Optimising the arrangement of the storage yard of a brickworks

P.H.D. Lubbers

June 22, 2008

Abstract

This paper presents a method, using Microsoft Excel[1], to optimise the placement of a brickyard. Eventually, this will yield lower expenses by saving time, fuel and space. An algorithm was developed that takes all permutations of an entered schedule, consisting of a type of brick, an amount of pallets and a distribution point. All of these permutations are processed by another algorithm, which finds the distance from the specified distribution point to the nearest available location and stacks the pallets there, taking into account that different pallets have different dimensions and constraints. The distance to be driven will decrease if multiple pallets are stacked on the same location. All permutations are compared and the smallest total distance is chosen as the optimal solution. A major improvement on the placement was obtained on manually deciding where to stack the pallets. However, due to computer restrictions, it is not able to function as an optimising placement algorithm for larger schedules.

1 Introduction

A well-known problem is how to store as many products as possible, with the least effort, on a fixed place. In case forklift trucks are used, the positioning and fuel consumption are directly related. Hence, if the placement of products is improved, the fuel consumption will be reduced. This effect will increase with the size of the storage facility, making it worth to investigate and possibly improve the placement. Mathematics can be used to find a solution to this problem. Since the older computers nowadays are capable of calculating with enormous amounts of data, not only multinationals, but smaller organisations can benefit from this development as well. A brickworks named "Rodruza BV"[2] serves as the plant model for this thesis. Their placement of pallets with bricks on the brickyard is based solely on experience and this may not yield an optimal placing. The plant is mathematically modelled to allow improvement of the placement through the use of a computer. The novelty of this study is the use of Microsoft Excel to create a schedule for optimal placement with respect to fuel consumption. The company already works with this software and thus will not have any troubles in becoming acquainted with it. Microsoft Excel ships with *Visual Basic for Applications* (VBA) which is a programming language to automate scripts in applications like Microsoft Excel.

Much research has been conducted on the optimisation of transporting multiple products from different sources to locations. These are known as transportation problems, as first formulated by F.L. Hitchcock [3]. A well-known classical method to solve this type of problem is with the aid of linear programming[1]. An interesting article written on this topic[6] describes a model with two ports, where ships can load and discharge. This resembles the placement problem faced in this study. However, the article encloses more than this, as ship can sail empty and all ports are sources as well as destinations. Certain parts of this study can be used. Another approach to solve the transportation problem could be to model it as a min-cost flow problem [5]. Creating two sources will change the studied problem into the same kind of problem encountered in this thesis.

Overall, two well-known mathematical approaches[2] combined with aspects mentioned above, seem appropriate for optimising this storage problem. The cases studied[2] involve a company with multiple factories, which manufactures different products at different costs. The first method used to solve the placement problem is to deal with it as an assignment problem. The second approach handles it as if it is a minimum cost flow problem.

In this paper the following research question is investigated:

Is it possible to model the placement process of Rodruza BV to obtain an optimal placement?

The second Section of this study focuses on properties of the problem, how it was formed, which special constraints it has and which special cases of variables there are. Section three informs on different models that were constructed and tested, in addition to a description of the model that was finally used for the experiments.

[1] http://office.microsoft.com/en-us/excel/default.aspx
[2] http://www.rodruza.nl/

These experiments and their results are reported in Section four. Section five concludes the results of the experiments, while Section six presents a discussion and gives future recommendations.

2 Properties of the Problem

The model used in this study is based on Rodruza BV, specifically, the plant located in Gendt. This Section will describe which properties the model has; it is split into several parts: The first Subsection relates the real plant to the mathematical model. The second Subsection describes the constraints that the models must satisfy. Finally, some comments on the capacities of spots and distances to the spots will be elucidated in the next two Subsections.

2.1 Formulating the Problem

Rodruza BV uses two distribution points which place the bricks they produce on pallets; these are then taken to the allocated spot on the brickyard by fork lifters, this is illustrated in Figure B. In total, there are five different types of pallets used and there are multiple types of bricks that fit on them. Since there are over fifty types of bricks, with new ones developed periodically, each type is given a specific number. However, the pallets are named: (1) Waal Formaat (WF), (2) Euro Formaat (EF), (3) Hilversum's Formaat (HF), (4) Hulo Tang (HT) or (5) Hulo (HP). More details on these pallets can be found in Appendix A. Each distribution point has its own fork lifter to transport the pallets, hence there are two fork lifters. The brickyard consists of 346 spots which are marked areas on the brickyard. Appendix B contains an overview of the brickyard. Most of these spots have different depths and the width can be 2.75 or 2.30 meters. In the latter case, the spot will be indicated with a "P", meaning "Pallet" and is reserved for the first two types of pallets only (WF and EF). Three models were considered, implemented and revised to acquire an adequate model of the plants distribution area as will be discussed in Section 3. All models had to satisfy certain constraints to produce genuine results.

2.2 General Constraints

The models have to satisfy the following constraints to be adequate:

- The capacities should not be static, as will be described in Subsection 2.3.
- The spot capacities are restrictions for the model which may not be exceeded.
- There is a path from each distribution point to each spot.
- If the fork lifters place a pallet on a spot, the distance from the distribution point to the sport obviously shortens with the depth of the pallet. This requires the model to have a variable distance to the spot. Specifically: for each pallet a different distance.
- It has to be taken into account space between the pallets of different spots is necessary. There is an exception though: if two adjacent spots are stacked with the same pallets (WF or EF) and brick type, two spots will be able to contain five rows of pallets next to each other, instead of four rows which is normally the case.
- It should be easy to remove pallets from the brickyard in the model, since orders will be collected at daily basis.
- Different types of pallets and bricks will be produced during a week and two distribution points can simultaneously distribute different types, implying that the model should be able to process multiple jobs simultaneously.

