Abstract. In concrete precasting plants a gantry robot is used to plot lines for casting concrete elements like walls on a pallet. The order of the plot lines is optimized by an evolutionary algorithm. Its implementation was a part of a project of the European Union to reduce the cycle time of the gantry robot movements. The algorithm and its implementation are described briefly.

The Problem

There are concrete precasting plants to produce concrete elements with the help of industrial automation, see /1/. The base of the automation is the use of a gantry robot (so called the formwork robot), which is controlled by NC-based software tools. The position data is generated from a CAD-system, which calculates the robot move targets by geometrical data and converts them into NC-code. This code is directly
downloaded to the robot control unit and performed after a process dependent time delay of about 15 to 45 minutes.

On a metal base plate (pallet) the outer mould of the concrete element is set by using metal bars and magnetic blocks (standard elements) or bars made of polystyrene (which are implemented manually) for fixing. The profiles are the boundary lines for the outline of walls, and they are positioned on pallets of a bright of about 4.5m and a length of 11m. After positioning the profiles the pallets move under a silo, and the concrete is filled into the moulds formed by the profiles. After some other procedures and heating of the concrete, the walls are ready to be transported to the building area.

The position for the profiles installed by humans are plotted on the pallet by the gantry robot. The sequence of the lines to be plotted should be optimized to reduce the sum of distances between the end of one line and the beginning of the next line. The robot moves to plot these lines on the pallet are generated by a program using some heuristics and an optimizing method based on a gradient oriented search. These robot move sequences are loaded down from the overall control to the robot control and executed.

The problem of optimizing the sequence of plot lines is not the same as the famous traveling salesman problem, but the same kind of combinatorial problem. If you have only one line to plot, the number of possibilities to plot the line by the robot is two. If there are two lines to plot, the number of possibilities is 8, i.e. there are 4 start points of lines the robot can make its first move and then it can continue with two possibilities, see following figure. In general, the number of combinations of different ways to move between the lines is

\[ K = 2 \times 4 \times 8 \times \ldots \times (2^n) \]

where \( K \) is the number of possibilities and \( n \) is the number of lines to plot. Of course, because the direction of a robot move doesn’t care, the number has to divide by two for real application. For example, the number of possibilities for 10 lines is

\[ K = 2 \times 4 \times 6 \times 8 \times 10 \times 12 \times 14 \times 16 \times 18 \times 20 = 3,715,891,200, \]

and for 15 lines there are \( 4.285 \times 10^{16} \) combinations possible. Therefore it is not practicable to calculate all possibilities for finding the global optimum. As a solution an evolutionary algorithm will be applied.

![Diagram showing start point and lines with two possibilities to continue]
The Evolutionary Algorithm GLEAM

Nowadays there exist many different ideas and procedures about Genetic Algorithms and Evolutionary Programming, which differ mainly in the representation of genotype, e.g. in \cite{2} or \cite{3}. The proposed algorithm GLEAM (Genetic Learning Algorithm and Methods) is based on evolutionary strategy. The principles of GLEAM including its genetic operators and genotype representation are defined with close connection to the biological evolution (see detailed description in \cite{4}). The genotype of GLEAM consists of a number of so called „actions”, which represents one step of a plan to be executed. Here an action describes where to continue the robot move, e.g. “continue with line number x by the line start point 1 (or 2)”. Because GLEAM was designed for different applications, the kind of action is always “continue robot move”, but for other applications there are different kinds of actions, e.g. “move robot joint” or “stop robot joint”. The mentioned line number is related to the line sequence given by the CAD system. The line start point can be of value 1 or 2 for the two ends of a plot line.

The most important improvement of GLEAM is the introduction of the concept of so called „sections”. A section is formed by a (variable) number of actions as a substructure of the plan, i.e. the action chain is partitioned into segments. A segment can be regarded as a chromosome of the biological genetic information. So, one member, that means the action chain representing the plan, consists of segments, and each segment of a number of actions. Such a segment can be regarded as a partial solution of the optimization problem, which can be combined with other (good) partial solutions to perform a much better plan.

There are new defined evolution operators (for mutation and recombination) to be applied to the sections, like delete a segment or move a segment to another place in the action chain (similar to actions). This concept enables the recombination of „good” sub-structures of a problem solving process to speed up the evolution. If two plans are merged by the recombination, the combination of a segment of plan A with a segment of plan B can be treated as a combination of two sub-solutions. If the two sub-structures res. segments are good in single, they both together in the resulting plan will give a much higher fitness value than the fitness of the both parent plans. To come to such a good solution only by mutation of the genes takes probably much longer time. The concept of segments (the „chromosomes”) and actions (the „genes”) was very successful for planning and optimization tasks of dynamic processes like robot moves.
Evaluation and Code Execution

The quality of the resulting robot movement path is measured by the total length of intermediate routes which must be performed to reach the lines to be plotted. This length varies with the task on hand and an effective evaluation comprise the range between the optimal and the worst value as close as possible. This is achieved as follows:

1. Calculate for the start point and the two end points of each line to be plotted the shortest (longest) distance to an end point of all other lines.

