Optimisation of a complex manufacturing process using discrete event simulation and a novel heuristic algorithm

Blaž Rodič, Tadej Kanduč

Abstract: In this paper we present the methods and results obtained in a manufacturing process optimisation project. The client company is a major regional manufacturer of specialized furniture with 30.000 items in its catalogue. Their main goals are a reduction of manufacturing costs and order fulfilment lead time. We have used discrete event simulation (DES) to build a model that reflects the current manufacturing processes and allows us to test optimisation methods. Due to the large number of products and their manufacturing processes we have developed an automated model construction method that uses customer order data and manufacturing process database to build an ad-hoc simulation model. The model and method were tested in the first optimisation task: reduction of product travel distance through modifications of factory layout. We have developed a novel heuristic optimisation method based on force-directed graph drawing. The method outperformed other more general heuristic methods for QAP (Quadratic Assignment Problem) and produced significantly improved factory layouts.

Key Words*:* manufacturing optimization, discrete event simulation, automated model construction, factory layout optimization, heuristic optimisation

I. Introduction

UNDERSTANDING and analysing complex manufacturing systems can be a tedious and time-consuming work since the manufacturing processes are interleaved and impossible to treat separately. Manufacturing processes in larger production companies are generally complex and need to be systematically organised in order to achieve high levels of efficiency. Companies needs to consider several criteria and restrictions in the processes such as costs, due dates, amounts of stock materials, different measurements in efficiency, etc. A smaller change in a subsystem can impact the entire manufacturing system considerably. [1]

Processes are usually too complex to be modelled with exact mathematical representations as mathematical models are convenient only if the system is simple and small. Methods more suited for modelling of complex manufacturing systems include discrete event simulation (DES) modelling, which despite its simplicity can provide enough details to understand and analyse all processes on a factory floor.

Construction of a DES simulation model requires that the data that describe the manufacturing processes are obtained, analysed, extracted and prepared in a suitable format for the model. In order to maintain model accuracy despite changes in manufacturing processes, integration of simulation software, auxiliary applications and databases is necessary.

Optimisation through modification of model structure can be performed by constructing several versions of the model and input data (i.e. scenarios) and comparing simulation results. To accelerate the development of model versions and scenarios one can construct algorithms that build or modify simulation models according to model input data. This is especially useful in cases of large simulation models and if the model variants are prepared by an algorithm, e.g. an optimisation algorithm. Automated model building and modification however requires that the model structure can be modified with an algorithm, without manual interventions.

In the paper we present main steps of the project of optimising manufacturing processes in Podgorje Ltd., a Slovenian furniture company, including a novel factory layout optimisation method, based on force-directed graph drawing. Our goal was to investigate how the layout of machines on the factory floor affects the efficiency of manufacturing processes. Furthermore, the objective was to develop an optimised layout that would be implemented in real life. Our primary optimisation criterion was the total distance the manufactured products need to travel on the floor, however we have also monitored other criteria during the optimisation processes. The results of our project are used within an ongoing micrologistics optimisation process.

In the paper we highlight all important steps in simulation model development and optimisation, such as preparation of databases and interaction between the programs and algorithms for optimisation and extend the problem by minimising two dependant criteria. The considered criteria are the total costs of one-time machine relocation and the labour costs in transport of products between the machines as a result of changes in machine relocation.

A. Problem situation

The factory floor in our client company, Podgorje Ltd., contains approximately 140 machines, from simple woodworking workstations to expensive multi-purpose CNC machines (see *Figure 1*). There are more than 30,000 different

Work supported by Creative Core FISNM-3330-13-500033 'Simulations' project funded by the European Union, The European Regional Development Fund. The operation is carried out within the framework of the Operational Programme for Strengthening Regional Development Potentials for the period 2007-2013, Development Priority 1: Competitiveness and research excellence, Priority Guideline 1.1: Improving the competitive skills and research excellence.

products and semi-finished products produced in the factory. Each product is manufactured according to a specific bill of materials (BOM) and a technical procedure defining the sequence and duration of operations and machines to be used. The BOM contains a list of materials and semi-finished products required to make the given product. On average it takes 8 operations to finish a product, but the number can be from 3 to 20 depending on the complexity of the product. The technical procedure data include lists of suitable machines or groups of machines for each operation and standard machine setup and machine operation duration times. Complex products are manufactured by joining smaller semi-finished products according to the BOM. Production scheduling is based on customer orders and performed using the Preactor scheduling system. Typically, there is more than one active customer order in production at the same time. Products are manufactured in groups (series) ranging from approximately ten to several hundred pieces with a typical series containing approximately 30 pieces. For every operation, unfinished products are stored at input pallets or cart next to the machine. After an operation at a machine is finished, the entire series of products is moved by carts and pallets to the next location (machine). Therefore the total number of required transport workers depends on the number of simultaneous carts in transition between machines. The assumption in the optimisation process was that shorter transport routes will result in shorter transport worker tasks, increasing the worker availability and potentially reducing the required number of transport workers. Operation of a single machine is formally described by UML diagram in Figure 2

