

USING ERP-DRIVEN FLOW ANALYSIS TO OPTIMIZE A CONSTRAINED FACILITY LAYOUT PROBLEM

Andreas Beham^(a), Monika Kofler^(b), Stefan Wagner^(c), Michael Affenzeller^(d), Walter Puchner^(e)

^(a-d) Upper Austria University of Applied Sciences
School for Informatics, Communications, and Media
Heuristic and Evolutionary Algorithms Laboratory
Softwarepark 11, 4232 Hagenberg, Austria

^(e) ROSENBAUER INTERNATIONAL AG
Paschinger Strasse 90
4060 Leonding, Austria

^(a)andreas.beham@fh-hagenberg.at, ^(b)monika.kofler@fh-hagenberg.at, ^(c)stefan.wagner@fh-hagenberg.at,
^(d)michael.affenzeller@fh-hagenberg.at, ^(e)walter.puchner@rosenbauer.com

ABSTRACT

Enterprise Resource Planning (ERP) systems are nowadays widely used in large companies and even starting to appear in small and medium sized businesses. These systems hold many enterprise specific aspects and store them in a machine readable format. In this paper we will show how to use job dependency and demand information extracted from the data of a certain ERP system to simulate the process and material flows in a given production scenario. Material flows are among the most expensive processes in manufacturing businesses since they do not increase the value of the manufactured goods. We will show how to use the simulated flow information to optimize the arrangement of work centers. This problem is defined as a constrained facility layout problem using several real-world constraints in order to find realistic solutions that are close to implementation. The goal is to reduce traffic and increase efficiency on the shop floor.

Keywords: material flow simulation, facility layout problem, constrained optimization

1. INTRODUCTION

The wide availability of ERP systems in manufacturing companies has made it considerably easier to obtain data about manufacturing processes. Data can be easily exported or accessed directly through the database and a number of possibilities have opened regarding the automated analysis and, more importantly, steering of the enterprise. Several ERP systems already come with planning modules and demand for customer specific automated production steering has risen over the years.

Material-handling costs are among the highest cost factors in many manufacturing businesses these days. Their expensiveness simply stems from the fact that material handling increases the cost of the product without increasing the value of the product. Material

handling however is a necessity for manufacturing and thus increasing the efficiency of the process is one of the goals successful companies strive for.

There exist a multitude of different material-handling systems from simple conveyors to complex automated guided vehicles with readily available commercial solutions. These systems have their advantages and disadvantages and carefully choosing the right type is certainly advised. But, regardless of the underlying system the simple fact that one unit of material is transported from A to B means that there is cost involved and usually an increasing distance also means increasing costs.

One way to consider this general problem of transportation within an enterprise is to first identify an optimal arrangement of the underlying machines, and then decide on the material handling system. This becomes more and more important the more time has passed after a layout has been implemented, for example when the company has grown over the years, new machines and technologies have replaced older ones and the original assumptions regarding the flow of materials are not up to date with the actual layout. On the other hand if the arrangement cannot be altered easily the problem becomes highly constrained, several facilities need to remain in place, some need to maintain a certain distance between each other or to infrastructure end points and others need to remain separated from each other by a certain distance. The costs of moving the work centers on the shop floor may have to be taken into account when they can be estimated accurately enough.

Because of the dynamic nature and change in production processes as well as in the product portfolio it is important to continuously monitor the layout and plan ahead to be able to make the decisions on time. Ideally, the processes and flows thus should be generated automatically from the company data and

presented to the layout planner to make effective decisions quickly, based on past and present data.

1.1. Literature Review

Existing work on facility layout optimization seems to concentrate more on the problem of solving the layouting scenario with fully parameterized models than treating the question about getting the parameters. In the view of the authors, to gather the right data and configure the models requires a deep understanding of the processes under consideration. The task of obtaining the flow values is not always a trivial one and there are several obstacles present that one has to overcome. The following brief review lists some recent publications on the topic of facility layout optimization.

(Benjafaar 2002) shows the difficulty still present with the “simplified” view on layout optimization as a quadratic assignment problem (QAP). The assumption that shorter connecting paths are beneficial to the underlying plant does not hold in all cases. Several situations are shown in which the work in process (WIP) increases while the formulation of the QAP attributes the layout a better fitness. The paper concludes that even departments without material flows can have a strong relationship e.g. when they share the same material handling resource.

