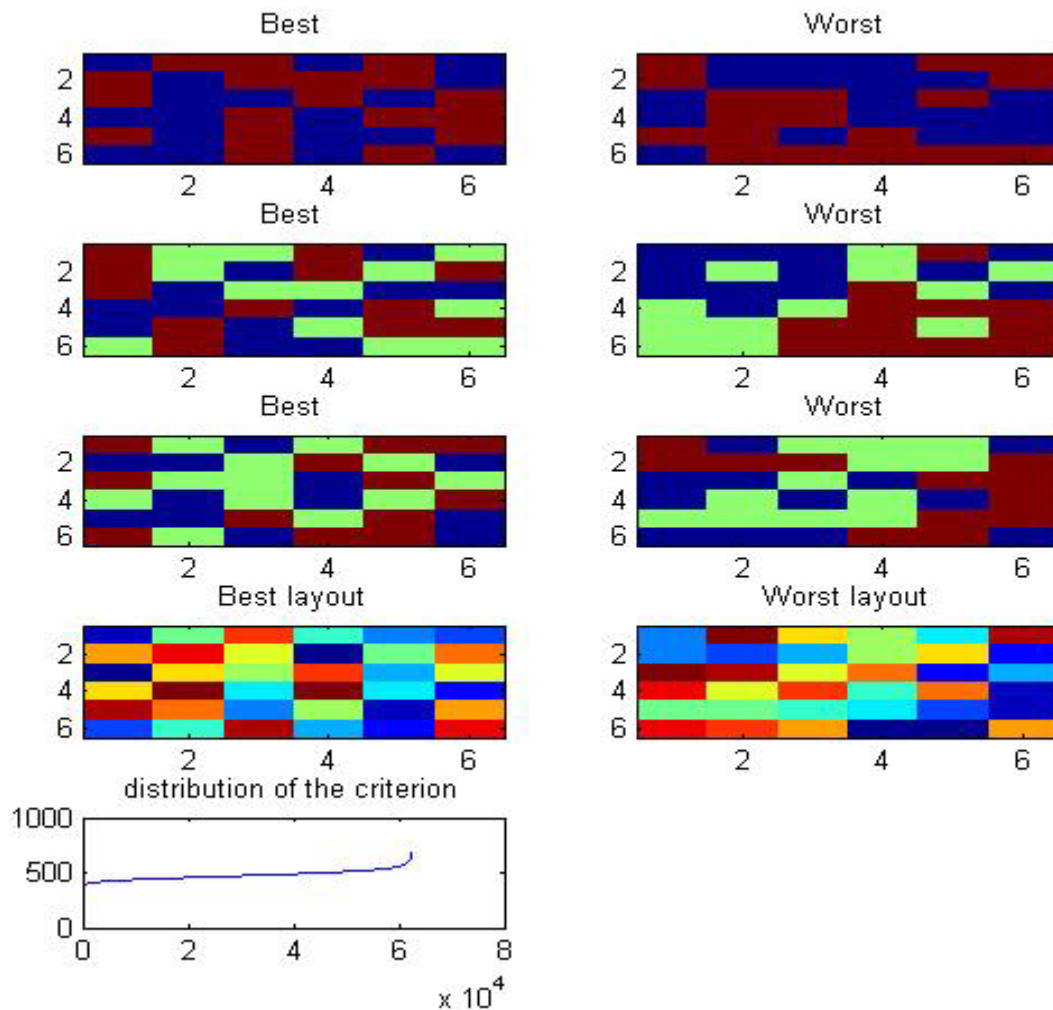


Figure 6: The results from our program.

With the new layout tool, our client’s experiments should be far more robust, meaning fewer experiments will need to be repeated due to the loss of similar specimens. This will not only save our client’s expenses by reducing the number of experiments, but also increase the speed with

which research can be studied and also the reliability of the results.

4.2. Success of the project

In general the project can be considered a success. Through teamwork we attained the objectives set out at the beginning of the project in a timely fashion, and our team met and worked together in a pleasant and productive atmosphere. Our project manager handled our communications toward our client and organized meetings making sure we had ample time to attain our objectives.

4.2.1. Timetable

At the outset of the project we set a timetable for ourselves, which can be seen in Table 1.

