Evolutionary Search for Optimal Combinations of Markers in Clothing Manufacturing

Bogdan Filipič
Department of Intelligent Systems
Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana
Slovenia
bogdan.filipic@ijs.si

Iztok Fister
Mura, European Fashion Design
Plese 2
SI-9000 Murska Sobota
Slovenia
iztok.fister@mura.si

Marjan Mernik
Faculty of Electrical Engineering and Computer Science
University of Maribor
Smetanova 17
SI-2000 Maribor, Slovenia
marjan.mernik@uni-mb.si

ABSTRACT

Optimizing combinations of placements of parts, known as markers, is an important preparatory step in order-based industrial production of clothes. Given a work order in the form of a matrix of pieces in size numbers and designs, the task is to find a list of combinations of size numbers to complete the work order. The outcome of this step influences the number of cut out pieces, the amount of material used in the production phase, and the speed of the work order processing. The optimization task is demanding since a number of factors affect production costs and several conflicting criteria can be involved in marker assessment. We consider minimum number of markers per work order as an optimization criterion and transform the problem into the knapsack problem which is then solved with several variants of an evolutionary algorithm. Numerical experiments are performed on real problem instances from industrial clothes production and the results compare favorably with those produced by the algorithm regularly used in practice.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, constrained optimization*; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Performance, Experimentation

Keywords

marker, layout, clothes production, knapsack problem, evolutionary algorithm, penalty functions, repair algorithm

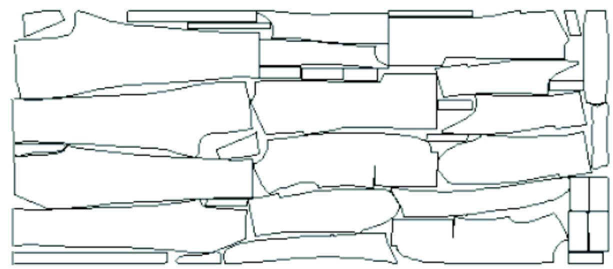

Figure 1: A marker for two suits of different sizes, each consisting of a jacket and pants

1. INTRODUCTION

Optimal placement of two-dimensional shapes is a frequently studied problem with applications in various manufacturing processes, such as production of clothes, shoes and sheet metal products [1, 2, 9]. In clothes production, layouts of pieces or stencils, called markers, are used for cutting parts of clothes from cloth, and finding the tightest non-overlapping placement of parts on a rectangular cloth surface is known as marker making or marker generation. This task is difficult as the parts are irregularly shaped and have three degrees of freedom, two translational and one rotational. Humans need a long training period to become experts in generating high-quality (minimum-waste) markers for a certain type of clothes. Many researchers and companies have worked on computer-aided marker generation, but the resulting algorithms and software tools do not reach the performance of human experts. What is however practised in real-world production is interactive marker generation where the capabilities of computer programs and human experts are combined. An example of a marker generated in this way is shown in Figure 1.

Orders for industrial production of clothes typically come in batches specifying a number of different sizes and designs (colors or patterns). The number of sizes in an order is usually much higher than the number of sizes in a marker. Hence, to accomplish an order, appropriate markers need to be generated first. A question arising at this point is how many sizes and which of them should appear in each marker. It turns out that besides the optimization of each marker with respect to the placement of parts, the preceding phase of combining the sizes into markers is an optimization task, too. Unlike in layout optimizaton, where minimum waste is an obvious goal, the optimization criterion for combining the sizes into markers is not so evident. Optimality of a combination of markers can namely be defined in different ways. It can either be the lowest number of markers needed to fulfil a given work order, the combination of markers that results in the shortest time to fulfil the order, or, more generally, the combination of markers minimizing the production costs [4].