(McKendall, Shang and Kuppusamy 2006) as well as (McKendall and Hakobyan 2009) investigate the case of dynamic facility layouting problems, that is they consider rearrangement costs as well as material handling costs when optimizing over several periods with different flow characteristics. In the 2006 article they describe two simulated annealing metaheuristics with look-ahead/look-back strategies adapted to the dynamic facility layout problem that they test on a problem instance taken from the literature. The problem formulation is still very close to the quadratic assignment problem (QAP). In the 2009 article they use a tabu search heuristic to optimize the layout of rectangles on a continuous floor. Their results are interesting for future work when trying to find those points in time that benefit from a reorganization automatically.

(Scholz, Petrick and Domschke 2009) describe a different approach to optimizing facility layout in that they use a slicing tree representation and a tabu search heuristic for optimization. The slicing tree is a binary tree with the departments as leafs and the nodes specify whether its sub-nodes are vertically aligned or horizontally aligned. While this representation is quite interesting, the problem is that the departments are always packed tightly and that it is difficult to extend the problem into situations with e.g. distinguished locations for placing the departments. The authors have addressed some of these concerns, for example including aisles in a very recent publication (Scholz, Jaehn and Junker 2010). It is however a very interesting approach as the representation automatically locates departments close together. In a coordinate based representation the algorithm needs to find the right

layout by manipulating the coordinates, which is a more general abstraction of the problem.

In this paper we will focus a little more on the parameterization of the models and show how to use data from the ERP system to perform a flow simulation and calculate the process and material flows between the work centers of a given shop floor. With these results we will look at how to model the problem of arranging these work centers and show some of the constraints that are considered important. Finally we will apply an optimization method to obtain high quality solutions and show the results given a close to real-world instance of the problem.

2. FLOW SIMULATION

ERP systems hold the relevant data for the operation of manufacturing companies. Common to most implementations is the notion of a *job* that is split into several *operations* which are performed using several *resources*. A job results in a product or material which can be added to the company warehouse or which is finished and shipped to the customer. The operations describe basic tasks that need to be completed to finish such a job and are executed in a given order. Operations are not limited to production tasks; they may also include management tasks such as monitoring or coaching. Operations can be visualized via a connected graph that provides information on the predecessor-successor relationship between them. Any given operation may have multiple successor and multiple predecessor operations.

The resources that are used to perform the operations range from manpower to machines, raw materials and tools. While job and operations are abstract concepts resources are real. For the purpose of layout optimization some of these resources are considered part of the optimization such as manpower, machines. These are assumed to be grouped and available at fixed locations, but there is a *demand* of raw materials and tools that flows between the locations or warehouses. This demand holds information about which materials flow in the production facility.

Given this representation of a manufacturing process consisting of jobs, operations, resources, and demands we simulate the execution of these jobs and obtain the flows in the form of a matrix specifying the strength of the source-destination relationship for the resources that are to be located. To calculate the actual strength value several different ways are identified.

The strengths can be accumulated by the weighted number of transitions between any two operations. If the weights are equal to 1 each transition is considered to be an atomic event, if the weights are set to a different value for each such event different situations emerge. For example if the actual number of resources being transported is used as weight the flow strength denotes the total amount of materials being passed. It depends on the actual shop floor however if such a number is realistic or rather misleading. If the material diversity is high the transportation of a hundred small

screws does not likely represent an effort similar to the transportation of a single big item. Choosing the right weighting factor is an important step in the preparation of the problem data.

Other impacts on the outcome of the flow simulation emerge in 1:N, and N:1 transitions. When a single operation has several possible successors or predecessors, the question arises to which successor the actual flow is moving. Two possible ways that this can be dealt with, without specifying additional data such as process flow charts, are to either split and combine or duplicate the handling events. This does not seem to be an unrealistic assumption per se, given that the successor relationship implicitly encodes dependency information and waiting conditions for those items that leave the machine on which the operation is executed.

Which one of these possibilities to interpret the given data is more appropriate depends largely on the problem situation and can even depend on certain parts of the problem situation. It is necessary to discuss and decide on these possibilities in the preparation stage. Otherwise the strength of the flow is not a valid approximation of the necessity of the two involved departments to be located closer to each other.