A model that obeys all of these constraints will be usable for the research of this study.

2.3 Spot Capacities

The capacity of a spot depends on four properties:

1. Spot depth, width and pallet type determine how many pallets of that type can be placed on the spot. Because the spots have equal capacities for WF and EF pallets, there are only four different pallet sizes. Hence, each spot has four ground capacities. This means there is no static capacity for a spot unless it is in use. Thus, the models are required to be flexible enough to permit changing capacities.
2. Depending on the type of pallet, a number of pallets can be stacked on top of each other. Four WF or EF pallets, or three HF or HP pallets can be stacked. HT pallets are stacked as illustrated in Figure 1. The number of pallets that can be stacked on top of each other has to be multiplied with the ground capacity to get the total capacity of the spot.
3. If a spot contains a type of bricks, its current total capacity will be taken, and the number of pallets that are stacked on it are subtracted. This is the new spot capacity which is set until pallets are taken away from the spot or more pallets of the same type are added.
4. All spots will be checked to see if they contain pallets of the same type as the one being distributed. If a spot contains pallets of a different type, the capacity will temporarily be set to zero. In the case WF or EF pallets are distributed, a sub method will

be triggered. This method will attempt to find the first empty spot or the first spot containing WF or EF pallets. If the adjacent spot is empty or contains WF or EF pallets, the first mentioned spot will obtain a larger capacity. The capacities of the two spots are compared and the smallest is divided by two. When orders are collected the smallest spot will be the first to be empty again. If a different type of brick or pallets is stacked, there needs to be space between the spots again. This half capacity is summed with the capacity of the first mentioned spot found by the sub method. Algorithm 1 presents the pseudo-code that explains this procedure.

Algorithm 1 Calculate the Capacity of the Spots

```
for each spot on the brickyard do
  if spot is empty then
    if spot + 1 is empty or spot + 1 is same type as produced then
      Find smallest capacity of spot and spot + 1
      Add half of the found capacity to capacity of spot
    else
      capacity of spot stays the same
    end if
  else
    if spot is same type as produced then
      if next spot is empty or spot + 1 is same type as produced then
        Find smallest capacity of spot and spot + 1
        Add half of the found capacity to capacity of spot
      else
        capacity of spot stays the same
      end if
    else
      capacity of spot stays the same
    end if
  end if
end for
```

2.4 Distances

There are two types of distances used in this study: the *inter-distances* and the *intra-distances*.

The *inter-distance* is the shortest distance from a distribution point to the beginning of a spot. This is computed using the distances between junctions until the edge is reached where the designated spot is located. It is possible to calculate the remaining distance from the last junction to the spot from the widths of the spots. Half of the width of the road will be added since it is assumed that the fork lifters drive in the middle of the road. The distance table, containing the distance from each distribution point to each spot, was constructed using this design. This table is included in APPENDIX 3.

The *intra-distance* is the distance from the entrance of a spot to the place where the pallet will be stacked. This distance keeps changing when pallets are placed on the spot. The formula which describes this distance, assuming the pallets are stacked right on top of each other, will also be used to describe the intra-distance in a spot containing HT pallets, which are actually stacked as shown in Figure 1.

The general formula for the total distance travelled

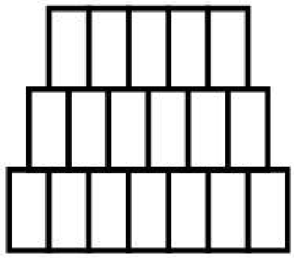


Figure 1: The formation of Hulotang pallets.

within a spot which will be explained in a moment will suffice to describe Figure 1. The fork lifters have to lift and bend forward to stack pallets higher than the ground. Resulting in the same distance as when they were stacked right on top of each other. Thus the same formula is used for all pallets. The only difference is the height they can be stacked. This formula is constructed as follows. It takes the total amount of pallets that are going to be stacked on the specific spot. If it is an empty spot, three pallets will be subtracted from the total. The distance covered to place these three pallets is:

$$d_{firstthree} = 3 \times sd - 2 \times pd \tag{1}$$

In this formula, d stands for the distance, sd stands for the depth of the spot and pd represents the depth of the pallet. The first pallet is placed to the back of the spot. The second and third pallets are placed on top of each other in front of it. The distance covered to place the rest of the pallets is calculated as follows:

$$d_{rest} = (sd - 2 \times pd) + (sd - 2 \times pd - (pd \times (\frac{x}{h-3} - 1))) \times \frac{x}{2} \tag{2}$$

In this formula, x and h are added, x stands for the total amount of pallets that is going to be stacked on the specific spot whereas the variable h corresponds to the height which pallets are allowed to be stacked. Formula (2) can be divided in three parts, separated by the first + and the last × in the formula. The first part of this equation is the distance to the furthermost pallet (excluding the first three). The second part is the distance to the nearest pallet. The last part is to average the distance found. Before calculating formula (2), the modular arithmetic (mod) of the entire amount of pallets is taken. If it is zero, only the formulas above are used. In the case that it is one, x will be reduced by one in all of these formulas and formula (3) will be added to d. If the modular arithmetic is two, x will be reduced by two in all formulas and formula (3) will be added twice.

$$d_{modx} = (sp - 2 \times pd - (pd \times (\frac{x - \pmod x}{h-3} - 1))) - pd) \quad (3)$$

The d in the three formulas summed yields the total distance travelled within a spot.

