

Evolutionary Optimization of Markers in Clothes Production

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

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

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

Keywords: marker optimization, clothes production, knapsack problem, evolutionary algorithm, empirical study

Received: July 29, 2003

Optimization of markers plays an important role in preparation of order-based industrial production of clothes. Given a matrix of pieces in size numbers and designs, the task is to find a list of combinations of size numbers to accomplish a 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. As numerous factors affect the production costs and several conflicting criteria can be involved in marker assessment, marker optimization is a demanding task. In this paper, minimum number of markers per work order is used as an optimization criterion. Marker optimization is formally defined as a knapsack problem, and an evolutionary algorithm is proposed to solve the task. It is tested on real problem instances from industrial clothes production and compared with several other algorithms. Its results on complex work orders are shown to be superior to those of other tested algorithms.

1 Introduction

The preparatory process for order-based production of clothes consists of four phases: creation, construction, multiplication and combining of markers. In the creation phase, a fashion designer conceives the sketch of a model together with the appropriate materials and designs to be used in production. In the construction phase, a constructor sets the sketch in the basic size and defines the constituent parts, such as sleeves and pockets, and materials, such as lining and buttons. The multiplication phase depends on the sales department which, according to customers' requirements, determines the so-called work order. The work order is a matrix of pieces in size numbers and designs. Once the work order is known, the basic size of the model can be multiplied into additional sizes with respect to the work order. Finally, combining of markers can start where the size numbers should be combined according to a prescription defining the outlook of markers. Depending on the outlook of markers are the number of possible

cut out pieces, the amount of material used in the production phase, and the speed of the work order processing, in particular laying and cutting.

The optimization of markers comes before the phase of their combining. The key question for the optimization procedure is what the optimal combination of markers actually is. Is it the lowest number of markers to accomplish the work order, or the combination that results in the shortest time to fulfil the work order, or would it not be even easier to solve the work order with separate markers in one size only? As the factors influencing the production costs are numerous and several conflicting criteria can be involved in marker assessment, the optimization of markers is a demanding and challenging task.

In the paper, the optimization of markers for clothes production is first defined as the knapsack problem. Although the task is multi-objective, a single objective is treated which is usually considered as the most relevant in practice. This is to minimize the number of markers needed for the work order. An evolutionary algorithm with several

variants is proposed for solving the marker optimization task and tested on real-world problem instances. Its results are compared with those produced by other algorithms, and directions for further improvement of the evolutionary algorithm for marker optimization are given.

2 Problem definition

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

$$\sum_{i=1}^n x_i w_i \leq M \quad (1)$$

and the objective function

$$f(X) = \sum_{i=1}^n x_i p_i \quad (2)$$

returns the maximum value. In other words, the objects to fill up the knapsack should be chosen in such a way that the capacity constraint is satisfied and the profit maximized. The knapsack problem is known to be NP-hard [3]. For solving this problem in practice, various approximation and stochastic algorithms are used.

The marker optimization problem in clothes production can be formally treated as a knapsack problem. Here, 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. Elements a_{ij} are integer values representing the number of pieces of each size and design. The sizes correspond to the objects in the knapsack problem, and the weights are

$$w_i \in [lb..ub] + \{0\}, \quad i = 1..n \quad (3)$$

where lb and ub are the minimum and maximum number of the same sizes in a marker, respectively. The maximum number of sizes in a marker n_M corresponds to the knapsack capacity.

The binary vector X denotes the presence of the sizes in a marker. The profits are obtained as

$$p_i = \frac{w_i}{s_i} \sum_{j=1}^m b_j \quad (4)$$

where s_i is the sum of pieces of the i -th size over all designs in the work order A :

$$s_i = \sum_{j=1}^m a_{ij} \quad (5)$$

and $B = (b_1, \dots, b_m)$ is a vector of layers for the application of a solution (marker) $y(X) = XW$ to the work order A . The elements of the vector B are obtained by

$$b_j = \min\left(\frac{a_{ij}}{w_i}\right), \quad i = 1..n \wedge x_i \neq 0, \quad j = 1..m \quad (6)$$

From Equation (4) it follows that the vectors P and W are highly correlated. A valid solution of the task is any subset $Y \subseteq X$ for which Equation (1) holds. The goal is to find the optimal solution Y^* for which the value of the objective function $f(Y^*)$ is maximum [8].