Table 1: Timetable

		Week	6	7	8	9	10	11	12	13	14	15	16	17	18
Planning	Initial meetings with stakeholders		■	■											
	Defining the problem			■	■										
	Initial Research				■	■	■								
	Making the project plan				■	■	■	■							
Research	Literature survey						■	■	■	■					
	Risk analysis						■	■	■	■	■				
	Modeling the problem						■	■	■	■					
Problem solving	Programming MATLAB							■	■	■	■	■			
	Testing the program										■	■	■		
	Formulating quality indicator											■	■	■	
Reporting	Project plan reporting					■									
	Mid-term reporting									■					
	Final reporting														■

Our team managed to stick to this schedule very well, as almost no changes were required, nor were we in a real hurry at any stage of the project. In fact the only change made was a positive one, as the

literature review was finished earlier than planned, enabling more focus on the MATLAB-implementation of the project.

4.2.2. Goals

In our original research plan, we defined certain objectives for our study. I will now examine how these objectives were accomplished.

1. Literature survey. In order to better understand the restrictions and finer points of agricultural experiment design, we will conduct a review of the relevant literature.

A thorough literature review was conducted, and although we did not directly use any of the methods from earlier studies, the review did provide us with an understanding of what the possibilities in this field were. The study will also provide a practical handbook for our client to use as a summary of existing literature.

2. Identification of risks. We will attempt to identify the different kinds of risk factors that can affect the quality of agricultural experiments for our client company.

Through discussions with our client we uncovered various threats to their experiments including uneven exposure to the sun, uneven irrigation and rampaging moose among others. Although these did not play a direct part in our solution, they did help our understanding of the problem.

3. Mathematical model. We will describe the problem our client company is facing in mathematical terms by creating a model for the layout of their experiments.

Our description of the problem was a function of the proximity of similar specimens, which is both easily understandable and functional.

4. MATLAB-program. We aim to create a piece of software for MATLAB which will enable the user to automatically generate good layouts for specimens depending on variables the user enters.

Our MATLAB-program is delivered ready-to-use, with several adjustable features and with enough flexibility for future improvements. The program produces clear visual tools for the placement of

the specimens and is fairly straight-forward to use.

5. Quality. We will define a measure for the quality of the layout of any given experiment. This will enable us to show the benefit of our solution compared to the current practice in our client company.

Through discussions within the project group we identified a relatively straight forward measure of quality for experiments. The measure can be fine-tuned further by adjusting multipliers for various penalties certain designs incur.

By fulfilling these sub objectives we were able to complete the main objective for this study, which was defined as:

To be able to provide the theory and tools for designing better layouts for specimens in agricultural experiments.

In our opinion this objective has been indisputably attained.

4.2.3. Quality of solution

In retrospect, certain aspects of the solution could be improved upon. For example, at the moment the measure of quality has no real-world interpretation, thus making it hard to say precisely how much better our solution is compared to the original one. However, the superiority of our solution is evident just by looking at the distributions of individual treatments from figure 6. Also, at the moment our MATLAB-program can only handle rectangular experimental areas. It is noteworthy however that most of the time the areas will be rectangular anyway, so the program can be used in most cases.

At the moment our solution is based on simulation, which is very unlikely to locate a global optimum solution. An analytic solution for finding the global optimum would thus be more effective, but in our opinion simulation provides substantial benefits already. Also, going through every single possible solution could very quickly become an arduous task for any computer, whereas our algorithm is relatively quick to process.

4.3. Future research

This study provides an interesting insight to the world of agricultural experiment design. However there are still many interesting topics for future research in this area. Below we have discussed some of them.

One topic for future research could be trying to solve the problem through optimization rather than using brute force randomization. Problems with this approach are the optimization process, which could become complicated and the calculation time, which could become too long. But it would be interesting to compare the global optimum (according to our criterion) of the problem with the results of our model and see if it significantly better.

One interesting field of study could be to let the user define the experiment area and freely. If the user could enter the area in terms of x,y –coordinates, the model could be applied to even more wider range of problems. However in this case the optimization criteria should be defined differently and it might be hard to find such a criterion that the results would match up to the method proposed in this study.

Another interesting topic could be to study how much the robustness of the results could be improved if the number of replicates would be increased. Or on the other hand how much the robustness suffers if one replicate would be excluded in order to save resources. In addition it would be interesting to study how changing the shape of the experiment area would affect the robustness of the layout.

5. References

Petersen, Roger G. 1994. Agricultural Field Experiments Design and Analysis. First edition. Marcel Dekker, Inc.