In this paper we deal with search for optimal combinations of markers to accomplish orders in clothes production, and not with optimal placement of parts within the markers themselves. In this context, the term marker optimization should be understood as search for combinations with minimum number of markers per work order. Since the orders in real-world production are too complex for the task to be solved in one part, we propose to solve it in iterations, starting with the original order, finding a marker that covers the highest number of pieces possible, applying it to the order, and repeating the procedure on the remaining pieces until the entire order is covered. This is actually a greedy heuristic approach ,

performing locally at the level of an individual marker. The rationale behind it is that pursuing the maximum coverage criterion at each step could minimize the number of markers needed for the entire work order. We show the problem of finding an appropriate marker at each step can be transformed into the knapsack problem and solved by an evolutionary algorithm. We tested the approach on a set of orders from industrial clothing production and the results show this approach outperforms the deterministic optimization of marker combinations used in practice.

The paper is further organized as follows. The problem is first defined formally and an illustrative example is provided. A transformation into the knapsack problem is then described. Next, an evolutionary algorithm with several variants is proposed for solving the optimization task. It is tested on real-world problem instances and its results are compared with those produced by other algorithms. A summary of the results and directions for further improvement of the approach are given in the conclusion.

2. THE PROBLEM OF FINDING THE OPTIMUM COMBINATION OF MARKERS

2.1 Formal Definition

A work order is given in the form of a matrix A with elements a_{ij} and dimension $n \times m$, where n is the number of sizes and m the number of designs of clothes to be produced. The matrix elements a_{ij} are integers representing the number of pieces of each size and design. The objective is to find vectors $y^{(k)} = (y_1^{(k)}, y_2^{(k)}, \ldots, y_n^{(k)})$ with elements $y_i^{(k)} \in [lb..ub] + \{0\}$, $i = 1..n$, where lb and ub are the minimum and maximum numbers of equal sizes, respectively, and vectors of layers $b^{(k)} = (b_1^{(k)}, b_2^{(k)}, \ldots, b_m^{(k)})$, such that

$$c_{ij} = \sum_{k=1}^{r} y_i^{(k)} b_j^{(k)} \geq a_{ij} \qquad (1)$$

The matrix C of elements c_{ij} represents the maximum coverage of the work order A. A is solved in r steps by applying vectors $y^{(k)}$ that denote markers, and vectors of layers $b^{(k)}$, obtained as

$$b_j^{(k)} = \min\left(\frac{a_{ij}^{(k)}}{y_i^{(k)}}\right), \quad i = 1..n \wedge y_i^{(k)} \neq 0, \quad j = 1..m \qquad (2)$$

In each step, the value

$$f(y^{(k)}) = \sum_{i=1}^{n} y_i^{(k)} \sum_{j=1}^{m} b_j^{(k)} \qquad (3)$$

is to be maximal and the requirement

$$\sum_{i=1}^{n} y_i^{(k)} = M^{(k)} \qquad (4)$$

needs to be fulfilled, where $M^{(k)}$ is the maximum number of sizes in the marker. In a real production process, $M^{(k)}$ changes dynamically depending on the type and properties of the current marker. Empirically defined funtions or tables are usually used for this purpose.

If for each size in the matrix A the number of pieces is calculated as

$$s_i^{(k)} = \sum_{j=1}^{m} a_{ij}^{(k)}, \quad j = 1..n \qquad (5)$$

the task can be represented by the following scheme:

$$\begin{array}{rcll} Y & = & [\ y_1 \cdots\cdots\cdots y_n\ ] & B \\ A & = & \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} & \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \\ S & = & [\ s_1 \cdots\cdots\cdots s_n\ ] & \sum_{i=1}^{n} s_i \end{array}$$

2.2 A Practical Example

Suppose the following work order with five sizes, two designs and the total number of pieces equal to 86 is given:

$$\begin{array}{rcc} \text{Sizes}: & [\ 40\quad 42\quad 44\quad 46\quad 48\ ] & \text{Colors}: \\ A = & \begin{bmatrix} 12 & 14 & 12 & 6 & 2 \\ 9 & 12 & 10 & 6 & 3 \end{bmatrix} & \begin{bmatrix} \text{red} \\ \text{blue} \end{bmatrix} \\ S = & [\ 21\quad 26\quad 22\quad 12\quad 5\ ] & 86 \end{array}$$