Regardless of the actual weighting factor and the ways a flow's trajectory is computed, three different kinds of flows are identified to occur in a manufacturing environment. The importance of these flows to a certain production facility may be different and it is necessary to look at each of them, as well as decide on a proper weighting when combining them into the final flow matrix which can be used to parameterize the problem model. These different kinds are:

- Sequential process flows
- Parallel process flows
- Material flows

2.1. Sequential Process Flows

These flows occur whenever there is a transition from one operation to the next. The dependency in the operations is interpreted as a flow from the current operation to its successor. In the case of multiple successors, there are several possibilities: The flows could be duplicated and thus passed to each of the successor, or split according to some distribution. If the assumption is true that there are indeed materials flowing between the different resources that execute these operations, then this kind of matrix may be very relevant for the problem model.

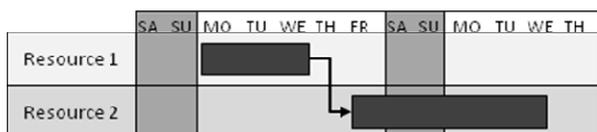


Figure 1 Example of a sequential process flow from Resource 1 to Resource 2

2.2. Parallel Process Flows

In manufacturing scenarios there is also a degree of parallelism in the operations. It happens that when a

product is assembled two operations are applied to it in parallel and thus there is a need for coordination and frequently a benefit in efficiency if those resources are located physically closer to each other.

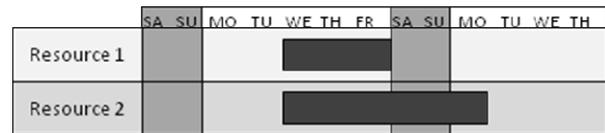


Figure 2 Example of a parallel process flow between Resource 1 and Resource 2

2.3. Material Flows

Material flows occur when an operation needs certain materials from the warehouse or a buffer location and also when the job has finished and produced a number of resources. These resources are then stored again in the warehouse or in the distribution center. In some cases the materials for a sequence of operation are requested together with the first operation and passed through to the others, in some cases these are requested within a sequence. For production environments that are served mostly by warehouses the material flow does not represent inter-facility flows to a very large degree.

Thus for realistic results a combination of these three different flow types has to be considered for optimizing an arrangement.

3. FACILITY LAYOUT PROBLEM

This problem was introduced as the Machine Placement Problem (MPP) in (Beham, Kofler, Wagner, and Affenzeller 2009) and has since changed slightly, as well as it has been extended with more real-world constraints. To describe the problem briefly: It consists of arranging a set of rectangular shapes \mathbf{R} on a flat surface \mathbf{G} such that they lie completely within a boundary polygon \mathbf{P} with p_i being the points of the polygon. The polygon is constructed by connecting each point with the next in sequence and finally the last point with the first. The problem further contains the set \mathbf{B} of fixed blocks, which are immobile locations in the layout. The set \mathbf{L} then describes the layout such that $\mathbf{L} = \mathbf{R} \cup \mathbf{B}$. Each shape in \mathbf{R} represents a machine or work center and is specified by the location of the center coordinates, the dimensions of the rectangle, and a number denoting the rotation in 90° intervals. Each shape in \mathbf{B} is specified by the lower left and upper right points. Finally the matrix \mathbf{F} that specifies the flow strength is given as a $N \times N$ matrix with $N = |\mathbf{R}|$. Elements of this matrix are called f_{ij} and denote the strength of the flow from i to j .

There are several constraints regarding the distance between shapes: Some shapes must be within a specified distance to other shapes, some must maintain a minimum distance to other shapes, and some may have even both a minimum and a maximum distance. The shapes itself are also constrained with bounds on their aspect ratio, e.g. a shape may not be stretched to the very extreme, or not stretched at all.

A solution to this problem specifies the location as x and y position, dimension as width and area, and rotation of each shape. The solution thus can be encoded in the form of multiple vectors of integer values. Two vectors \vec{x}, \vec{y} encode the location on the plane, one vector \vec{w} denotes the width (the height is automatically calculated given that the area A_i of each rectangle remains constant) and the last vector denotes the rotation state $\vec{\omega}$.