Cox, D. R. 1958 Planning of Experiments. First Edition. John Wiley & Sons, Inc.

Montgomery, Douglas C., 1976, Design and Analysis of Experiments, Third Edition, John Wiley & Sons, New York

Appendices

APPENDIX 1: WHAT WENT WELL VERSUS WHAT COULD HAVE GONE BETTER

In our opinion, this project was a schoolbook example of good planning and teamwork. We divided up the work whenever possible and then worked either in smaller groups or individually, but also met as a team on several occasions. We also maintained constant e-mail contact with each other, thus ensuring that everyone knew exactly how the project was progressing, and contacted our client a number of times to ensure the progress was within his expectations as well. The timetable of the project was strict enough to make sure our project would be completed on time, but also lax enough to allow for a leisurely work pace and for any unexpected changes. Taking all this into account, no real improvements come to mind regarding the actual process of our project.

It must also be said that the topic definitely turned out to be more interesting than it seemed at first. Many of our team-members were hoping for a more business-oriented project, but in the end our team was genuinely excited about this one! Several of us had previous experience in the way projects are designed and executed, but it was nevertheless inspiring to see again how results are made from nothing. Especially in this case it has been rewarding to see how all of us have begun to understand experiment design better although at the outset none of us knew anything about this field!

WHAT WE LEARNED

Most of what we learned during this project was very directly related to experimental design and MATLAB-programming, but certainly also to managing and scheduling projects successfully. However, learning MATLAB was definitely the most rewarding part of the project for most of us. Many times our team was struggling with a piece of code for hours on end in the school's computer labs, and the feeling of elation after the problem was solved was indescribable. I think most of us would not hesitate to call ourselves proficient MATLAB-users after this project.

APPENDIX 2: THE MATLAB-CODE

Below the MATLAB-code produced. The code is commented in Finnish.

Generate.m

```
6.     function [Q]=generate(a,b,c,d)
7.
8.     %a koealueen rivien lkm
9.     %b koealueen sarakkeiden lkm
10.    %c vektori johon syötetään eri faktorit sekä niiden tasot.
11.    %d kerranteiden lkm
12.
13.    % Lasketaan eri käsittelykombinaatioiden määrä
14.    tulo = 1;
15.    for i=1:length(c)
16.        tulo = tulo*c(i);
17.    end
18.
19.    %Lasketaan matriisit, jotka kuvaavat eri kombinaatioiden määrää, eli jos
20.    %tasoa saadaan matriisi M1 = [1 2 3]
21.    for n = 1:length(c)
22.        eval(['M' num2str(n) ' = linspace(1,c(n),c(n))']);
23.    end
24.
25.    %Talletetaan kaikki mahdolliset kombinaatiot matriisiin B
26.    %Käytetään valmista funktiota allcomb.
27.    if length(c) == 1
28.        B = allcomb(M1);
29.    elseif length(c) == 2
30.        B = allcomb(M1, M2);
31.    elseif length(c) == 3
32.        B = allcomb(M1, M2, M3);
33.    elseif length(c) == 4
34.        B = allcomb(M1, M2, M3, M4);
35.    elseif length(c) == 5
36.        B = allcomb(M1, M2, M3, M4, M5);
37.    elseif length(c) == 6
38.        B = allcomb(M1, M2, M3, M4, M5, M6);
39.    elseif length(c) == 7
40.        B = allcomb(M1, M2, M3, M4, M5, M6, M7);
41.    elseif length(c) == 8
42.        B = allcomb(M1, M2, M3, M4, M5, M6, M7, M8);
43.    else
44.        disp('laitoit liikaa käsittelyjä.')
45.    end
46.
```

```

47. %Sijoitetaan eri käsittelyt kolmiulotteiseen matriisiin A, jossa syvyys
48. %kuvaa eri käsittelyominaisuuksia. Aloitetaan vasemmasta yläkulmasta ja
49. %täytetään rivi kerrallaan. Käsittelyjä laitetaan matriisin B
    järjestyksessä
50. %niin monta kertaa kuin käyttäjä on määritellyt toistoja. Loppuihin
51. %mahdollisiin ylimääraisiin ruutuihin jätetään nollat. k ja r ovat
52. %apumuuttujia.
53.
54. k=1;
55. r=1;
56. A = zeros(a,b,length(c));
57. for i = 1:a
58.     if r > d*tulo
59.         break
60.     end
61.     for i2 =1:b
62.         if r > d*tulo
63.             break
64.         end
65.         for i3 = 1:length(c)
66.             A(i,i2,i3) = B(k,i3);
67.         end
68.         k = k+1;
69.         r=r+1;
70.         if k > tulo
71.             k=1;
72.         end
73.         if r > d*tulo
74.             break
75.         end
76.     end
77. end
78.
79. talteen = {}; %Ruudukko johon kerätään kaikki matriisit
80. talteen_kriteeri = []; %Vektori johon kerätään kaikkien matriisien
    hyvydet
81. laskuri = 0; %Laskee iteraatioiden määrän
82. max_laskuri = 200000; %Määritetään iteraatioiden maksimi
83. laskuri2=0; %Tarkastelee, kuinka paljon iteraatioita on
    kulunut viimeisen minimin saavuttamisesta
84. kerroin=0.1; %Määritetään, kuinka monta iteraatiota saa
    kulua ilman uuden minimin saavuttamista suhteessa iteraatioiden
    maksimimäärään
85.
86. while laskuri2<max_laskuri*kerroin && laskuri < max_laskuri
87.
88.     B=A(:, :, 1); %otetaan matriisin A ensimmäinen käsittely eli