3 Modelling the Problem

This Section will describe different models that are examined to see whether they suffice to serve as the plant model. In all of the cases, a scale model will be used to give an impression and to find their strengths and weaknesses. This scale model will consist of two distribution points (sources) and five spots (sinks).

3.1 Transportation Model

The transportation problem is a linear programming problem. It involves determining how to optimally transport goods and allocate them.

A model has been constructed after the example of Hillier and Lieberman[2](page 327-335). The example used three sources/factories, four destinations/warehouses and a costs table. In the scale model, a dummy source will be added. This is necessary since in this approach all of the demand must be fulfilled. The model is illustrated in Figure 2, it consists of:

- A table with the distance between sources and destinations, including the dummy source which has a distance of zero to each destination
- A column that contains the supply for each distribution point, hence the produced pallets
- A row which contains the demand of each spot, thus the capacity of a spot
- A table which contains the number of transported pallets, the shipment quantity table

The shipment quantity table is multiplied with the distance table and then summed, to obtain the total distance travelled. The "Total Received" row and "Total Shipped" column are to check whether the set conditions are met.

Distance				Destination (Spot)					
		1	2	3	4	5			
Source	Ontl 1	2	4	6	12	14			
	Ontl 2	14	12	10	4	2			
	Dummy	0	0	0	0	0			
Shipment Quantity				Destination (Spot)					
		1	2	3	4	5	Total Shipped		**Supply**
Source	Ontl 1	4	1	0	0	0	5	=	5
	Ontl 2	0	0	1	2	2	5	=	5
	Dummy	0	3	3	0	0	6	=	6
	Total Received	4	4	4	2	2			
		=	=	=	=	=			**Total Cost**
	Demand	4	4	4	2	2			34 Meters Travelled

Figure 2: The scaled transportation model.

The drawback of this model is the use of the dummy source. In small cases, this will not have a strong influence because a single row in the results table consists of 346 variables. Adding an extra row would increase the amount of variables by a third, causing the model to be slower. Additionally, it would also increase the complexity to interpret the model and its results. When multiple types of pallets and bricks would be included in the model, two more tables would have to be included as well. This would increase the complexity of the model as well.

3.2 Minimum Cost Flow Model

This model has been constructed after the example of Hillier and Lieberman[2](page 396-403). It is a special case of a minimum cost flow model, namely another case of the transport problem. This time however, no dummy source is needed, which was the main reason why the previous model was not chosen for this study. The scaled minimum cost flow model consists of two supply nodes (sources), and five demand nodes (sinks). In this special case of the minimum cost flow model, there are no transshipment nodes present. Each source is directly connected to all sinks, giving rise to ten arcs.

This is represented in Microsoft Excel as shown in Figure 4. There are four columns of which the first one indicates the supply node. The second column is the spot, hence, the brickyard is stated twice. The third column contains the distances from the sources to the sinks. The last column contains the results, thus, how many pallets were sent to a spot. This column is multiplied with the distance and summed to get the total distance again.

From	To	Shipment Quantity		Capacity	Distance	Nodes	Net Flow		Supply/Demand
O1	S1	4	<=	4	2	O1	5	=	5
O1	S2	1		4	4	O2	5	=	5
O1	S3	0		4	6				
O1	S4	0		8	12				
O1	S5	0		2	14	**Total Cost**			
		0				34	Meters Travelled		
O2	S1	0	<=	4	14				
O2	S2	0		4	12				
O2	S3	1		4	10				
O2	S4	2		2	4				
O2	S5	2		2	2				

Figure 3: The scaled minimal cost flow model.

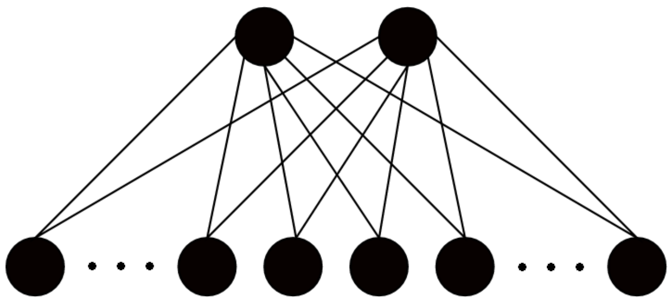


Figure 4: General structure of a flow model with two sources and multiple sinks, without any transhipment nodes.

The major drawback why this model (or the transportation model) could not be used for research was that Microsoft Excel was not capable of maintaining both brickyards at the right amount, a property of the minimum cost flow model was that each source had its own sinks. Microsoft Excel calculates everything at the same time, making for-loops impossible. Because the two distribution points could produce simultaneously, it was possible and quite likely that two different types of pallets or bricks were used in the model. The Premium Solver[3] is a solver that can find the optimal value for a variable with a specified target. It is possible to apply constraints to restrict values. For example, a variable can be minimized using the constraint that there are no negative variables. This solver is an extended version of the standard Microsoft Excel solver. It executes once, meaning it runs through the model until it finds a solution. Afterwards, VBA was able to adapt both arrangements in such a way that both versions of the brickyard would contain the same numbers again. This would work when the distribution points were not using the same spot. However, eventually both distribution points would want to use the same spot and spots would be occupied with multiple types of pallets or bricks. No solution or work-around to this problem was found, making this model unusable. Although a solution may exist, the spreadsheet would still depend on the Premium Solver add-in. This is not freeware, and an additional extension of the add-in was needed as there were too many constraints for the regular solver. Considering these drawbacks, it was not useful to continue research in this field.