The evaluation function computes the layout from this solution vector and first calculates the distance matrix \mathbf{D} with elements d_{ij} between all shapes $i, j \in \mathbf{R}$. Each d_{ij} represents the shortest Manhattan-distance between the rectangles' edges. The main fitness characteristic, the flow-distance-fitness Q_{flow} can then be given as

$$Q_{\text{flow}} = \sum_{i=1}^N \sum_{j=1}^N d_{ij} * f_{ij}$$

The second fitness characteristic, the relayouting costs Q_{relayout} represent the cost of transforming the initial layout that can be defined by the user into the optimized layout. For this purpose each shape $i \in \mathbf{R}$ can be attributed with a movement cost mm_i that depends on the distance that the shape is moved as well as a fixed cost ms_i , e.g. for packing and unpacking or calibration. For this purpose a vector of transition distances t_i is calculated that contains the Manhattan-distance between initial and optimized location.

$$Q_{\text{relayout}} = \sum_{i=1}^N x_i * (ms_i + mm_i * t_i)$$

Where x_i is a decision variable that is 1 if $t_i > 0$ and 0 otherwise. If the shape is in a different rotation state Q_{relayout} is added half the area multiplied by the Manhattan-distance to the new location. To adhere to the constraints during the optimization a soft constrained approach was used and infeasible solutions were penalized, by adding a penalty value that depends on the strength of the constraint violation. This penalty guides the optimization strategy in creating feasible solutions.

The first penalty C_{overlap} deals with the problem of creating layouts with overlapping elements. This can occur frequently as an element's location is modified in a manner that does not consider feasibility, by simply adding a value to the coordinates or swapping the location of two elements. The penalty is computed according to

$$C_{\text{overlap}} = \sum_{i \in \mathbf{L}} \sum_{j \in \mathbf{L}, j \neq i} \frac{A_{ij}}{2}$$

where A_{ij} is the intersecting area between shape $i, j \in \mathbf{L}$ or zero in case there is either no overlap or both $i, j \in \mathbf{B}$. The second penalty C_{boundary} puts a penalty on shapes that lie at least partly outside \mathbf{P} by summing all areas that fall beyond the boundary. In case the whole shape is outside the bounds, the penalty for that shape is further increased by adding its area times the closest

distance to the boundary. Let $\mathbf{O}_p \subset \mathbf{R}$ be the set of all shapes that are partially outside, $\mathbf{O}_f \subset \mathbf{R}$ be the set of all shapes that are fully outside, $A_{\text{outside}}(i)$ with $i \in \mathbf{R}$ be the area of the shape minus the intersecting area with \mathbf{P} , and $d(i, \mathbf{P})$ a distance function specifying the distance between the bounds of shape $i \in \mathbf{R}$ and \mathbf{P} , the constraint can be formulated as

$$C_{\text{partial}} = \sum_{i \in \mathbf{O}_p} A_{\text{outside}}(i)$$

$$C_{\text{full}} = \sum_{i \in \mathbf{O}_f} A_i * d(i, \mathbf{P})$$

$$C_{\text{boundary}} = C_{\text{partial}} + C_{\text{full}}$$

There is also a constraint violation when the distance of two shapes is smaller or larger than the bounds specified on their distance. Let $dmax_{ij}$ be the maximum allowed distance and $dmin_{ij}$ be the minimum required distance between shape i and j , the constraint C_{distance} can be formulated as

$$C_{\text{distance}} = \frac{1}{2} * \sum_{i \in \mathbf{L}} \sum_{j \in \mathbf{L}, j \neq i} \max(dmin_{ij} - d_{ij}, 0) + \max(d_{ij} - dmax_{ij}, 0)$$

Finally the last constraint adds a penalty to solutions in which the aspect ratio of the shape is outside the given bounds. Let $amin_i$ be the minimum allowed aspect ratio, $amax_i$ be the maximum allowed aspect ratio of shape i , and a_i be the aspect ratio (width / height) of shape i , the constraint C_{aspect} can be formulated as

$$C_{\text{aspect}} = \sum_{i=1}^N \max(amin_i - a_i, 0) + \max(a_i - amax_i, 0)$$

The fitness value is then computed as weighted sum of the qualities and constraints as formulated by

$$\text{fitness} = \alpha_1 * Q_{\text{flow}} + \alpha_2 * Q_{\text{relayout}} + \alpha_3 * C_{\text{overlap}} + \alpha_4 * C_{\text{boundary}} + \alpha_5 * C_{\text{distance}} + \alpha_6 * C_{\text{aspect}}$$

3.1. Optimization Scenario

In the optimization scenario that we are considering a small part of the facility is examined. Naturally, the data does not show the true situation of the real world scenario, but has been modified, such that the main characteristics are still present, but that the results cannot be traced back to the actual facility. The flows have been calculated through the simulation as described in Section 2.

The optimization scenario can be seen graphically in Figure 3.