Let us find a vector Y such that it results in the maximum number of pieces. Assume the maximum number of sizes in the marker M is 4, and consider, for example, the vector $Y = [1, 1, 0, 1, 1]$. Calculating the vector B according to Equation (2) we obtain

$$b_1 = \min\left(\frac{12}{1}, \frac{14}{1}, \frac{6}{1}, \frac{2}{1}\right) = 2$$

and

$$b_2 = \min\left(\frac{9}{1}, \frac{12}{1}, \frac{6}{1}, \frac{3}{1}\right) = 3$$

and therefore

$$\begin{array}{rcc} Y = & [\ 1\quad 1\quad 0\quad 1\quad 1\ ] & B \\ A = & \begin{bmatrix} 12 & 14 & 12 & 6 & 2 \\ 9 & 12 & 10 & 6 & 3 \end{bmatrix} & \begin{bmatrix} 2 \\ 3 \end{bmatrix} \\ S_Y = & [\ 5\quad 5\quad 0\quad 5\quad 5\ ] & 5 \end{array}$$

Vector S_Y denotes the number of pieces of each size eliminated from the work order. According to Equation (3), the total number of pieces eliminated is $\sum_{i=1}^{n} y_i \sum_{j=1}^{m} b_j = 4 \cdot 5 = 20$ or $20/86 = 23.3\%$. This solution is far from optimum, since the size 48, if eliminated entirely, contributes only five pieces. However, if this size is not included, we obtain

$$\begin{array}{rcc} Y = & [\ 1 \quad 1 \quad 1 \quad 1 \quad 0\ ] & B \\ A = & \begin{bmatrix} 12 & 14 & 12 & 6 & 2 \\ 9 & 12 & 10 & 6 & 3 \end{bmatrix} & \begin{bmatrix} 6 \\ 6 \end{bmatrix} \\ S_Y = & [\ 12 \quad 12 \quad 12 \quad 12 \quad 0\ ] & 12 \end{array}$$

and the number of eliminated pieces is $4 \cdot 12 = 48$ or $48/86 = 55.8\%$. In this solution, which is evidently much better, the number of pieces is limited by the size 46. Now suppose the number of equal sizes ub in the marker is 2 and consider the vector $Y = [1, 2, 1, 0, 0]$, where the size 42 is applied twice. It gives

$$\begin{array}{rcc} Y = & [\ 1 \quad 2 \quad 1 \quad 0 \quad 0\ ] & B \\ A = & \begin{bmatrix} 12 & 14 & 12 & 6 & 2 \\ 9 & 12 & 10 & 6 & 3 \end{bmatrix} & \begin{bmatrix} 7 \\ 6 \end{bmatrix} \\ S_Y = & [\ 13 \quad 13 \quad 13 \quad 0 \quad 0\ ] & 13 \end{array}$$

where the number of eliminated pieces is $4 \cdot 13 = 52$ or $52/86 = 60.2\%$. Say we are satisfied with this result. It actually represents the result of the first step of the optimization procedure and consists of the vectors $y^{(1)}$ and $b^{(1)}$:

$$\begin{array}{rcc} y^{(1)} = & [\ 1 \quad 2 \quad 1 \quad 0 \quad 0\ ] & b^{(1)} \\ A^{(1)} = & \begin{bmatrix} 12 & 14 & 12 & 6 & 2 \\ 9 & 12 & 10 & 6 & 3 \end{bmatrix} & \begin{bmatrix} 7 \\ 6 \end{bmatrix} \\ s^{(1)} = & [\ 13 \quad 13 \quad 13 \quad 0 \quad 0\ ] & 13 \end{array}$$

To continue with the optimization procedure, the original order matrix $A = A^{(1)}$ needs to be reduced by subtracting the numbers of pieces eliminated in step 1, and the resulting matrix $A^{(2)}$ is then used in step 2, which may, for example, yield

$$\begin{array}{rcc} y^{(2)} = & [\ 1 \quad 0 \quad 1 \quad 2 \quad 0\ ] & b^{(2)} \\ A^{(2)} = & \begin{bmatrix} 5 & 0 & 5 & 6 & 2 \\ 3 & 0 & 4 & 6 & 3 \end{bmatrix} & \begin{bmatrix} 3 \\ 3 \end{bmatrix} \\ s^{(2)} = & [\ 8 \quad 0 \quad 9 \quad 12 \quad 5\ ] & 6 \end{array}$$

In this manner the procedure iterates until the elimination of all pieces from the work order.