3.3 Greedy Model

The final model is based on the inadequacies of the previous two models: it was necessary that the model would be able to process multiple productions at the same time and, if possible, known productions in the near future as well. These requirements were the foundation of the new model. Since VBA could not adapt multiple brickyards, only one brickyard would be used. Because none of the solvers (neither the standard Microsoft Excel Solver, nor the Premium Solver) could solve this problem, a very basic method in VBA was constructed that achieved the same as the solver: take the smallest distance from a distribution point to a spot with positive capacity, stack as much as possible on the spot and continue. Because this solver was entirely constructed in VBA, more extensions could be added, which could be executed during runtime. The VBA code can be divided in two algorithms: (1) the permutation algorithm and (2) the placing algorithm.

The permutation algorithm processes the input of the user. A user is prompted to enter up to eight productions. Each production consists of four parts:

1. the number of pallets that will be produced
2. the type of pallet that will be used
3. the type of bricks that are on the pallets
4. the distribution point where the pallets are transported from by the fork lifters.

The permutation algorithm then computes all permutations of the input and sorts the output underneath each other, as shown in Figure 5. This is performed brute force. Since Microsoft Excel can handle 1.048.576 rows, it is impossible to process more than eight inputs ($9! * 9 = 3.265.920$ while $8! * 8 = 322.560$).

The placement algorithm deals with one production and the brickyard. One production is one line of input. It will copy all of the information stored in the production. The amount of pallets that needs to be produced is copied to a special cell, specifically, the *checkcell*. This cell will subtract the amount that is stacked on a spot from itself until it reaches zero. Hence, it is a variable cell, which supervises the placement algorithm. It counts how many pallets still have to be transported and tells the placing algorithm when to stop.

Number	Pallet	Brick	Source
20	HT	105411	ZB1
40	EF	105380	ZB2
20	HT	105411	ZB1
40	EF	105380	ZB2
40	EF	105380	ZB2
20	HT	105411	ZB1

Figure 5: Example of input and calculated permutations.

[3] http://www.solver.com/xlspremsolv.htm

The placement algorithm starts by setting all capacities of the brickyard to the according type of pallet. It will do so through the use of the method described in Algorithm 1, section 3.3.
Afterwards, the placement algorithm will find the spot closest to the distribution point that has a non-zero capacity. This is done in the occupation part of the model as can be seen in Appendix C. If any pallets are transported to that spot, information describing the type of pallet and brick will be copied to the spot. The placement algorithm will compare the free capacity of this spot and the amount in the checkcell. Two cases arise: (1) the amount in the checkcell is larger than the capacity; the capacity will be reduced to zero and subtracted from the amount in the checkcell. (2) The amount in the checkcell is smaller than the capacity; the amount in the checkcell will be stacked at the spot, subtracted from the capacity and the checkcell will tell the placing algorithm to stop. All distances will be multiplied with the amounts and summed to get the total distance for that production, thus, a part of the entire total distance of all input.
The placing algorithm will subsequently take the next production until finished. All total distances will be summed to get the entire total distance. This number is stored with the number of the permutation. The placing algorithm will start over with the next permutation until they are all calculated, and will finally compare the total distance of each permutation, and once again choose the smallest distance informing the user which distance is the optimal placement as found. This is shown in Figure 6. A list will be provided that states all the changes on the brickyard according to the found placing, as illustrated in Figure 7.

Perm.		Total Distance (km)	
1	6,066762	6,07	
2	6,134262		

Figure 6: Two permutations with their distances.

Tas	Amount	Pallet	Type	Source	Distance
249	20	HT	105411	ZB1	3864,464
256	40	EF	105380	ZB2	2269,798

Figure 7: The produced list of placement.

4 Experiments and Results

Two types of experiments are conducted in this Section. The first type shows how different aspects of the model influence the results. The second type tests what happens when the model is used to test a real case.

4.1 Different Aspects

Three aspects of the model are tested: (1) the effect of the size of the production schedule on the running time, (2) the effect of permutations on the total distance and (3) the effect of splitting schedules, larger than the maximum of the model into different groups.

Running time

When multiple inputs are entered, all available permutations are calculated and tried. It is possible to enter up to eight different inputs, however, the number of steps needed to calculate these will be $O(n^n)$, which is almost the worst running time possible for large n. Hence, this will yield running times for the algorithm, which might be worth reckoning with. Table 1 shows the running time as measured on a PC, containing an "Intel Core 2 Duo @ 3.0 GHZ" processor with 2 GB of RAM. This is only meant to give an indication of the running time. No further results are derived from it.

Inputs	Running Time (seconds)
1	<1
2	<1
3	<1
4	4
5	24
6	175
7	1690
8	591+ (Out of RAM memory)

Table 1: Running times for different numbers of productions.

Permutations

Because of the permutations, different orders of placing are used. To test whether these permutations are useful, the total distance per permutation is printed. To maximize the difference per permutation, without maximizing manually on purpose, seven real productions were used as input. Figure 8 shows 5040 permutations. In the left part, they are split into seven groups. The height represents the number of permutations that belong to a particular group. The right part shows the total distance per permutation, after the permutations have been arranged in ascending order.

Split Larger Schedules

It may happen that more than eight productions are known. Since the model can only handle eight productions at most, larger schedules will have to be split. This can happen in various ways:

- The schedule is split in such a way, that each production is a group (Thus, one member per group).