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 at each stage, an instance of the knapsack problem has to be solved, and the assumption is this will result in the minimum number of markers to complete the work order.

3 An evolutionary algorithm for marker optimization

An evolutionary algorithm (EA) can be used to optimize the markers. EAs are stochastic search and optimization algorithms from the field of evolutionary computation [1] which considers biological evolution as an inspiration for computer problem solving. The key idea is to search for good solutions by means of computer-simulated evolution where candidate solutions compete against each other. Their quality in solving the problem is used as a fitness measure. Low-quality solutions are excluded from the process, while high-quality solutions produce offspring and undergo variation. This procedure runs in iterations (see Fig. 1) until a termination criterion, such as a prescribed number of iterations, is fulfilled.

```

procedure Evolutionary_algorithm;
begin
   $t := 0$ ;
  initialize_population  $P(t)$ ;
  evaluate  $P(t)$ ;
  repeat
     $t := t + 1$ ;
    select  $P(t)$  from  $P(t - 1)$ ;
    variate  $P(t)$ ;
    evaluate  $P(t)$ ;
  until termination_criterion
end;

```

Figure 1: A sketch of an evolutionary algorithm

Despite the lack of theoretical predictions for the quality of the evolved solutions, EAs are more and more often used for practical problem solving in numerical optimization, production scheduling, complex system design and other domains. Their popularity arises from the simplicity and generality of the algorithms and their ability to find near-optimal solutions. To apply an EA to a particular problem, one has to adjust the problem-specific elements of the algorithm: representation of candidate solutions, initialization of the starting population, fitness function to evaluate the

solutions, the operators to variate the solutions, and values of the algorithm parameters.

In the EA for marker optimization in clothes production, a candidate solution $y(X) = XW$ is represented as a vector of n integers, where n is the number of sizes in a work order. It is obtained as a product of a binary vector X and vector of weights W . In the binary vector, $x_i = 0$ denotes that i -th size is not present in a solution, and $x_i = 1$ denotes the presence of the size. The weights in W are generated in the range (3). The starting population of solutions is created randomly with uniform distribution. The objective is to search for the maximum number of pieces to be obtained by a marker on a work order of the size $n \times m$, hence Equation (2) is used to define the fitness of candidate solutions.

The algorithm includes fitness-proportional selection and two traditional genetic operators to variate the solutions during the evolutionary search process: simple (single-point) crossover and uniform bit mutation [4]. The algorithm parameter values set for experimental runs include population size, number of generations and probabilities of crossover and mutation.

Two approaches to solving the knapsack problem were applied in marker optimization: an EA with penalty function and a repair algorithm [7].

3.1 An evolutionary algorithm with penalty function

Penalty functions are a way of dealing with invalid solutions in EAs. The idea is to impose selection pressure to 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 (2):

$$g(Y) = \sum_{i=1}^n x_i p_i - \text{Pen}(Y) \tag{7}$$

where p_i is obtained from Equation (4), and $\text{Pen}(Y) = 0$ for valid solutions Y . The number of sizes in a marker for a given problem instance is equal to the maximum number of sizes n_M prescribed in advance.