2.3 Transformation into the Knapsack Problem

In the knapsack problem we are given a set of n objects and a knapsack of capacity C. In addition, the weights of objects are specified by a vector $W = (w_1, ..., w_n)$, and their profits by $P = (p_1, ..., p_n)$. The task is to find a binary vector $X = (x_1, ..., x_n)$ such that

$$\sum_{i=1}^{n} x_i w_i \leq C \tag{6}$$

and the objective function

$$f(X) = \sum_{i=1}^{n} x_i p_i \tag{7}$$

returns the maximum value, i.e. the objects to fill up the knapsack should be chosen in such a way that the profit is maximized and the capacity constraint satisfied. The problem is known to be NP-hard [5].

The problem of finding the optimal marker in a particular step of the search for optimal combination of markers can be transformed into the knapsack problem. The sizes in the work order correspond to the objects in the knapsack problem, the maximum number of sizes in a marker $M^{(k)}$ corresponds to the knapsack capacity C, and the requirement

$$\sum_{i=1}^{n} x_i^{(k)} w_i^{(k)} = M^{(k)} \tag{8}$$

has to be satisfied. The number of sizes in the marker is

$$y_i^{(k)} = x_i^{(k)} w_i^{(k)}, \quad i = 1..n \tag{9}$$

where $x_i^{(k)} \in \{0, 1\}$ and the weights are

$$w_i^{(k)} \in [lb..ub] \tag{10}$$

with lb and ub denoting the minimum and maximum number of equal sizes in a marker, respectively.

Binary vector X denotes the presence of sizes in the marker. The profit is obtained as

$$p_i^{(k)} = \frac{w_i^{(k)}}{s_i^{(k)}} \sum_{j=1}^{m} b_j^{(k)} \tag{11}$$

where s_i is the sum of pieces of the i-th size over all designs given by Equation (5). Furthermore, $B = (b_1^{(k)}, ..., b_m^{(k)})$ is a vector of layers for the application of a solution (marker) $Y = XW$ to the work order. The elements of B are obtained according to Equation (2). From Equations (7) and (11) the objective function can be expressed as

$$f(y^{(k)}) = \sum_{i=1}^{n} \frac{y_i^{(k)}}{s_i^{(k)}} \sum_{j=1}^{m} b_j^{(k)} \tag{12}$$

A valid solution of the problem is any vector $Y = XW$ for which Equation (8) holds. However, the goal is to find the optimal solution Y^* for which the value of the objective function $f(Y^*)$ is maximum.

The objective is therefore to maximize the number of pieces per marker. As a marker only partially solves the work order, a number of markers has to be found to accomplish the given work order. To find an optimal marker in each step, a corresponding instance of the knapsack problem has to be solved. The expectation behind this approach is that maximizing the number of pieces in each step will result in the minimum number of markers needed to complete the work order.

3. AN EVOLUTIONARY ALGORITHM FOR MARKER OPTIMIZATION

In the evolutionary algorithm (EA) for marker optimization, a candidate solution $Y = XW$ is represented as a vector of n integers, where n is the number of sizes in the work order. It stands for a product of a binary vector X and vector of weights W. The initial population of solutions is created by picking the values from $[lb..ub] + \{0\}$ randomly with uniform distribution. As the objective is to search for a marker to eliminate the maximum number of pieces from a work order of the size $n \times m$, Equation (12) is used to evaluate the fitness of candidate solutions.

The algorithm includes tournament selection and two genetic operators to generate candidate solutions: multi-point crossover and uniform mutation.

The algorithm parameters set for experimental runs include the population size, the number of generations, tournament size, the number of crossing sites in multi-point crossover, and probabilities of crossover and mutation.

Two approaches to solving the knapsack problem were applied in evolutionary marker optimization: an EA with penalty function and a repair algorithm [8].