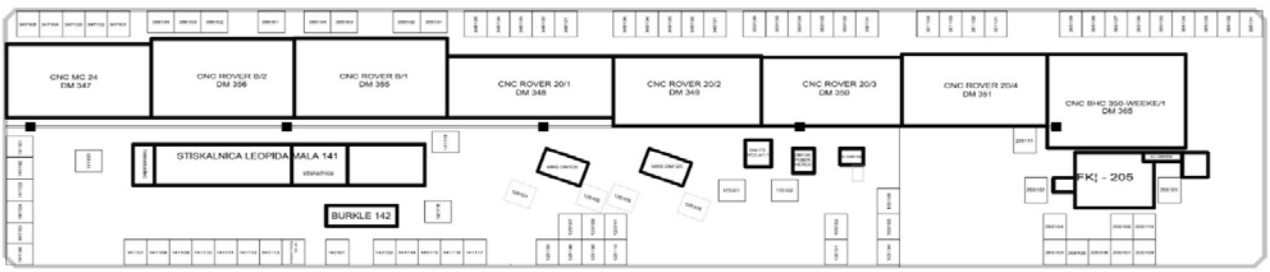

Figure 1: Section of the factor floor in Autocad DXF file

The Preactor database defines groups of equivalent machines (GEM) that can be used to perform operations defined by technical procedures. Most operations can be performed on several machines. These machines are grouped accordingly (i.e., a group of equivalent machines), with a preferred (optimal) machine defined for each group per technical procedure. Preference depends on suitability of machine for a technical procedure from the aspect of machine operation cost or operation duration. During manufacturing processes, a machine is selected from the group to perform a specific operation according to preference and machine availability. Typically a machine that is currently the least loaded in the group is chosen. Hence the manufacturing process can be referred to as “flexible manufacturing”. The simulation model has to reflect this flexibility and model the machine groups, group selection and machine selection process for each technical procedure (i.e. each product).

Manufacturing processes include large set of different products and variations of open orders during each working month. Developing a static simulation model that would cover all possible (i.e. 30,000) products that may appear in client's orders is not realistic as it takes 5 to 15 minutes to complete a model of a process for each product, and a model containing 30.000 process exceeds the memory limitations of our modelling tool (Anylogic, http://www.anylogic.com/). Instead, the model is built ad-hoc for each set of open orders. As orders change continuously, we have developed a method and application that automatically builds the model from a model template, the database of technical procedures and the database of currently open orders. Based on the list of ordered products and technical procedures only the necessary machines are placed in the model. Technical procedures and BOMs are read dynamically from input data during the simulation to adjust machine model parameters and assembly of products. The description of data structures and model building algorithm is given in later chapters.

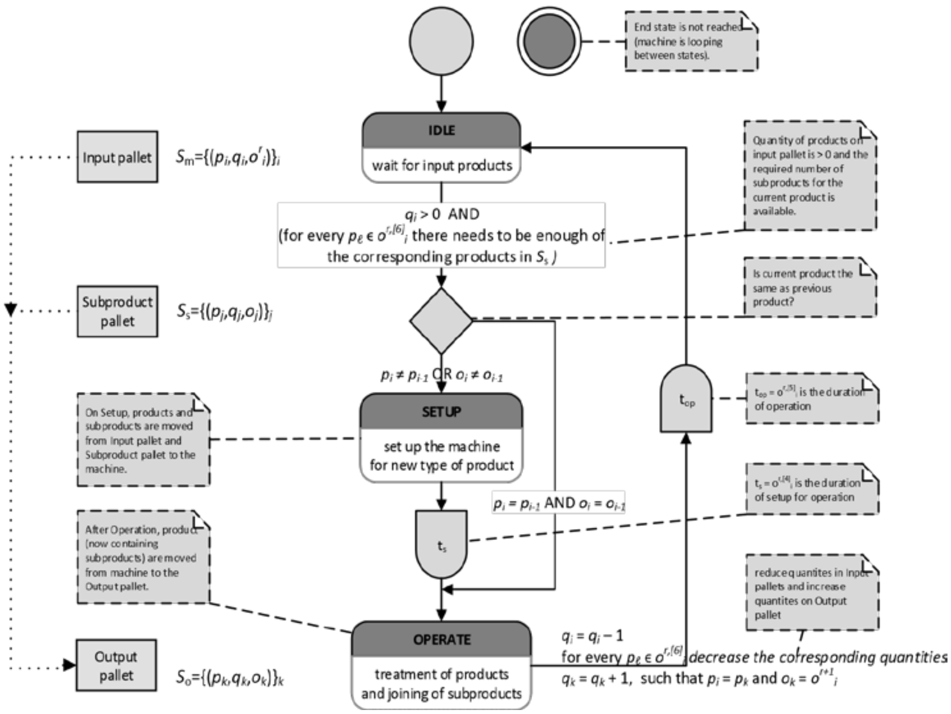

Figure 2: UML state diagram of a single machine

B. Previous research (review of literature)

Simulation is commonly used for the evaluation of scenarios [1],[2],[3],[4]. However, the models developed with the visual interactive modelling method (VIM) are usually manually constructed through careful analysis of the real-life system and communication with process owners. Automated model development is more common with methods that allow easier and more standardized formal description of models, e.g. Petri nets [5],[6]. Automation of model construction and adaptation can importantly facilitate the development of models of complex systems [7],[8] and generation of simulation scenarios.