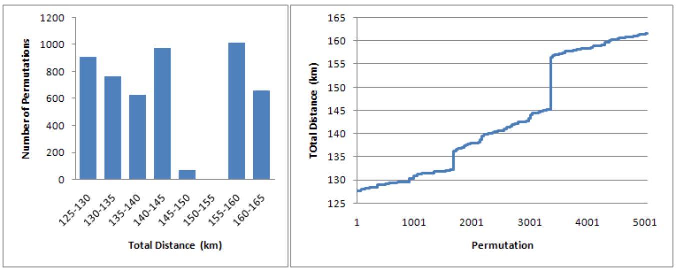


Figure 8: Graphical representation of total distance per permutation.

- The schedule is split into a minimum amount of equal groups, in such a way that no group consists of more than eight members.
- The schedule can be split into a minimum amount of groups, putting eight productions in the first group, eight in the second, until all productions are a member of a group.

To decide which production is a member of which group in these three ways, two methods are used. The first method sorts the productions according to production date, whereas the second method sorts according to production amount. This yields six different results for the total distance. For this experiment, a schedule containing nine productions will be used. These nine productions will be used in the next Subsection as well, as they are a practical example. Because a previous experiment has shown that the computer, used for the experiments, is incapable of handling eight productions at once, seven productions will be put into a group at most. The results of this experiment are provided in Tabel 2.

4.2 Real Plant Schedule

Rodruza BV contributed the data needed for this experiment. An overview of the brickyard at a certain moment was provided, in addition to every change after that moment, which can be picked up orders and relocated pallets. These changes were processed to get an updated overview of the brickyard, which could be used by the model. A transport list of the pallets that were produced was provided as well. Using the same formula to calculate the distance, as the model did, Table 3 was constructed.

The model has to place nine productions according to the schedule, which means that the schedule will be split into two groups. Due to the results of the previous experiment, it appears that it is best to create two groups, making the first one as large as possible with the greatest amounts in it. Thus, the model was executed twice. The schedule is shown in Table 4. The double line marks the position where the two groups were separated.

Spot	Num	Pallet	Brick	ZB	Distance
123	98	HT	105380	ZB1	9495,09
124	330	HT	105380	ZB1	24670,24
128	654	HT	105445	ZB1	27253,96
132	468	HT	105445	ZB1	24083,83
133	462	HT	105445	ZB1	25241,80
134	432	HT	105445	ZB1	25679,65
154	48	HT	105411	ZB1	7143,43
341	24	WF	111982	ZB1	4218,06
348	112	WF	111982	ZB1	23501,99
443	20	WF	111982	ZB1	5474,68
67	26	WF	111989	ZB2	2320,84
123	168	HT	105380	ZB2	27778,96
124	172	HT	105380	ZB2	28782,23
340	378	WF	111982	ZB2	12553,60
341	122	WF	111982	ZB2	580,14
348	116	WF	111982	ZB2	4899,63
354	136	HT	115330	ZB2	8906,61
361	200	WF	111330	ZB2	12518,59
Total distance (m):					**275103,33**

Table 3: The list of how the pallets were placed, manually. The line separates the sources. Num is an abbreviation of number.

After the model was executed twice, Tabel 5 could be constructed, which has a similar structure as Tabel 3.

Finally, Table 6 summarizes the results of the experiments concerning the total distances. The total distance of the manual placement is taken as 100%.

5 Conclusions

The purpose of this thesis was to optimise the placement of pallets which was decided manually until now. A new model was developed for this purpose, which is based on fundamental principles to minimize the total distance travelled by the fork lifters. Data provided by Rodruza BV showed that the distance travelled in reality measured 275,10 km. A couple of conclusions can be drawn

Amount	Pallet	Brick	Source
2016	HT	105445	ZB1
616	WF	111982	ZB2
428	HT	105380	ZB1
340	HT	105380	ZB2
200	WF	111330	ZB2
156	WF	111982	ZB1
136	HT	115330	ZB2
48	HT	105411	ZB1
26	WF	111989	ZB2

Table 4: The production schedule of three weeks, divided in two batches.

9 groups, arranged to date

Amount	Pallet	Brick	Source	Distance
48	HT	105411	ZB1	2,03
428	HT	105380	ZB1	13,91
2016	HT	105445	ZB1	106,67
200	WF	111330	ZB2	5,20
616	WF	111982	ZB2	22,55
26	WF	111989	ZB2	1,32
136	HT	115330	ZB2	6,64
156	WF	111982	ZB1	12,97
340	HT	105380	ZB2	23,81
Total distance travelled (km):				**195,09**

9 groups, arranged to descending amount

Amount	Pallet	Brick	Source	Distance
2016	HT	105445	ZB1	73,56
616	WF	111982	ZB2	18,33
428	HT	105380	ZB1	17,48
340	HT	105380	ZB2	15,83
200	WF	111330	ZB2	10,58
156	WF	111982	ZB1	8,77
136	HT	115330	ZB2	6,99
48	HT	105411	ZB1	2,24
26	WF	111989	ZB2	1,19
Total distance travelled (km):				**154,98**

2 groups, arranged to date

Amount	Pallet	Brick	Source	Distance
48	HT	105411	ZB1	98,48
428	HT	105380	ZB1	
2016	HT	105445	ZB1	
200	WF	111330	ZB2	
616	WF	111982	ZB2	55,62
26	WF	111989	ZB2	
136	HT	115330	ZB2	
156	WF	111982	ZB1	
340	HT	105380	ZB2	
Total distance travelled (km):				**154,10**