Penalty function for invalid solutions can be defined in various ways. In our problem, invalid solutions are those with the number of sizes in a marker less than n_M . We use three types of penalty functions with different growth of penalty for violations, i.e. logarithmic, linear and quadratic:

$$\text{Pen}(Y) = \log \left(1 + \rho \left(\sum_{i=1}^n x_i w_i - n_M \right) \right) \tag{8}$$

$$\text{Pen}(Y) = \rho \left(\sum_{i=1}^n x_i w_i - n_M \right) \tag{9}$$

$$\text{Pen}(Y) = \left(\rho \left(\sum_{i=1}^n x_i w_i - n_M \right) \right)^2 \tag{10}$$

where in all cases ρ is

$$\rho = \max_{i=1..n} \left(\frac{x_i}{s_i} \sum_{j=1}^m b_j \right) \tag{11}$$

which represents the ratio between profit and weight $\frac{p_i}{w_i}$.

3.2 A repair algorithm

In this approach only the solutions with n_M sizes are evaluated using Equation (2) 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 [6] for determining the angle θ between two vectors $u, v \in R^n$:

$$\cos \theta = \frac{uv}{\|u\| \cdot \|v\|} \tag{12}$$

Vectors u and v are defined as

$$u = (a_{i1}, \dots, a_{im}), \quad i = 1..n \tag{13}$$

and

$$v = \left(\sum_{i=1}^n \frac{a_{i1}}{n}, \dots, \sum_{i=1}^n \frac{a_{im}}{n} \right) \tag{14}$$

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

$$u \prec v \Rightarrow \cos \theta > \cos \vartheta \tag{15}$$

where $\theta = \angle(u, v)$ and $\vartheta = \angle(w, v)$. Relation (15) defines the heuristic order of putting the objects into the knapsack.

The order of picking the objects for the greedy method is defined by

$$\frac{s_i}{\sum_{j=1}^n s_j} < \frac{s_{i+1}}{\sum_{j=1}^n s_j} \tag{16}$$

and the random method generates 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 objective function (2). In underestimated solutions the sum of weights is less than the maximum number of sizes in the marker n_M . They are repaired to get valid by inserting additional sizes. This is done either by random generation of weights in the range $[lb..ub]$ or according to relation (15) or (16). In overestimated solutions, the sum of weights exceeds n_M . In such cases randomly selected sizes are excluded from the solutions.

4 Experiments and results

The evolutionary algorithm for marker optimization was tested on ten work orders from industrial clothes production taken from [2]. The orders differ in complexity and their characteristics are summarized in Table 1.

Table 1: Characteristics of the real-world work orders used in testing: n denotes the number of sizes in a work order, m the number of designs, n_M maximum number of sizes in a marker, lb minimum number of the same sizes in a marker, ub maximum number of the same sizes in a marker, h_lb minimum number of layers, h_ub maximum number of layers, and k the total number of pieces in a work order

No.	n	m	n_M	lb	ub	h_lb	h_ub	k
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 consequently 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 to be analyzed statistically to check for repeatability. The EA for each problem was executed ten times, and best, worst and average results recorded. The algorithm parameters were set as follows: population size 20, crossover probability 0.8, mutation probability 0.05, and the number of generations 50 for smaller work orders (No. 1–5 in Table 1) and 100 for larger orders (No. 6–10).

An example of results obtained in solving work order No. 6 by the EA with penalty functions is shown in Table 2. It can be seen that the algorithm variant with the logarithmic penalty function is able to find the best result in a single run and on average, and also the worst result in a single run. On the other hand, linear penalty yields much less dispersed results with lower average value. This performance is typical for most work orders. It is to be noted, however, that the EA with penalties was unable to solve large work orders (No. 7–10). An analysis of the population showed that the algorithm was dealing with invalid solutions where the number of sizes in markers was larger than n_M and the fitness values negative. Reducing the degree of violation and then finding valid solutions only worked on smaller work orders.

Table 2: Number of markers for work order No. 6 found by the evolutionary algorithm with penalty functions

Penalty method	best	worst	average
logarithmic	10	20	13.8
linear	14	18	15.6
quadratic	11	19	15.4

Difficulties with penalty functions can be avoided by applying the repair algorithm that maintains only valid solutions at each step of the evolutionary search. It turns out that this algorithm can solve all ten work orders. Results presented in Table 3 for the work order No. 10 illustrate a typical outcome on larger work orders. Greedy selection of sizes is better than the random approach, while the heuristic selection of sizes gives the best average and individual result for the number of markers needed.