Several papers deal with factory layout optimisation, with paper [9] stating that multiproduct enterprises requires a new generation of factory layouts that are flexible, modular, and easy to reconfigure. Evolutionary optimisation methods are often proposed due to problem complexity [10]. Layout optimisation problem is identified as hard Combinatorial Optimization Problem and the Simulated Annealing (SA) meta-heuristic resolution approach is proposed to solve to problem [11]. A novel particle swarm optimization method is proposed by [12] for intelligent design of an unconstrained layout in flexible manufacturing systems.

Factory layout design optimisation is further discussed in [13],[14] and [15]. Authors [13] propose a new facility layout design model to optimise material handling costs. Sources [14] and [15] propose genetic algorithm based solutions to respond to the changes in product design, mix and volume in a continuously evolving work environment.

The layout optimisation problem is similar to well-known quadratic assignment problem (QAP) [16],[17]. But the problem in its original form does not consider including fixed expenses to move the facilities (machines) to new locations as it is in our case.

There are numerous methods for graph drawing and class of force-directed methods are one of the most commonly used in practice due to their simplicity and visually appealing representation of the graphs (see [18],[19] and [20]) and references therein). One of the earliest and still commonly used methods was developed by Fruchterman and Reingold [20].

C. Placement of project within end-users activity

Podgorje Ltd. has been manufacturing furniture for more than half a century. During that time, customer demands changed, the number of products, size of orders and quality requirements constantly grew. Typically, new machines were added to the workshop as needed and placed within available floor space. Machine placements were determined by experiences of foremen in the shop. Typically, the machines stayed on the same location through the years and were never moved to a perhaps better location. Some machines were replaced with newer, faster and more efficient machines, but remained at the same location. Furthermore, clients' orders and technical procedures in the company have changed over the years, thus making the current factory floor suboptimal. No systematic analysis and optimisation of factory floor layout has been made by the company.

Our task in the project was to develop a better machine layout, which will fit the current production needs and projections for the next ten years. To complete this task we have developed a simulation model of the factory floor and optimisation methods based on the company data and their specific optimisation goals. Primary goal of the company is to reduce overall costs in manufacturing processes. This can be achieved by removing bottlenecks (overloaded machines), reducing transport distances (distances the carts need to travel between the machines), reducing overall time to finish a work order or by increasing overall machine utilisations. Repositioning of the machines will be done during the upcoming renovation of the factory floor. For most of the machines there are no specific location restrictions. It is also be possible to add some new machines on the floor if considerable improvements can be achieved.

II. Methodology

We have used discrete event simulation methodology to develop a simulation model that captures all of the important features of manufacturing processes. The purpose of the model is verification of new manually or algorithmically generated floor layouts. Optimisation of floor layout is conducted in cooperation with experienced manufacturing planners, managers and other experts within the company, and is facilitated by state-of-the-art optimisation algorithms that are employed to generate new layout scenarios, i.e. to search for the optimal layout within a large set of possible layouts.

A. Existing tools and data

As a part of established scheduling and planning procedure, Podgorje Ltd. uses Preactor software (http://www.preactor.com/) to schedule customer orders according to a set of priorities and availability of resources (machines) and daily monitor manufacturing processes on the factory floor. Preactor is a family of “advanced scheduling and planning” products that allows detailed definition of manufacturing and other processes and integrates with existing ERP and other company databases and applications. It allows costing, inventory control, transaction control, detailed management and monitoring of resources and orders. Since unplanned events can occur during manufacturing processes, Preactor can adapt current schedules and generate minor scheduling modifications/optimisation options.

However, the modelling process within Preactor is not flexible enough to allow easy modification of the system model or modelled processes and testing of scenarios, required for layout or process optimisation. To simulate processes in a different factory floor layout, an entire simulation model needs to be built from scratch or undergo lengthy manual modification. Preactor also does not offer physical layout modelling and 2D modelling of machine position and travel of products – this aspect is modelled as time required for transition of product between machines.

B. Selection of tools and methods

We decided to implement current production processes and optimisation procedure with a specialised simulation and modelling tool Anylogic – a powerful software that implement DES, SD (system dynamics) and agent based modelling (ABM) methodologies. Modelling is performed using VIM approach which is intuitive and clear, and it supports advanced visualisations techniques. Anylogic or other simulation and modelling tools are not a replacement for advanced scheduling and planning tools as Preactor or vice versa. Instead, they complement each other: Preactor contains a detailed process model that allows accurate scheduling and planning and provides detailed process data for Anylogic, while Anylogic allows fast design and optimisation of processes, addition of new machines and verification of scenarios using different factory layouts and sets of orders. The resulting optimal or sub-optimal layout selected by the customer (Podgorje Ltd.) can then be implemented in real life. Hence, Anylogic output can be used to simplify the design of a new Preactor model.

The existing implementation of Preactor in Podgorje has significantly accelerated our modelling process in Anylogic, as nearly all the required data on manufacturing processes has already been collected and stored in a relational database. Actual factory layout described in Autocad DXF file was used to design the 2D network of machines and paths between machines in Anylogic. To illustrate, the cataloguing of all the manufacturing processes and design of database in Podgorje within the Preactor project has taken about a year to complete.