2 groups, arranged to descending amount

Amount	Pallet	Brick	Source	Distance
2016	HT	105445	ZB1	124,89
616	WF	111982	ZB2	
428	HT	105380	ZB1	
340	HT	105380	ZB2	
200	WF	111330	ZB2	29,77
156	WF	111982	ZB1	
136	HT	115330	ZB2	
48	HT	105411	ZB1	
26	WF	111989	ZB2	
Total distance travelled (km):				**154,66**

2 groups, arranged to date

Amount	Pallet	Brick	Source	Distance
48	HT	105411	ZB1	127,73
428	HT	105380	ZB1	
2016	HT	105445	ZB1	
200	WF	111330	ZB2	
616	WF	111982	ZB2	
26	WF	111989	ZB2	
136	HT	115330	ZB2	
156	WF	111982	ZB1	32,59
340	HT	105380	ZB2	
Total distance travelled (km):				**160,32**

2 groups, arranged to descending mount

Amount	Pallet	Brick	Source	Distance
2016	HT	105445	ZB1	149,59
616	WF	111982	ZB2	
428	HT	105380	ZB1	
340	HT	105380	ZB2	
200	WF	111330	ZB2	
156	WF	111982	ZB1	
136	HT	115330	ZB2	
48	HT	105411	ZB1	3,43
26	WF	111989	ZB2	
Total distance travelled (km):				**153,02**

Table 2: Six different approaches to split more than seven productions in a schedule. The horizontal lines above the distances separate different groups.

Spot	Num	Pallet	Brick	ZB	Distance
28	129	HT	105445	ZB1	6558,76
33	129	HT	105445	ZB1	6558,76
38	129	HT	105380	ZB1	8332,51
40	52	HT	105380	ZB1	3911,14
117	93	HT	105445	ZB1	4428,15
119	156	WF	111982	ZB1	6926,42
120	48	HT	105411	ZB1	2240,20
123	255	HT	105445	ZB1	11461,65
124	282	HT	105445	ZB1	11392,26
128	282	HT	105445	ZB1	8290,26
132	282	HT	105445	ZB1	7514,76
133	282	HT	105445	ZB1	8290,26
134	282	HT	105445	ZB1	9065,76
135	101	HT	105380	ZB1	2345,81
136	50	HT	105380	ZB1	1185,78
137	48	HT	105380	ZB1	1268,20
140	48	HT	105380	ZB1	1664,20
236	38	WF	111982	ZB2	1480,93
256	176	WF	111982	ZB2	8859,77
340	222	HT	105380	ZB2	7302,87
344	136	WF	111330	ZB2	4917,74
351	208	WF	111982	ZB2	8826,47
356	88	WF	111982	ZB2	2955,42
357	26	WF	111989	ZB2	1187,21
360	52	WF	111982	ZB2	2466,18
430	63	HT	105380	ZB2	1116,21
443	55	HT	105380	ZB2	1962,31
444	64	WF	111330	ZB2	2263,17
459	16	WF	111982	ZB2	461,15
516	11	WF	111982	ZB2	568,66
524	27	WF	111982	ZB2	1109,50
557	16	HT	115330	ZB2	758,14
558	120	HT	115330	ZB2	5343,45
Total distance (m):					**153014,06**

Table 5: The list of how the pallets could have been placed, automatically. The line separates the sources. *Num* is the abbriviation of number.

Experiment	Total distance	Percent
Manual placement	275,10	100,00%
9 groups, date	195,09	70,92%
9 groups, amount	154,98	56,34%
2 groups (4/5), date	154,10	56,02%
2 groups (4/5), amount	154,66	56,22%
2 groups (7/2), date	160,32	58,28%
2 groups (7/2), amount	153,02	55,62%

Table 6: Results of all experiments concerning the total distance, including percent of manual placement.

from Table 6: (1) the choice on where to split a schedule is crucial for the outcome of the model. The amount of productions which can be processed at once should be as high as possible and contain the productions with the highest amounts of pallets. (2) The model is capable of stacking the same amount of pallets, while only travelling 55,62% of the distance. This is quite an improvement on the old manual placement. A direct consequence is that the fuel consumption can be reduced by almost 45%. This applies to the particular case studied. Hence, different schedules will lead to different results.
Another goal of this thesis was to be able to provide the fork lifters with a schedule, describing where to place the pallets. As Table 5 shows, it is possible to create this list. After ten minutes the computer ran out of RAM memory to calculate all the permutations of eight productions. The running time of seven productions however, involved more than 28 minutes of calculating. However, this means that it took less than ten minutes to calculate all the permutations, and thus more than 18 minutes to process all possible permutations. The bottleneck of the model is therefore the placing algorithm as explained in Section 4.3.

6 Discussion and Future Recommendations

The model can be improved on various points. The easiest is making the permutations algorithm recursive. This will increase the efficiency of the model greatly. Due to minor experience in VBA programming, it was not possible do this in the time given for this thesis. Clearly, this affects the running time in a negative way for multiple input. It would instantly increase the amount of productions that can be handled as well which has a great influence on the amount of total distance. Since all possibilities are examined if all productions can be processed in one schedule, the optimal one with respect to fuel consumption is bound to be one of them. A faster way to get the results could be by using heuristics[4]. A second improvement would be to cut certain sets of permutations. Figure 8 shows that there are gaps between the total distances. The model itself can show the solution to each permutation and it is clear that some of them are not worth examining. These could be cut using another algorithm. All the constraints described in Section 3.2 were fulfilled and the model is Microsoft Excel based, making this model useable.