2. Sum up the shortest (longest) length of all routes for all lines in xmin and xmax. In case that the robot must return to the origin xmin must be corrected accordingly.

3. These limits form together with the solution x_{s2} delivered by the heuristic algorithm the evaluation formula as shown in the following Figure.

The value of x_{s1} is calculated as follows:

\[ x_{s1} = x_{s2} - (x_{s2} - x_{min}) \ast vpot\_factor \]

where vpot\_factor is a parameter which is adjusted with respect to the so far gained experience. It reflects the expected range of improvement. 20% of the grading range of 0 to 100000 is assigned to the right part of the formula between x_{s2} and xmax while 25% is assigned to the left part between xmin and x_{s1}. So 55% are left for the part which is expected to be of greater interest.

The approach of segmenting the formula is motivated by the multi criteria optimisation usually done by GLEAM and is not really necessary in this case of one criterion. On the other hand it fits better into the internal control mechanisms of the evolutionary process of GLEAM itself.
To test the generated plot line sequence and visualize the robot move sequence, a special software tool was developed, which displays the gantry robot and its moves. The NC-code executed by the simulation program is the same as the input of the real robot control unit, see the following fig.

**Combination of Heuristic and Evolutionary Search Methods**

The convergence process of an evolutionary computation can be speed up drastically, if already existing knowledge is used for the initial population. Thus a modified version of the „look-ahead“-version of a heuristic algorithm was implemented which incorporates some sort of mutation-like disturbance in the construction of the path. The out coming results build up about a quarter of the initial population for the evolution process. The rest is generated randomly. Thus the results of the evolutionary search must be better (or at least equal) than the already found shortest path, regardless whether it was generated by the CAD system or by the heuristic algorithms. Or in other words, the evolutionary process starts where the conventional optimisation ends.

**Implementation and First Empirical Results**

The GLEAM method was implemented for several applications:
- planning collision free moves for industrial robots, see /5/
- generating production plans solving the job-shop-problem, see /6/
- process scheduling in chemical industry, see /7/
The new application to concrete precasting plants required only some adaptations and some implementation parts of the modules for user interface, simulation and evaluation, see following fig.

The GLEAM software tool was implemented in C and it is portable to other platforms, because of its modular implementation structure, which consists of the following main parts:

- **Basic machine**: E.g. initialization, data structure construction and management, error handling
- **Optimization kernel**: Evolution functions for mutation, recombination, population management
- **Simulation and evaluation**: Execution of plans, criterion’s check, restrictions, fitness calculation
- **Overall control**: Management of parameter input, display of results, interrupting simulation or optimization and others
- **User interface**: Criterion’s priority and fitness definition, target definition, simulation visualization, plan description and others

**module to generate a description for the plot line sequence**: generates the code to describe the order of the plot lines

Simulation and evaluation need the main amount of implementation effort to apply the GLEAM implementation to other applications. It means on the other side, that due to the generality of the method a new application requires less implementation costs because the basic machine, optimization kernel and overall control are not much effected.
The input of an additional simulation system (to present the optimization results) consists of NC data records, which are the same as the input of the robot control, generated by the planning system, see following fig. The records describe the robot move destination and the move velocity. The moves can be done by synchronizing the axes in X- and Y-direction. The following picture shows the result of the execution of a control file for the plot procedure on the palette. The view point is defined above the palette.

Figure : Execution and simulation of NC-Code
The GLEAM software was tested by a set of test runs with some job files from the real production with different amount of plot lines. The results are shown in the following table.

<table>
<thead>
<tr>
<th>Job</th>
<th>Line Number</th>
<th>Original Length</th>
<th>Heuristic Result</th>
<th>GLEAM Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>[m]</td>
<td>improvement</td>
<td>[m]</td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>21.32</td>
<td>21.20</td>
<td>99.4 %</td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td>24.77</td>
<td>19.21</td>
<td>77.5 %</td>
</tr>
<tr>
<td>3</td>
<td>38</td>
<td>19.10</td>
<td>20.32</td>
<td>106.4 %</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>35.26</td>
<td>31.18</td>
<td>88.4 %</td>
</tr>
<tr>
<td>5</td>
<td>48</td>
<td>34.35</td>
<td>30.79</td>
<td>89.6 %</td>
</tr>
<tr>
<td>6</td>
<td>229</td>
<td>35.95</td>
<td>25.45</td>
<td>70.8 %</td>
</tr>
</tbody>
</table>

Table 1: Results of 6 runs with different line numbers. The improvement is always calculated with respect to the original length.

As table 1. shows some jobs can be very well solved by the heuristic algorithm, e.g. jobs 5 and 6 while others cannot, like job 1, 2 or 3. It is remarkable that the heuristic can even produce worse results, as job 3 shows.

References