Table 3: Number of markers for work order No. 10 found by the repair algorithm with different approaches to size selection

Size selection	best	worst	average
heuristic	59	64	60.8
greedy	60	69	64.1
random	62	69	66.1

In addition to comparing various EAs on real problems from industrial practice, a comparative study with other marker optimization algorithms was performed. The following four algorithms were tested:

- exhaustive search that checks for all possible solutions of a marker,
- a deterministic algorithm with heuristic selection of sizes according to the vector similarity relation (15),
- an approximation algorithm with greedy selection of sizes [5],
- an evolutionary algorithm with solution repairing and heuristic selection of sizes.

Exhaustive search could only be applied on smaller work orders No. 1–5, while for larger orders it would need more space and time. The deterministic algorithm is currently used for marker optimization in the textile factory that provided the test problems for this study.

The results in terms of the total number of markers to solve the work orders No. 1–5 are shown in Table 4. The results for larger work orders No. 6–10 are given in Table 5.

Table 4: Total number of markers for work orders No. 1–5 found by different optimization algorithms

Algorithm	best	worst	average
exhaustive search	44	44	44.0
deterministic	53	53	53.0
approximation	43	51	46.3
EA (repair)	43	50	46.2

To appropriately interpret these results, one should bear in mind that the algorithms were run for each marker to partially solve a given work order. The resulting numbers of

Table 5: Total number of markers for work orders No. 5–10 found by different optimization algorithms

Algorithm	best	worst	average
deterministic	161	161	161.0
approximation	157	163	160.0
EA (repair)	150	161	153.1

markers confirm that solving the problem with maximum number of pieces at each step does not lead to the optimal solution for the entire work order. Regarding the performance of the algorithms no clear conclusion can be made for small work orders, while for more complex ones the EA outperforms other tested algorithms. It therefore seems to be an appropriate candidate to replace the currently used deterministic algorithm.

5 Conclusion

Optimization 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. A specific problem of finding the minimum number of markers to accomplish a given work order can be defined as the knapsack problem, and the algorithms for this problem used to maximize the number of pieces with each marker and consequently minimize the number of markers.

The key contribution of our work is the implementation of several variants of the EA for marker optimization, its application to real-world problem instances from industrial practice and comparison of the results by various optimization algorithms. The numerical experiments indicate the EA with solution repairing and heuristic selection of sizes outperforms other algorithms on complex work orders.

Future work on this problem will include experimental verification of the algorithms on more complex work orders, application of various optimization criteria, either in the form of a weighted sum or by means of search for Pareto-optimal sets of solutions. Finally, a database of the existing markers from previous orders will be utilized in solving new orders and the results compared with the current approach where all the markers are built in the optimization process.

References

- [1] Bäck, T., Fogel, D. B., Michalewicz, Z. (Eds.) (1997). *Handbook of Evolutionary Computing*, Institute of Physics Publishing, Bristol, Philadelphia, and Oxford University Press, New York, Oxford.
- [2] Fister, I. (2003). *Optimization of markers in clothing industry*, Technical report, University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor (in Slovenian).
- [3] Garey, M. R., Johnson D. S. (1979). *Computers and Intractability, A Guide to the Theory of NP-completeness*. W. H. Freeman, New York.
- [4] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.
- [5] Horowitz, E., Sahni, S. (1978). *Fundamentals of Computer Algorithms*, Computer Science Press, Rockville, MD.
- [6] Lipschutz, S. (1974). *Theory and Problems of Linear Algebra*, Schaum's Outline Series, McGraw-Hill, London.
- [7] Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag, Berlin.
- [8] Robič, B. (2002). *Approximation algorithms*, University of Ljubljana, Faculty of Computer and Information Science, Ljubljana (in Slovenian).