3.1 An Evolutionary Algorithm with Penalty Function

Penalty functions are a way of dealing with invalid solutions in evolutionary algorithms. The idea is to impose selection pressure on invalid solutions by assigning them lower fitness. The approach is expected to gradually lead to valid solutions in the population and then search for the best among them. The fitness function in this approach is determined by subtracting the penalty term from the objective function (12):

$$f(y^{(k)}) = \sum_{i=1}^{n} \frac{y_i^{(k)}}{s_i^{(k)}} \sum_{j=1}^{m} b_j^{(k)} - \mathrm{Pen}(y^{(k)}) \tag{13}$$

The number of sizes in markers for a given problem instance should be equal to the maximum number of sizes, $M^{(k)}$. Solutions with the number of sizes different from $M^{(k)}$ are invalid. The penalty function for invalid solutions can be defined in various ways. We use three types of penalty functions differing in the growth of penalty for violating the number of sizes, i.e. logarithmic, linear and quadratic:

$$\mathrm{Pen}(y^{(k)}) = \log\left(1 + \rho \left|\sum_{i=1}^{n} y_i^{(k)} - M^{(k)}\right|\right) \tag{14}$$

$$\mathrm{Pen}(y^{(k)}) = \rho \left|\sum_{i=1}^{n} y_i^{(k)} - M^{(k)}\right| \tag{15}$$

$$\mathrm{Pen}(y^{(k)}) = \left(\rho \left|\sum_{i=1}^{n} y_i^{(k)} - M^{(k)}\right|\right)^2 \tag{16}$$

where ρ is

$$\rho = \max_{i=1..n}\left(\frac{1}{s_i}\sum_{j=1}^{m} b_j^{(k)}\right) \tag{17}$$

3.2 A Repair Algorithm

In this approach only the solutions with $M^{(k)}$ sizes are evaluated using Equation (12) as a fitness function. However, if a solution is not valid, it is first repaired and then evaluated. Three approaches to repairing the generated vectors are used: heuristic, greedy and random.

Heuristic repair relies on Cauchy-Schwarz inequality [7] for determining the angle θ between two vectors $u, v \in R^n$:

$$\cos\theta \le \frac{uv}{\|u\| \cdot \|v\|} \tag{18}$$

Vectors u and v are

$$u = (a_{i1}, ..., a_{im}), \quad i = 1 \dots n \tag{19}$$

and

$$v = \left(\sum_{i=1}^{n} \frac{a_{i1}}{n}, ..., \sum_{i=1}^{n} \frac{a_{im}}{n}\right) \tag{20}$$

and a vector similarity relation, denoted by $\prec$, is defined as

$$u \prec v \;\Rightarrow\; \cos\theta > \cos\vartheta \tag{21}$$

where $\theta = \angle(u, v)$ and $\vartheta = \angle(w, v)$. Relation (21) defines a heuristic order of putting the objects into the knapsack. In search for an appropriate marker it tries to eliminates sizes such that the maximum reduction of the current work order is expected.

In the greedy method, the order of selecting the objects is based on decreasing sums of sizes in the work order as obtained by Equation (5), and the random method selects the sizes that appear in a solution randomly.

Candidate solutions can either be valid, underestimated or overestimated. Valid solutions need no repair and can be evaluated according to the objective function (12). In underestimated solutions the sum of weights is less than the maximum number of sizes in the marker $M^{(k)}$. Such solutions are repaired by inserting additional sizes. This is done according to either heuristic or greedy approach or by random generation of weights in the range $[lb..ub]$. In overestimated solutions, the sum of weights exceeds $M^{(k)}$. In such cases randomly selected sizes are excluded from the solutions.

4. EXPERIMENTAL RESULTS

The evolutionary algorithm for marker optimization in all its variants was tested on ten work orders from real-world clothes production taken from [3]. The orders differ in complexity and their properties are summarized in Table 1.