```

```

    ensimmäinen taso matriisista A
89.         [R,I] = randmat(B); %sekoitetaan ensimmäinen käsittely
90.
91.         for j=2:length(c)   %sekoitetaan muut käsittelyt s.e käsittelyjen
    järjestys säilyy
92.             C=A(:, :, j);
93.             D=C;
94.             for i=1:a*b
95.                 C(i)=D(I(i));
96.             end
97.             R(:, :, j)=C;
98.         end
99.
100.        talteen{end+1} = R;
101.        %matriisissa R nyt satunnaistetut tasot
102.        %aletaan tutkimaan onko matriisi tarpeeksi satunnainen
103.        %Tutkitaan aluksi uloimmat rivit/sarakkeet
104.        l=[];   %Vektori, johon talletetaan eri taasojen kriteerit
105.        for g=1:length(c)
106.
107.            l(end+1)=optimize(R(:, :, g), a, b)*(c(g));   %Laskee viereisten
    solujen samankaltaisuuden asteen ja kertoo sen tason eri alkioiden määrällä.
    Potenssia voi muuttaa.
108.        end
109.        l(end+1)=optimize(kokomat(R, length(c), d), a, b)*sum(c);   %Laskee
    viereisten kombinaatioiden samankaltaisuuden asteen.
110.        talteen_kriteeri(end+1)=sum(l);
111.        % Jos kriteeri on parempi kuin edellinen minimi, niin laskuri2
112.        % nollautuu, muuten sitä kasvatetaan yhdellä.
113.        if talteen_kriteeri(end)>min(talteen_kriteeri)
114.            laskuri2=laskuri2+1;
115.        else
116.            laskuri2=0;
117.        end
118.        laskuri=laskuri+1;
119.
120.        end
121.        [C,O]=min(talteen_kriteeri);   %Hakee parhaan
    matriisin ja sen hyvyyden
122.        Q=talteen{O};
123.
124.        [C1,O1]=max(talteen_kriteeri);   %Hakee huonoimman
    matriisin ja sen hyvyyden
125.        Q1 = talteen{O1};
126.
127.        for i=1:length(c)
128.
129.            B=Q(:, :, i);   %parhaan matriisin i:s taso
130.            B1=Q1(:, :, i);   %huonoimman matriisin i:s taso
131.
132.            subplot(length(c)+2,2, 2*i-1), imagesc(B)   %paras
133.            title('Best');