4. SIMULATED ANNEALING

In (Beham, Kofler, Wagner, and Affenzeller 2009) it was reported that simulated annealing (SA) was able to perform best when compared with a genetic algorithm and an evolution strategy. In this work the previously described algorithm was applied again.

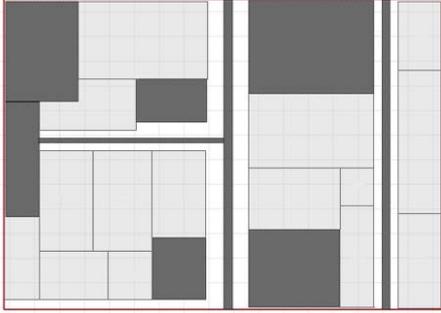


Figure 3 Image of the problem scenario with fixed blocks marked as dark gray and movable shapes in light gray.

Simulated annealing (SA) is among the oldest metaheuristics which offered an explicit strategy to escape from local optima. (Kirkpatrick 1983) was the first to term the algorithm simulated annealing in analogy to the annealing process in metallurgy, and successfully applied the algorithm to optimize computer chip design and the traveling salesman problem. SA employs the temperature as a simple control parameter that guides the search and balances phases of diversification and intensification. As the algorithm proceeds, a step-wise reduction of the temperature focuses the search on a promising region of the solution space, which eventually leads to convergence. The algorithm's performance depends largely on the initial temperature as well as on the cooling scheme. If the temperature drops too quickly the search might get stuck in a worse local optimum; on the other hand, if the cooling is too slow, the algorithm might not have converged when it reaches the stop criterion, which might be for example a maximum number of evaluated solutions. A typical annealing scheme is multiplicative annealing which decreases the temperature in an exponentially shaped curve.

4.1. Problem representation and operators

As has been mentioned in Section 3 the solution representation consists of several vectors. The x and y locations of each facility are given in an array of integer values each, the rotation state is given in another integer array, and finally the width is given in the fourth integer array. The solution thus consists of 4 integer arrays which are modified by several different operations.

- AdditiveNormalManipulation
- AdditiveNormalSingleManipulation
- SwapManipulation
- UniformManipulation

AdditiveNormalManipulation adds a normal distributed random variable with $\mu=0$ and $\sigma=2$ to each value in the vectors. AdditiveNormalSingleManipulation is a variant where only one shape is changed. The random value is rounded to the nearest integer before it is added. SwapManipulation swaps the indices of two randomly selected positions in the arrays.

UniformManipulation sets the values to randomly selected values within the respective boundaries.

These manipulating operations allow the solution to be modified in small ways by moving the shapes a little bit on the plane, exchange two shapes in their locations, or randomly place the shapes in a given area.

The parameters of the simulated annealing heuristic were set as given in Table 1.

Table 1 Parameters for SA

Parameter	Value
Iterations	50000
Temperature	200000
Annealing Factor	0.9996
Inner Iterations	100
α	(1, 0, 10000000, 10000000, 0, 100000000)

5. RESULTS AND CONCLUSIONS

The optimization of the problem leads to interesting conclusions regarding the rearrangement of the layout as can be seen in Figure 4. The layouts solved by the current model do not lead to immediate practical layouts, there are still a number of factors to consider and extend in the current model so that the practical relevance of the solution becomes higher, nevertheless the relationship between the machines becomes obvious and the optimized placement is a good start for the human planner to begin redesigning the layout.

In this paper a method was introduced to simulate process and material flows between facilities directly from the ERP data which can be computed in automated fashion for any given situation. There is no need to specify flow strengths or calculate them by hand. The problem model was described in more detail with the quality and constraint criteria as well as an optimization procedure to derive improved layouts.

In future work we would further extend the problem model to include pathways as well as define infrastructure endpoints in more details. There is still some work necessary regarding the fitness function, the optimizer frequently violates the aspect ratio constraint as it can reach even better, but infeasible solutions. The penalty regarding the aspect ratio is likely too small in contrast to the others.

ACKNOWLEDGMENTS

The work described in this paper was done within the Josef Ressel Centre for Heuristic Optimization *Heureka!* sponsored by the Austrian Research Promotion Agency (FFG).