The simulation model allows us to monitor various manufacturing process statistics and to better understand the manufacturing system by discovering rules and connections in the manufacturing system. The model was verified by comparing the simulation results (e.g. manufacturing time, machine utilisation) using synthetic and real historic order data prepared by the company planners with the statistics that were generated in the manufacturing of the set of orders in the past year. The model was prepared using VIM tool (Anylogic) and can be easily adapted in order to test the effects of alterations to the manufacturing process, floor layout or the set of orders, which accelerates the optimisation process considerably.

An important part of the project was the preparation and export of manufacturing process data and customer order data from the company database and the connection of all software components (databases, simulation model, model construction application and auxiliary applications) in order to achieve the required level of integration.

C. Data based modelling

Manufacturing process data includes the data for technical procedures and BOM, and is stored in Microsoft SQL Server database, which serves several applications used by the company but mainly used by the manufacturing scheduling application Preactor. Preactor is used in Podgorje for the generation and scheduling of manufacturing orders based on customer orders and for online monitoring of the manufacturing processes via approximately 100 control points. This allows the company to daily update and if necessary modify the manufacturing schedule.

We have analysed the structure and content of database tables and prepared a set of queries that were used to extract the data required for model construction and simulation scenario generation, i.e. the preparation of model input data. The queries were stored in the Microsoft SQL Server database in the form of views and later called by an Microsoft Excel workbook that was used as an intermediate data storage that allowed us to examine and modify the data as required. Some corrections were necessary as the original database contained some errors and some data was missing for certain technical procedures and BOMs. This is an inevitable step when dealing with real-life data. Table 1 shows an example of an SQL query used to obtain the data on machines in machine groups (referred to as Resources and ResourceGroups).

Table 1: Example of an SQL query

```
/*Furniture company_baza_20140403.LSI.*/
CREATE VIEW Test19Projects_equivMachines AS
        SELECT ResourceGroupId, RGR.ResourceId,
      ResourceCode FROM
              Furniture
              company_baza_20140403.LSI.ResourceGroupReso
              urces RGR, Furniture
              company_baza_20140403.LSI.Resources R
      WHERE RGR.ResourceId=R.ResourceId
;
```

D. Input and output data

All the input data (orders, technical procedures, BOMs, list of GEMs) are primarily stored in SQL databases, generated by Preactor software. Relevant data are saved as queries and exported to intermediate Excel file. In Excel, the data are slightly manually modified, since inaccurate and inconsistent in real data occasionally occur. In Excel, the following input data are stored:

- An order is described as a list of products (catalogue numbers). For every product from the list, name, quantity, earliest start time, priority parameter and volume are assigned.
- Each product has a specific technical procedure. For every operation there is a group of equivalent machines, a preferred machine, set up time and time per item.
- More complex products also have bill of materials, a list of required semi-finished products or materials that are joined at a specific operation in specific quantity.

At start-up of the simulation, input data from Excel are read and stored in internal Anylogic arrays. From there on, all data are read from internal data structures to remove constant communication with external files, which would slow down the simulation.

During simulation, various statistical data are measured and stored:

- For every pair of machines, different types of flows (number of products, number of used carts, overall volume of products and total distance of carts) are measured.
- For every machine, utilisation, overall setup time, flow of products and volume, and queue of products are monitored.
- For each series of products, completion times and sequences of machines, which were chosen during simulation, are stored.
- Different, less significant measurements, such as flow of carts and routes of the carts, are recorded.

Once the simulation is finished, all the data are stored in the output Excel file.

E. Simulation model

The simulation model was prepared using DES methodology in Anylogic software. Anylogic stores the models as standard XML files, which allows easy manual or algorithmic modifications of the model. To this end we have developed an application in Java that reads input data from Excel and constructs the corresponding Anylogic model by modifying a template model. Layout of machines and the underlying network of nodes describing the paths between the machines were designed according to the actual factory layout described in an Autocad DXF file (see *Figure 1*).

Output data of the simulations, such as time, utilisation of machines, product quantity flows, supply levels and product travel distance, are stored by the model in an Excel file. This allows additional manipulation of the data and data visualisations. Every machine is modelled as a machine block in Anylogic as shown in *Figure 2*. On the input of the block, carts filled with products enter the system. Products are sorted according to their type at *productOrSemifinished*. The corresponding *sinks*, *sink1* and *sink2* monitor products on input pallets. Once the product is chosen for operation, it is injected at source. Blocks *setUpMachine* and *machineDelay* are standard service blocks. Block *waitForWholeSeries* plays a role of output pallet. Products wait there until complete series of products is finished. Some products need to wait at *dryingDelay* according to the technical procedure (paint, varnish drying, etc.). Filled carts are injected at *cartSource* and moved to the next location at *moveCartTo*. Output of the main machine block sends a cart to the input of the next machine.