Table 1: Properties of the real-world work orders used in numerical experiments: n denotes the number of sizes in a work order, m the number of designs, M maximum number of sizes in a marker, lb minimum number of equal sizes in a marker, ub maximum number of equal sizes in a marker, h_lb minimum number of layers, h_ub maximum number of layers, and P the total number of pieces in a work order

Order No.	n	m	M	lb	ub	h_lb	h_ub	P
1	5	2	4	1	2	4	50	182
2	9	7	8	1	2	5	40	339
3	9	4	14	1	10	5	40	244
4	6	6	4	1	2	17	70	637
5	6	4	4	1	2	4	50	49
6	8	20	4	1	2	1	60	416
7	29	12	2	1	2	10	40	125
8	23	4	2	1	1	10	40	318
9	20	4	2	1	2	10	40	205
10	45	76	4	1	2	4	60	1236

The objective of marker optimization is to maximize the number of cut out pieces for each marker and consequentially minimize the number of markers needed to accomplish a work order. As a stochastic technique EAs generally return different solutions in multiple runs, hence their results have

Table 2: Evaluation of EAs with penalty functions: success rate (SR), and minimum, maximum and average number of generated markers

Order	Logarithmic penalty				Linear penalty				Quadratic penalty			
No.	SR	Min	Max	Avg	SR	Min	Max	Avg	SR	Min	Max	Avg
1	0,97	5	6	5,66	1,00	5	9	6,30	1,00	5	8	6,20
2	1,00	16	17	16,77	1,00	15	18	16,30	1,00	16	16	16,00
3	0,00	–	–	–	0,00	–	–	–	0,00	–	–	–
4	1,00	8	8	8,00	1,00	8	8	8,00	1,00	7	8	7,97
5	0,57	5	6	5,41	1,00	5	6	5,17	1,00	5	6	5,03
6	1,00	19	22	20,70	1,00	17	23	20,37	1,00	15	22	20,23
7	0,00	–	–	–	0,00	–	–	–	0,00	–	–	–
8	0,00	–	–	–	0,00	–	–	–	0,00	–	–	–
9	0,00	–	–	–	0,00	–	–	–	0,00	–	–	–
10	0,00	–	–	–	0,00	–	–	–	0,00	–	–	–

to be analyzed statistically to check for repeatability. The EA for each problem was run 30 times and best, worst and average results recorded. The algorithm parameters were set as follows: population size 50, the number of generations 100 for smaller work orders (No. 1–5 in Table 1) and 500 for larger orders (No. 6–10), tournament size 2, the number of crossing sites n div $10 + 1$, where n is the number of sizes in the work order, crossover probability 0.8, and mutation probability 0.05.

The results of solving the work orders with EAs with penalty functions are shown in Table 2. Success rate (SR) is defined as the proportion of successful algorithm runs and the run is considered successful when it accomplishes the work order. On the accomplished work orders the results of the three penalization methods seem to be very similar and one of them performs significantly better than the other two. It is to be noted, however, that the EAs with penalty functions were particularly unsuccessful on large orders (No. 7–10). An analysis of their populations showed that the algorithms were dealing with invalid solutions where the number of sizes in markers was larger than $M^{(k)}$. In general, reducing the degree of violation and then finding valid solutions through penalties only worked on smaller work orders.

Difficulties with penalty functions can be avoided by involving repair mechanisms that maintain only valid solutions during the evolutionary search. Numerical experiments on the real-world work orders showed the EAs with solution repair were able to solve all problem instances (see Table 3). Again, the results do not indicate any of the algorithm variants as significantly better.

The ten test problems were also solved with a deterministic marker optimization algorithm that heuristically selects the sizes according to the vector similarity relation (21). This algorithm is part of an existing computer-aided marker optimization system regularly used in the textile company that provided the test orders for this study. Numbers of markers generated by this algorithm to complete the work orders are given in Table 4.

Finally, its results were compared with those of solution repairing EAs for the entire test work load. Table 5 shows the numbers of markers needed to accomplish all ten work orders. It can be seen that the EAs outperform the currently used algorithm both on average and in the best solution found. It therefore seems to be a suitable candidate to replace the currently used deterministic algorithm. Its acceptable time complexity additionally makes it appropriate in this respect. To find all markers for accomplishing the most complex work order (No. 10) it takes a few minutes on an up-to-date desktop computer.