```

```

134.     subplot(length(c)+2,2, 2*i),imagesc(B1)      %huonoin
135.     title('Worst');
136.     end
137.     subplot(length(c)+2,2,2*length(c)+3),plot(sort(talteen_kriteeri))
    %Plottaukset
138.     title('distribution of the criterion')
139.     E=kokomat(Q1,length(c),d);
140.     D=kokomat(Q,length(c),d);
141.     subplot(length(c)+2,2,2*length(c)+1),imagesc(D)
142.     title('Best layout');
143.     subplot(length(c)+2,2,2*length(c)+2),imagesc(E)
144.     title('Worst layout');
145.     [k1,s1]=normfit(talteen_kriteeri);
146.
147.     if jbstest(talteen_kriteeri,0.01)==1
    %Testaa, onko kriteeri normaalijakautunut
148.     Todennakoisyys_loytaa_parempi_matriisi_satunnaisella_yrityksella=normcdf
        (C,k1,s1)          %Kertoo, kuinka todennäköistä on löytää parempi
        iteraatio, saavutetun jakauman pohjalta.
149.     else
150.         disp('Tulokset eivät ole normaalijakautuneita');
151.     end
152.
153.     iteraatioiden_lkm=laskuri      %tulostetaan iteraatioiden lkm
154.
155.     a3=optimize(kokomat(Q,length(c),d),a,b)*sum(c);      %kuvaa sitä kuinka
        paljon rangaistaan käsittelyjen läheisyydestä parhaassa sekoituksessa.
        Kyseessä siis kokonaiskäsittelyt ei tietty faktoritaso.
156.     a4=optimize(kokomat(Q1,length(c),d),a,b)*sum(c);      %kuvaa sitä kuinka
        paljon rangaistaan käsittelyjen läheisyydestä huonoimmassa sekoituksessa.
        Kyseessä siis kokonaiskäsittelyt ei tietty faktoritaso.
157.
158.     whitebg('white');

```

Kokomat.m

%Piirtotiedosto, muuttaa matriisin eri tasot yksitasoiseksi matriisiksi,
%jonka alkiot koostuvat numeroista 1, 2, 3 jne. Tätä käytetään apuna
%plottauksessa, sekä hyvyyskriteerin laskennassa.

```

function C = kokomat(B, c,d)

C=B(:, :, 1);
for i = 2:c
    C = C+B(:, :, i)*10^(i-1);

```

```

end
C;

for j = 1:numel(C)/d
    y = max(max(C));
    for k = 1:numel(C)
        if C(k) == y
            C(k) = j*0.01;
        end
    end
end
end
C = C*100;

```

Optimize.m

%Tämä tiedosto sisältää hyvyyskriteerimme määrittelyn.

```
function t =optimize(A,r,c) %A matriisi, r on rivien lkm, c sarakkeiden lkm
```

```

u = 0; % hyvyyskriteerin arvo
%lasketaan tason eri alkioiden lkm
d = length(unique(A));

```

```

kerroin1 = 2.5; %kerroin, jolla reunaehtojen rikkomisesta sakotetaan.
kerroin2 = 2.5; %kerroin, jolla reunaehtojen rikkomisesta sakotetaan.

```

```

if min(min(A)) == 0 %tarkastetaan onko matriisissa tyhjiä kohtia. Jos on
muokataan d:tä.
    d = d-1;
end

```

```

for x = 1:1:r
    for y=1:1:c

```

```

        %sakotetaan jos kaakossa oleva sama eikä kyseessä ole tyhjät
        %ruudut.

```

```

        if x+1 <= r && y+1 <=c
            if A(x,y) == A(x+1,y+1) && A(x,y) ~= 0
                u = u+0.75;
            end
        end

```

```

        %sakotetaan jos koillisessa oleva sama. tyhjistä ei sakkoa.

```

```

        if x >1 && y +1 <= c
            if A(x,y) == A(x-1,y+1) && A(x,y) ~= 0
                u = u+0.75;
            end
        end

```

```

    end

```

```

%tutkitaan onko alkion alapuolella samoja
%alkioita max etäisyydellä d, jos on sakotetaan. Tarkastetaan myös
%millä kertoimella sakotus tapahtuu.
for i=x+1:(x+d-1)
    if i<=r
        if A(i,y) == A(x,y) && A(i,y) ~= 0
            er = i-x;
            if y==1 || y == c
                u = u+(1/(er))*kerroin1;
            else
                u = u+(1/(er));
            end
        end
    end
end
%tutkitaan onko alkion oikealla puolella samoja
%alkioita max etäisyydellä d, jos on sakotetaan.Tarkastetaan myös
%millä kertoimella sakotus tapahtuu.
for j=y+1:(y+d-1)
    if j<=c
        if A(x,j) == A(x,y) && A(x,j) ~= 0
            er = j-y;
            if x==1 || x==r
                u = u+(1/(er))*kerroin2;
            else
                u = u+(1/(er));
            end
        end
    end
end
end
end
t=u;

```