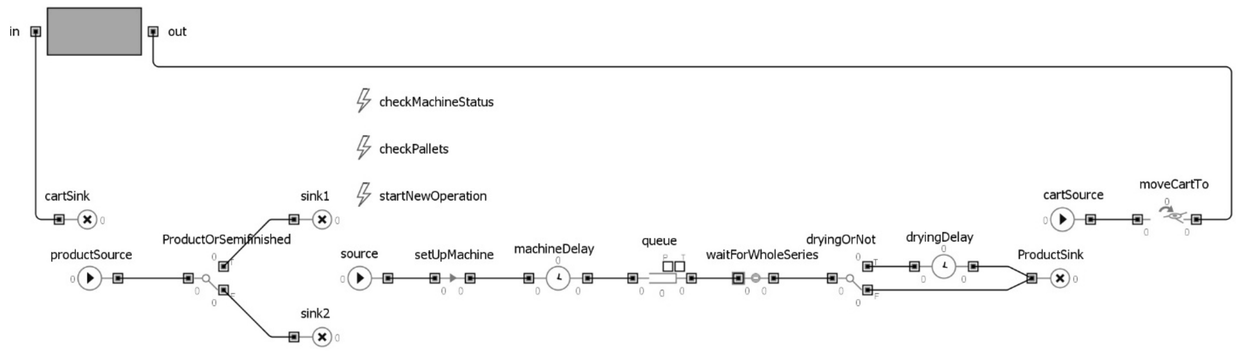

Figure 3: Schematic of a single machine in the DES model

F. *Components of the simulation system*

Modelling and simulation system is composed of four main elements:

- Core manufacturing process simulation model in Anylogic environment.
- Java application that constructs XML Anylogic model from a template file.
- MS Excel as an intermediate input and output data storage, and analysis tool.
- MS SQL server database describing technical procedures and client's orders.

The resulting system is shown in *Figure 3*. Firstly, we prepare Anylogic template file (XML). Simulation model (new Anylogic XML file) is constructed by running the Java algorithm for automatic model building. Next, we run the Anylogic simulation model. During simulation, input database is read dynamically. When simulation is finished, simulation results are stored in output Excel file.

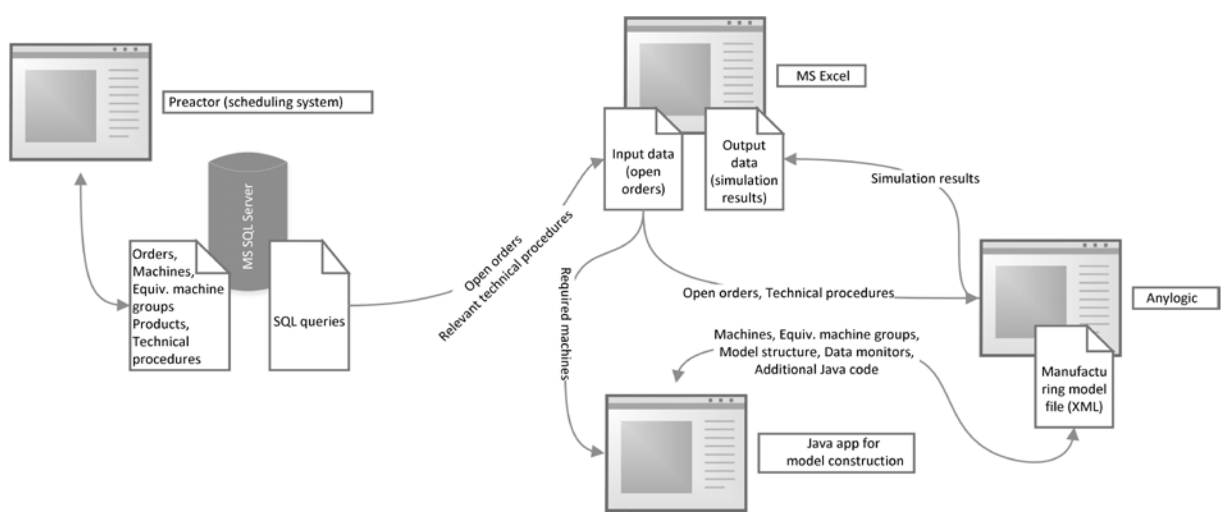

Figure 4: Simulation system schematics

G. *Optimisation methods*

In this section we describe the problem of finding the factory floor that minimises total transportation distances of the products during the production. The costs of manufacturing can be reduced by decreasing the need for labour in the transport of products between machines through better machine placement, i.e. factory layout. However, relocation of a machine is a difficult and costly measure and disrupts the manufacturing process. Therefore it makes economic sense to move a machine only if the relocation will considerably reduce product travel distance and consequently the need for labour. Namely, relocating the machines is associated with additional expenses: moving the machine m_i costs g_i amount of currency. Presumably, good candidates for relocation are machines with high product flows.