References

[1] A. Charnes and W.W. Cooper (1954). The stepping stone method of explaining linear programming calculations in transportation problems. *Management Science*, Vol. 1, pp. 49–69.

[2] Hillier and Lieberman (2007). *Introduction to Operations Research*. McGraw-Hill.

[3] Hitchcock, F. (1941). The distribution of a product from several sources to numerous localities. *Journal of Mathematics and Physics*, Vol. 20, pp. 224–230.

[4] Klincewicz, John G. (1990). Solving a freight transport problem using facility location techniques. *Operations Research*, Vol. 38, pp. 99–109.

[5] Skutella, Martin (2002). Approximating the single source unsplittable min-cost flow problem. *Mathematical Programming*, Vol. 91, pp. 493–514.

[6] T. Koopmans and ed (1951). Activity analysis of production and allocation. *Cowles Commission for Research in Economics*, Vol. 13, pp. 222–259.

A Details on the Pallets

The next table gives some specifics on the different pallets.

Type of pallet	Depth (m)	Width (m)
Hulotang (HT)	0,4	2,3
HuloPallet (HP)	0,4	1,15
WF Pallet (WF)	1,12	0,84
EF Pallet (EF)	1,12	0,96
HF Pallet (HF)	0,47	1,15

Table 7: Details on the pallets.

B Overview of the Brickyard

This Figure gives an overview of the arrangement of spots on the brickfield.

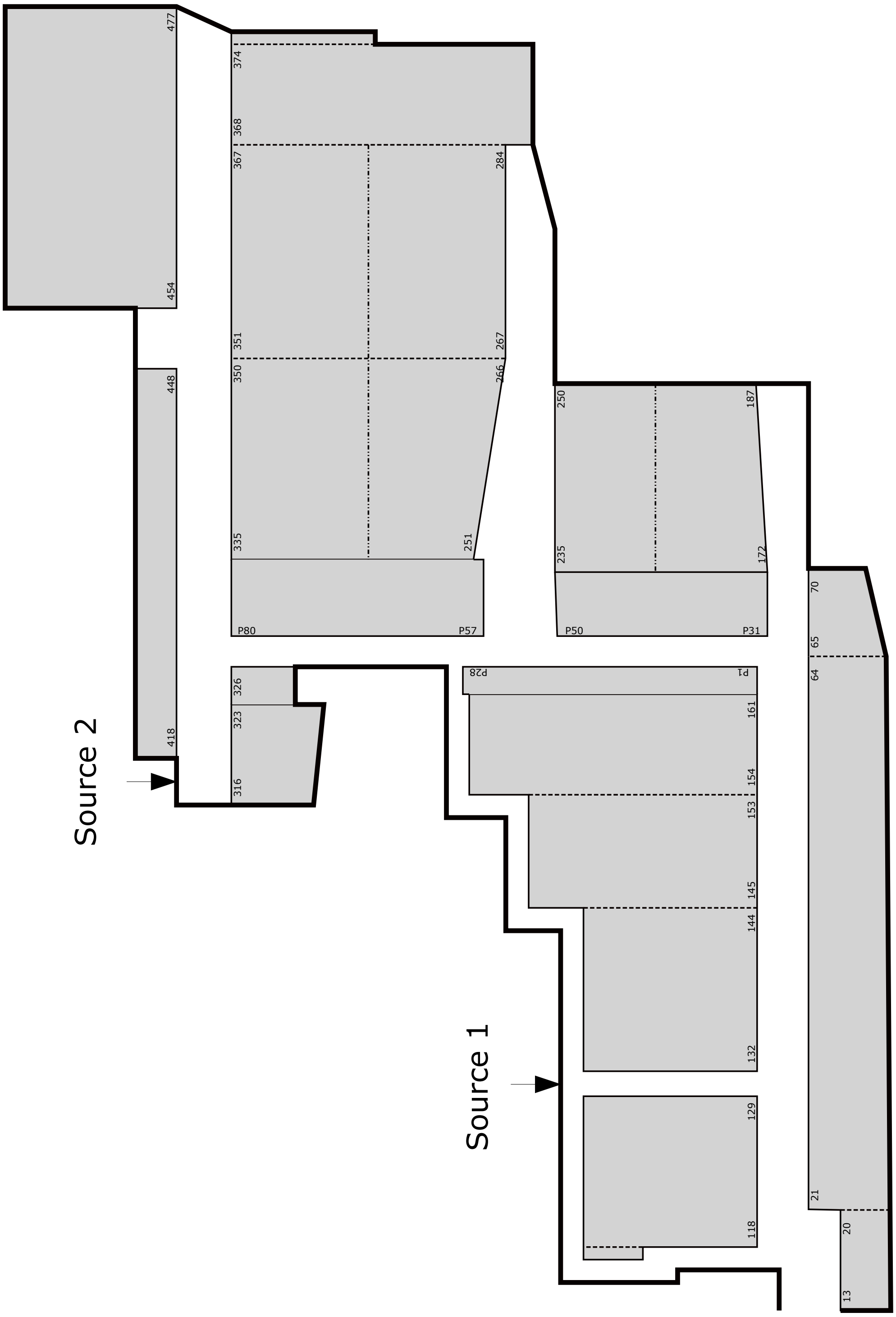
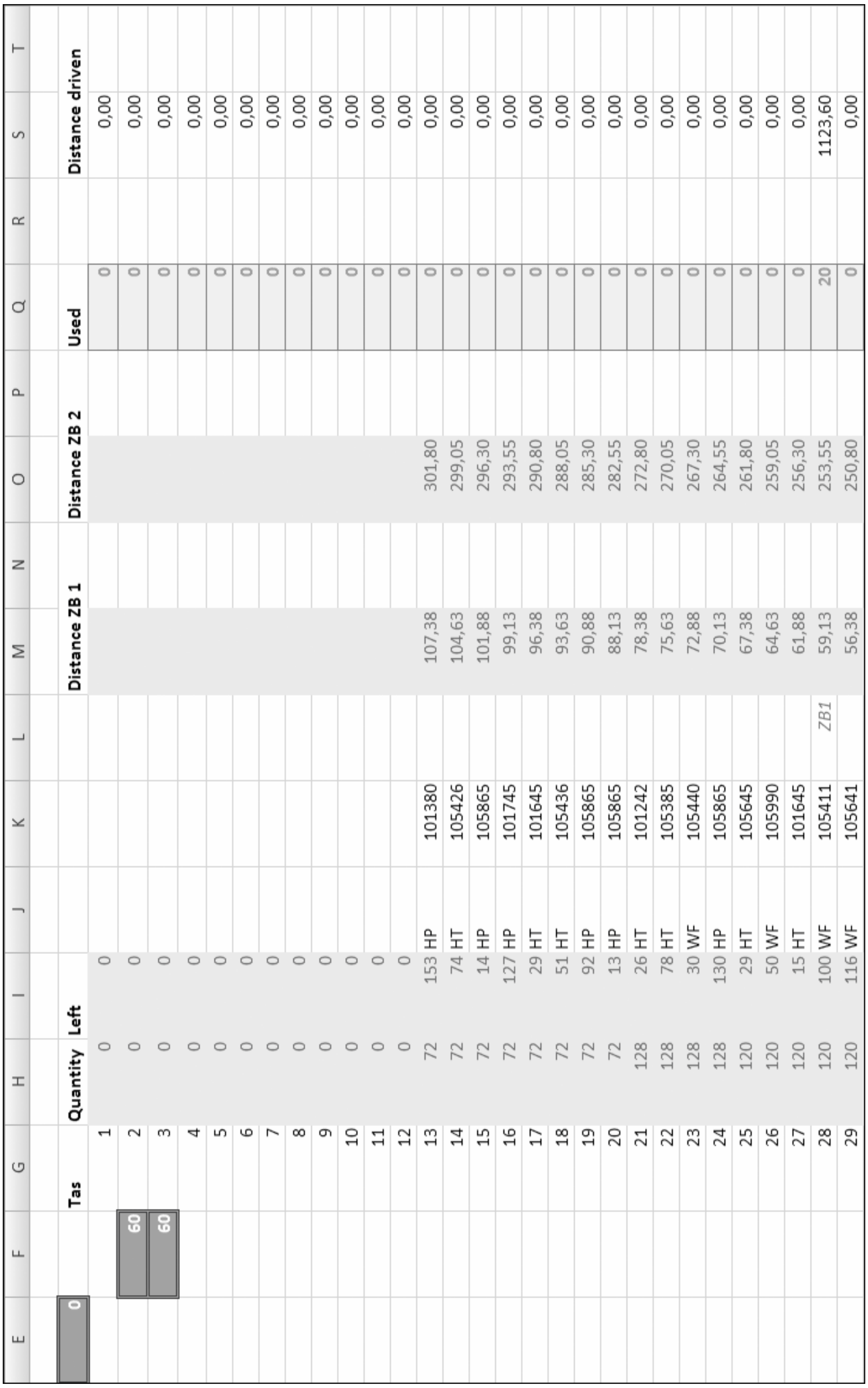