Table 4: Performance of the deterministic algorithm used at the plant

Order No.	1	2	3	4	5	6	7	8	9	10
Markers	9	17	13	8	6	22	30	24	21	64

Table 5: Statistical evaluation of the total number of markers generated for ten work orders

Algorithm	Min	Max	Avg	StDev
Deterministic	214	214	214,00	0,00
EA with heuristic repair	194	214	203,60	16,58
EA with greedy repair	201	217	207,99	17,47
EA with random repair	197	218	207,86	17,55

5. CONCLUSION

Finding optimal combination of markers is a preparatory step for order-based clothes production that critically affects production costs. It can be considered from the point of view of various criteria and remains a challenging optimization task. In this paper we dealt with a specific problem of finding the minimum number of markers to accomplish a given work order. This problem was transformed into the knapsack problem, and the algorithms for this problem used to maximize the number of pieces with each marker and simultaneously minimize the number of markers.

The contribution of this work is the implementation of several variants of the EA for marker optimization, its application to work orders from real-world environment and comparison of the results by various optimization algorithms. Numerical experiments have confirmed that EAs with solution repair outperform the deterministic algorithm currently in use at the plant and therefore represent a suitable platform for upgrading the existing marker optimization methodology.

Table 3: Evaluation of EAs with solution repair: success rate (SR), and minimum, maximum and average number of generated markers

Order No.	Heuristic repair SR	Min	Max	Avg	Greedy repair SR	Min	Max	Avg	Random repair SR	Min	Max	Avg
1	1,00	5	6	5,70	1,00	6	8	6,47	1,00	6	6	6,00
2	1,00	15	18	16,30	1,00	16	16	16,00	1,00	16	17	16,77
3	1,00	10	14	12,10	1,00	11	14	12,53	1,00	11	13	11,93
4	1,00	8	8	8,00	1,00	8	8	8,00	1,00	8	8	8,00
5	1,00	5	5	5,00	1,00	5	6	5,03	1,00	5	5	5,00
6	1,00	19	22	20,90	1,00	19	22	21,13	1,00	20	22	20,83
7	1,00	30	30	30,00	1,00	30	30	30,00	1,00	29	30	29,43
8	1,00	23	23	23,00	1,00	24	24	24,00	1,00	23	24	23,80
9	1,00	21	22	21,20	1,00	21	21	21,00	1,00	21	22	21,03
10	1,00	58	66	61,40	1,00	61	68	63,83	0,97	58	71	65,07

Future work on this problem will include experimental verification of the algorithms on more complex work orders and application of various optimization criteria. Moreover, a database of the existing markers from previous orders is planned to be utilized in solving new orders and the results compared with the current approach where all the markers are generated in the optimization process.

Acknowledgement

The work presented in the paper was partially supported by the Slovenian Research Agency under Research Programme P2-0209-0106 *Artificial Intelligence and Intelligent Systems*.

6. REFERENCES

[1] A. Albano and G. Sapuppo. Optimal allocation of two-dimensional irregular shapes using heuristic search methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 10 (5): 242–248, 1980.

[2] A. Crispin, P. Clay, G. Taylor, R. Hackney, T. Bayes, and D. Reedman. Genetic algorithm optimisation of part placement using a connection-based coding method. In T. Hendtlass and M. Ali, editors, *Developments in Applied Artificial Intelligence: 15th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2002, Cairns, Australia, June 17-20, 2002, Proceedings*, pages 232–240. Lecture Notes in Artificial Intelligence, Vol. 2358, Springer-Verlag, Berlin, 2002.

[3] Fister, I. *Optimization of Markers in Clothing Industry*. Technical report, University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia, 2003 (in Slovenian).

[4] Fister, I. *Optimization of Markers in Clothes Production with Evolutionary Algorithms*. MSc thesis, University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia, 2004 (in Slovenian).

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-completeness*. W. H. Freeman, New York, 1979.

[6] E. Horowitz and S. Sahni. *Fundamentals of Computer Algorithms*. Computer Science Press, Rockville, MD, 1978.

[7] S. Lipschutz. *Theory and Problems of Linear Algebra*. Schaum's Outline Series, McGraw-Hill, London, 1974.

[8] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, Berlin, 1992.

[9] T. Ono and G. Watanabe, Genetic algorithms for optimal cutting. In D. Dasgupta and Z. Michalewicz, editors, *Evolutionary Algorithms in Engineering Applications*, pages 515–530. Springer-Verlag, Berlin, 1997.