Various model statistics are measured during the simulation runs. Most important in this part of the project is the flow of products $\tilde{f}_{ij}$ between a pair of machines m_i, m_j, $i,j = 1,2, \ldots, N$. Flow $\tilde{f}_{ij}$ represents the total amount of volume of products that was directly transported between these two machines. From the product flow $\tilde{f}_{ij}$ we straightforwardly compute cost flow $\tilde{f}_{ij}$, i.e., cost to move all products between the two machines for distance of 1 m. The distances between the machines are yet not known and are obtained from the optimisation of the factory floor layout. To reduce the overall costs we need to solve the following optimisation problem:

$$\min_{\{p_1,p_2,\ldots,p_N\}} \left(\sum_{\substack{i,j=1 \\ i\neq j}}^{N} f_{ij} \cdot d\left(p_i, p_j\right) + \sum_{i=1}^{N} g_i \right) \quad (1),$$

Where p_i represents position of machine m_i and d is a distance functional. If we neglect the costs g_i and restrict the positions p_i to a predefined grid, the problem simplifies to well-known quadratic assignment problem (QAP). The latter is NP-hard optimisation problem and exact optimisation methods are successful only for smaller number of machines, usually around N > 30. In our case, where N = 140 , the exact methods are not feasible and we need to apply heuristic methods instead, which in practice return a near optimal solution.

We have tested different optimisation algorithms to minimise the total distance of the products. We have tested freely available open source heuristic algorithms in C++ and Matlab for quadratic assignment problem that are based on simulated annealing [21], iterative local approach [22] and ant colony algorithm [23]. As an alternative to QAP algorithms, we have developed a promising alternative optimisation method, which is based on force-directed graph drawing methods. For every machine we calculate an attractive force, which is proportional to the weight and the distance and the corresponding repelling force towards every other machine. Attractive forces move the machines with larger volume transactions closer to each other. Repelling forces keep the machines away from each other since we want sufficient space between the machines. The machines are repositioned according to the defined forces in the system. When the machines do not move any more, the system has reached a local minima.

The problem is presented as finding the optimal mathematical network, in which nodes of the network represent the machines on the factory floor and weighted edges between the nodes represent transactions between the machines. Real routes on the floor between the machines are neglected in this case, since it considerably complicates the optimisation problem. The optimisation method should only propose a basic outline of the layout, since the final layout needs to be further tuned by the company experts to meet other less precise criteria.

Factory floor is described as a region Ω in the plane $\mathbb{R}^2$. We will simplify the problem by restricting Ω to the rectangular shape,

$$\Omega = \left\{ (x, y) \in \mathbb{R}^2; x_{min} \le x \le x_{max},\ y_{min} \le y \le y_{max} \right\} \quad (2),$$

Where x_{min}, x_{max}, y_{min}, y_{max} represent boundaries of the rectangular factory floor.

Let us denote machines by m_i , i = 1,2, …, N. Position of the machine m_i is described by

$$p_i = \left\{ \left(x_i, y_i \right) \in \mathbb{R}^2 \right\}. \quad (3)$$

Each machine takes certain amount of space which can be conveniently described by a metric rectangular-like ball $B_{ri}(p_i)$ with radius r_i and centre p_i in ∞ norm L_∞,

$$B_{(r_i)}(p_i) = \begin{Bmatrix} (x, y) \in \mathbb{R}^2; d_\infty((x, y), p_i) = \\ max\left\{ | x - x_i |, | y - y_i | \right\} < r_i \end{Bmatrix}. \quad (4)$$

For every pair of machines m_i and m_j , $i,j = 1,2, \ldots, N$, we obtain a flow of products $f_{ij} > 0$ as a result of the simulation of the manufacturing processes.

Distance $d(m_i, m_j)$ between the pair of machines m_i and m_j is defined as the shortest path between the machines in a predefined network of routes.

The optimisation problem of minimising the total distance is described as

$$\min_{\{p_1,p_2,\ldots,p_N\}} \left(\sum_{\substack{i,j=1 \\ i\neq j}}^{N} f_{ij} \cdot d\left(m_i, m_j\right) \right), \quad (5)$$

where positions p_i p_i must satisfy the conditions

$$B_{r_i}\left(p_i\right) \cap B_{r_j}\left(p_j\right) = \varnothing \quad (6)$$

for every $i \neq j$ and

$$B_{r_i}\left(p_i\right) \subset \Omega \quad (7)$$

for every $i = 1,2, \ldots, N$.

The first conditions states that the regions of machines must not intersect between each other and the second that every machines must lie entirely in the factory flow.

For every layout of machines one would also need to define a suitable network of routes. To simplify the tedious problem of defining the network from the machine positions, we presume the distance between the machines is a well-known Manhattan distance,

$$d_M\left(m_i, m_j\right) = \left|x_i - x_j\right| + \left|y_i - y_j\right| \ . \tag{8}$$

Since the original routes in the factory are defined on a rectangular grids, differences in lengths of paths, if we use the functional d_M instead of d, are small.

If we presume that all machines take the same amount of space on the floor (all r_i are the same) we can restrict the positions p_i to discrete points on a predefined grid. Hence the problems simplifies to well-known quadratic assignment problem.

H. Force-directed graph drawing algorithm

In this section we present the heuristic optimisation algorithm for assigning positions p_i to machines m_i. The algorithm is based on force-directed graph drawing methods. Every machine is presented as a node on a plane. To every node n_i we prescribe the corresponding repelling force F_{ij} to all other nodes n_j,

$$F_{ij} = H_{ij}\left(\left\|p_j - p_i\right\|_2\right) \cdot \frac{p_j - p_i}{\left\|p_j - p_i\right\|_2} \ , \tag{9}$$

Where H_{ij} is a positive monotonically decreasing function. Typically, H_{ij} is defined as $H_{ij}(r)=r^{-2}$. Repulsive forces keep the nodes away from each other since we want sufficient space between the machines.