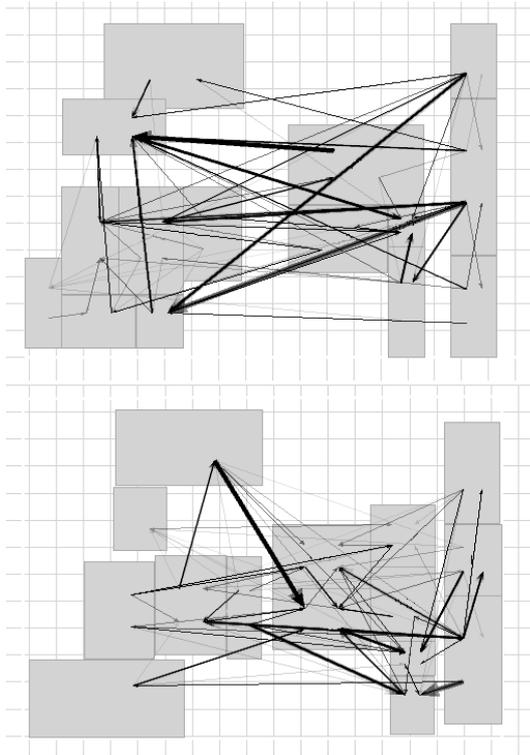


Figure 4 Original (top) and optimized (bottom) layout of the facilities under consideration with arrows marking the flow strength. The thicker and darker an arrow is the stronger the flow. The fixed blocks are not present in this figure.

REFERENCES

- Beham A., Kofler M., Wagner S., and Affenzeller M., 2009. Coupling Simulation with HeuristicLab to Solve Facility Layout Problems. *Proceedings of the 2009 Winter Simulation Conference*, pp. 2205-2217. December 13-16, Austin (Texas, USA).
- Benjafaar S., 2002. Modeling and Analysis of Congestion in the Design of Facility Layouts. *Management Science*, 48, pp. 679-704.
- Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., 1983. Optimization by simulated annealing. *Science*, 220, pp. 671-680.
- McKendall, A.R., and Hakobyan, A., 2009. Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research*, 201, pp. 171-182.
- McKendall, A.R., Shang, J., and Kuppasamy, S., 2006. Simulated annealing heuristics for the dynamic facility layout problem. *Computers & Operations Research*, 33, pp. 2431-2444.
- Scholz, D., Jaehn, F., and Junker, A., 2010. Extensions to STaTS for practical applications of the facility layout problem. *European Journal of Operational Research*, 204 (3), pp. 463-472.
- Scholz, D., Petrick, A., and Domschke, W., 2009. STaTS: A Slicing Tree and Tabu Search based heuristic for the unequal area facility layout problem. *European Journal of Operational Research*, 197, pp. 166-178.

AUTHORS BIOGRAPHY



ANDREAS BEHAM received his MSc in computer science in 2007 from Johannes Kepler University (JKU) Linz, Austria. His research interests include heuristic optimization methods and simulation-based as well as combinatorial optimization. Currently he is a research associate at the Research Center Hagenberg of the Upper Austria University of Applied Sciences (Campus Hagenberg).



MONIKA KOFLER studied Medical Software Engineering at the Upper Austrian University of Applied Sciences, Campus Hagenberg, Austria, from which she received her diploma's degree in 2006. She is currently employed as a research associate at the Research Center Hagenberg and pursues her PhD in engineering sciences at the Johannes Kepler University Linz, Austria.



STEFAN WAGNER received his MSc in computer science in 2004 and his PhD in engineering sciences in 2009, both from Johannes Kepler University (JKU) Linz, Austria; he is professor at the Upper Austrian University of Applied Sciences (Campus Hagenberg). Dr. Wagner's research interests include evolutionary computation and heuristic optimization, theory and application of genetic algorithms, machine learning and software development.



MICHAEL AFFENZELLER has published several papers, journal articles and books dealing with theoretical and practical aspects of evolutionary computation, genetic algorithms, and meta-heuristics in general. In 2001 he received his PhD in engineering sciences and in 2004 he received his habilitation in applied systems engineering, both from the Johannes Kepler University of Linz, Austria. Michael Affenzeller is professor at the Upper Austria University of Applied Sciences, Campus Hagenberg, and head of the Josef Ressel Center *Heureka!* at Hagenberg.



WALTER PUCHNER works for ROSENBAUER INTERNATIONAL AG since 1978. He started his career as machinist apprentice, after which he pursued a master and worked in toolmaking and assembly. Since 1989 he coordinates special projects, among them the extension and adaption of the company's infrastructure.