Figure 9: The brickyard of Rodruza BV.

C Screenshot of the Model

The next Figure shows a screenshot of a part of the model, where the occupation of the brickfield is stored.



E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
0		Tas	Quantity	Left				Distance ZB 1		Distance ZB 2		Used		Distance driven	
		1	0	0								0		0,00	
	60	2	0	0								0		0,00	
	60	3	0	0								0		0,00	
		4	0	0								0		0,00	
		5	0	0								0		0,00	
		6	0	0								0		0,00	
		7	0	0								0		0,00	
		8	0	0								0		0,00	
		9	0	0								0		0,00	
		10	0	0								0		0,00	
		11	0	0								0		0,00	
		12	0	0								0		0,00	
		13	72	153	HP	101380		107,38		301,80		0		0,00	
		14	72	74	HT	105426		104,63		299,05		0		0,00	
		15	72	14	HP	105865		101,88		296,30		0		0,00	
		16	72	127	HP	101745		99,13		293,55		0		0,00	
		17	72	29	HT	101645		96,38		290,80		0		0,00	
		18	72	51	HT	105436		93,63		288,05		0		0,00	
		19	72	92	HP	105865		90,88		285,30		0		0,00	
		20	72	13	HP	105865		88,13		282,55		0		0,00	
		21	128	26	HT	101242		78,38		272,80		0		0,00	
		22	128	78	HT	105385		75,63		270,05		0		0,00	
		23	128	30	WF	105440		72,88		267,30		0		0,00	
		24	128	130	HP	105865		70,13		264,55		0		0,00	
		25	120	29	HT	105645		67,38		261,80		0		0,00	
		26	120	50	WF	105990		64,63		259,05		0		0,00	
		27	120	15	HT	101645		61,88		256,30		0		0,00	
		28	120	100	WF	105411	ZB1	59,13		253,55		20		1123,60	
		29	120	116	WF	105641		56,38		250,80		0		0,00	

Figure 10: Screenshot of the model.