For every pair of nodes n_i, n_j we define a weighted edge e_{ij} with weight f_{ij}. Attractive forces between the nodes are defined as

$$G_{ij} = -f_{ij} \cdot I_{ij}\left(\left\|p_j - p_i\right\|_2\right) \cdot \frac{p_j - p_i}{\left\|p_j - p_i\right\|_2} \tag{10}$$

Where I_{ij} is a positive monotonically increasing function. In our case, I_{ij} is defined as $I_{ij}(r) = d_M(p_i, p_j)$. Attractive forces move the nodes with large edge weights closer to each other.

To keep the nodes inside the prescribed location Ω, we also need to define forces that pull the nodes back to the interior if they are outside the prescribed region Ω,

$$J_i = \begin{cases} 0, \ p_i \in \Omega \\ dist\left(p_i, \Omega\right), \ p_i \notin \Omega \end{cases} \ , \tag{11}$$

And $dist(\cdot,\cdot)$ is a function measuring the distance between objects.

A. Implementation of the layout optimisation algorithm

The force-directed graph drawing algorithm was implemented in Anylogic using built-in discrete event and system dynamics elements and Java code, therefore its implementation can be considered a hybrid DES/SD layout optimisation model. This heuristic optimisation algorithm is implemented in custom Java code within the model. Machines in the model are represented as nodes in a network.

To prevent clustering and immobilization of machines in the layout model the machines are added to the model sequentially. Machines are added to the model in order of decreasing machine product volume in order to ensure that the most important machines are most likely to be eventually moved to a near-optimal position and that their movement is not obstructed by machines with less product volume.

To prevent deterministic approach to a local minimum, a random element was added to the algorithm: the starting position of machines is randomized. Therefore several runs of the algorithm are required to get a good set of possible layouts.

III. Results and discussion

The main outcomes of the project are an integrated simulation model of the factory in Anylogic that communicates with external database files, the method for automatic model construction and it's implementation in Java and the method (heuristic optimisation algorithm) and its implementation in Anylogic, that generates optimized layout of machines on the factory floor. The optimisation criterion is the total distance the products travel while being manufactured. The heuristic optimisation algorithm outperformed other more general heuristic methods for QAP (Quadratic Assignment Problem) in terms of the optimisation criterion. The factory model serves as an indispensable tool for in-depth analysis of the manufacturing process.

To obtain an optimized floor layout based on real historic data, our final simulation test included 19 historical orders and approximately 440,000 ordered products, which corresponds to a month's worth of customer orders. The newly proposed layout has up to 28% shorter total product travel distance than the current layout, depending on the layout version. Results and progress of the optimisation for five main proposed machine layout are presented in *Table 1*. The first layout in the table is the current, unmodified factory layout. The second layout was developed manually by factory planner. The third layout was generated by our heuristic method and then modified by factory planners to satisfy company requirements regarding micrologistics (current paths of products and current location of installations). The fourth and fifth layouts were generated by our heuristic method in two different runs. Due to heuristic nature of our method, the results cannot be referred to as "optimal", however they are significantly better than the status quo or the manually modified status quo layout.

Table 1: Total product travel distance optimisation results.

Layout version	Total production time	Total product travel distance	Relative product travel distance
1: Status Quo	30.9 days	690 km	100 %
2: Manual	30.6 days	617 km	89 %
3: Modified auto.	30.2 days	564 km	82 %
4: Auto. A	30.3 days	506 km	73 %
5: Auto. B	30.2 days	492 km	72 %

Shorter travel means less time is required for transport of products. As transport is performed by workers pushing the carts, this means that fewer carts and workers will be required. Other workers can then be relocated on other assignments on the factory floor. The customer has responded very favourably to these results, and is willing to implement the suggested changes. They have also prepared several manually adjusted floor layout based on our generated layout and submitted them to us for verification with the simulation model.

An interesting discovery is that the optimisation of layout for shortest product travel distance only negligibly affected the total manufacturing time for the given set of orders. The result is however predictable since machine operation times are much longer than transport times. Further steps in our project will include alterations to the set of machines: replacement of one or several machines by newer multipurpose CNC machines. Other optimisation goals and criteria will be explored. One parameter that we will need to examine is the sequence of production of orders. The due dates for orders are fixed, however the sequence may influence the total order manufacturing time. We will examine the scheduling problem modelling methods described in [24].

To significantly reduce the total manufacturing time, the company would need to buy additional CNC machines to remove the existing bottlenecks - several CNCs have a very high (70%+) utilization. As CNC machines are expensive (order of 100.000 EUR), purchase of new machines will be considered only in the frame of within a currently running micrologistics optimisation process, which also includes new transport methods

The current model for cart travel assumes constant cart speed based on estimation of average speed and uses optimal path between nodes (machines). In the further research we will consider additional modes of transport such as automated guided vehicles (AGV) and optimisation of their path using methods such as described in [25].

REFERENCES

[1] M. Kljajić, I. Bernik, A. Škraba, "Simulation approach to decision assessment in enterprises," *Simulation*, 75(4), 2000, pp. 199-210.

[2] R. S. Edis, B. Kahraman, O. U. Araz and M. K. Özfirat, "A facility layout problem in a marble factory via simulation," *Mathematical and Computational Applications*, 16(1), 2011, pp. 97-104.

[3] P. Tearwattanarattikal, S. Namphacharoen and C. Chamrasporn, "Using ProModel as a simulation tools to assist plant layout design and planning: Case study plastic packaging factory," *Songklanakarin Journal of Science and Technology*," 30(1), 2008, pp. 117-123.

[4] E. Romano, C. Cesarano, and D. Iuliano, "An hybrid simulation model to support decision making in a manufacturing plant," *WSEAS Transactions on Systems 13*, 2014, pp. 312–321.

[5] R. Conner, "Automated Petri net modeling of military operations," in *IEEE Proceedings of the IEEE 1990 National Aerospace and Electronics Conference - NAECON 1990*, Volume 2, Dayton, Ohio, USA, 1990, pp. 624-627.

[6] D. Gradišar and G. Mušič, "Automated Petri-Net Modelling for Batch Production Scheduling," in (ed: Pawel Pawlewski) *Petri Nets - Manufacturing and Computer Science*, InTech, 2012, pp. 3-26. http://dx.doi.org/10.5772/48467

[7] A. D. Lattner, T. Bogon, Y. Lorion and I. J. Timm, "A knowledge-based approach to automated simulation model adaptation," in *Proceedings of the 2010 Spring Simulation Multiconference (SpringSim '10)*, Society for Computer Simulation International, San Diego, CA, USA, 2010, Article 153.

[8] R. Kannan and H. Santhi, "Automated construction layout and simulation of concrete formwork systems using building information modeling," in (eds: Djwantoro Hardjito & Antoni) *Proceedings of The 4th International Conference of Euro Asia Civil Engineering Forum 2013 (EACEF 2013)*, National University of Singapore, 26-27 June 2013, pp. C7-C12.

[9] S. Benjaafar, S. S. Heragu and S. A. Irani, "Next generation factory layouts: Research challenges and recent progress," *Interfaces*, 32(6), 2002, pp. 58-76.

[10] A. Sadrzadeh, "A genetic algorithm with the heuristic procedure to solve the multi-line layout problem," *Computers and Industrial Engineering*, 62(4), 2012, pp. 1055-1064.

[11] G. Moslemipour and T. S. Lee, "Intelligent design of a dynamic machine layout in uncertain environment of flexible manufacturing systems," *Journal of Intelligent Manufacturing*, 23(5), 2012, pp. 1849-1860.

[12] M. Ficko, S. Brezovnik, S. Klancnik, J. Balic, M. Brezocnik, and I. Pahole, "Intelligent design of an unconstrained layout for a flexible manufacturing system," *Neurocomputing*, 73(4-6), 2010, pp. 639-647.

[13] K. K. Krishnan, S. H. Cheraghi and C. N. Nayak, "Facility layout design for multiple production scenarios in a dynamic environment," *International Journal of Industrial and Systems Engineering*, 3(2), 2008, pp. 105-133.

[14] J. S. Kochhar and S. S. Heragu, "Facility layout design in a changing environment," *International Journal of Production Research*, 37(11), 1999,pp. 2429-2446.

[15] M. Enea, G, Galante and E. Panascia, "The facility layout problem approached using a fuzzy model and a genetic search," *Journal of Intelligent Manufacturing*, 16(3), 2005, pp. 303-316.

[16] T. Koopmans, and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*, vol. 25, 1957, pp. 53–76.

[17] S. Sahni, and T. Gonzalez, "P-complete approximation problems," *Journal of the ACM*, vol. 23, 1976, pp. 555–56.

[18] S. G. Kobourov, "Force-directed drawing algorithms", in (R. Tamassia, Ed.) *Handbook of Graph Drawing and Visualization* CRC Press, 2013, pp. 383–408.

[19] M. J. Bannister, D. Eppstein, M. T. Goodrich, and L. Trott, "Force-directed graph drawing using social gravity and scaling," *Lecture Notes in Computer Science 7704*, 2013, pp. 414-425.

[20] T. Fruchterman, and E. Reingold, "Graph drawing by force-directed placement," *Software-Practice and Experience*, vol. 21, 1991, pp. 1129–1164.

[21] E. Taillard. (1998) *Simulated annealing (SA) procedure for the quadratic assignment problem*, [Online]. Available:http://mistic.heig-vd.ch/taillard/codes.dir/sa_qap.cpp.

[22] S. Shah. *Implementation of iterative local search (ILS) for the quadratic assignment problem*, [Online]. Available: http://shah.freeshell.org/ilsassignment/ cILSAssignment.cpp.

[23] K. Tsourapas. (2008) *Ant Algorithm for the Quadratic Assignment Problem*, [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/1663-qap.

[24] B. Chramcov, B., R. Bucki, "Information control of the autonomous production system," *International Journal of Mathematics and Computers in Simulation*, 6(2), 2012, pp. 298–305.

[25] T. Galpin, Y. Han, and P. Voulgaris, “Fuel minimization of a moving vehicle in suburban traffic,” *WSEAS Transactions on Systems and Control* 9, 2014, pp. 672–686.