Volume 2 No. 4, April 2012 ISSN 2223-4985

International Journal of Information and Communication Technology Research

http://www.esjournals.org

Hybrid Approach to Optimize Cut Order Plan Solutions in Apparel Manufacturing

R P Abeysooriya[1], T G I Fernando[2]

[1]Lecturer, Department of Textile and Clothing Technology, University of Moratuwa, Sri Lanka
[2]Lecturer, Department of Statistics and Computer Science, University of Sri Jayewardenepura, Sri Lanka

ABSTRACT

The paper examines the combination the conventional heuristic of COP generation and Genetic algorithm (GA) to optimize Cut order plan (COP) solutions in apparel industry. Cut planners in apparel organizations need to decide the cut templates of fabric cutting when the cut order requirement is known. As NP-hard problem with several constrains, COP requires a high speed processing algorithm to find a near optimal solution. This study presents a hybrid type of solution search algorithm to reduce the long execution time of GA based algorithm implemented for COP problem. The suggesting algorithm combined the two search procedures; conventional heuristic and genetic algorithm, to find better solutions for COP. A mask encoding string defined to improve the encoding mechanism of basic GA using conventional heuristic method of COP generation, determined a reduction in population size of the algorithm without changing the convergence power of the algorithm. Experimental results based on several practical cases proved that the proposed hybrid approach lower execution time without changing the searching accuracy given by the GA only method.

Key words: *Genetic Algorithm, Conventional heuristic algorithms, Cut order Plan*

1. INTRODUCTION

1.1. Cut order plan

The cut order plan (COP) is a significant task in apparel production which arranges cut templates to execute the fabric cutting operation [*1*]. A cut template consists of garment qualities to be cut in different garment sizes and number of fabric plies to be laid under that particular cut template [*2*].The term *size ratio* is defined as ratio of number of garments to be selected from garment sizes to create a particular cut template. In COP problem, this will be the primary objective which the size ratios of cut templates will be derived as COP solution by cut planners. As major input, the quantities of garments need to cut in different garment sizes will be entered to the problem. Researches define the COP problem as NP-hard problem when it entails with higher number of garment sizes [*3,4*]. In fact, number of possible solutions is exponentially propagated when number of garment sizes increases in the cut order [*3*].

In addition, researches formulate the COP problems using different objective functions such as minimize the number of cut templates per cut order [*4*], minimize cutting set up cost [*5*], minimize idle time of cutting department [*2*], and minimize the cost of template making, spreading and cutting [*3*]. In solving, most of algorithm followed heuristic based techniques [*3*] until they finds evolutionary search algorithms. As computational techniques, linear integer programming [*5*] and non-linear integer programming [*6*] were also used by several authors though they weren't successful in finding near optimal solution.

1.2 The cut order plan Problem

The main input of the COP problem is the cut order A.

$$A = [a_{ij}]_{mxn}$$

m: Number of garment sizes
n: Number of different fabric types
a_{ij}: Quantities of garments need to be cut in size i and fabric type j.

i = 1, 2,… ,m
j = 1, 2,… ,n

The objective of COP is to determine an optimal set of cut templates. The objective function defines in relation to the number of cut templates required to satisfy the cut order requirement. Cut order planners attempt to derive minimum number of cut templates for a given cut order. Thus the objective function is set to optimize the number of garments included for each cut template *k* in the cut-order and thereby minimize the cost.

$$f\left(Y^{(k)}\right) = max(\sum_{i=1}^{m} y_i^{(k)} \sum_{j=1}^{n} b_j^{(k)})$$

where the size ratio $Y^{(k)} = \left[y_1^{(k)} \quad \dots \quad y_m^{(k)}\right]$

$b_j^{(k)}$: Number of plies that will be cut from the k^{th}marker and $b_j^{(k)} \le b_{max,j}^{(k)}$

International Journal of Information and Communication Technology Research

http://www.esjournals.org

In calculating the parameter $b_{max,j}^{(k)}$, the size-ratio $y_{\mathrm{i}}^{(k)}$, is used according to the following formula;

$$b_{max,j}^{(k)} = min_j\left(\left[\frac{a_{ij}^{(k)}}{y_i^{(k)}}\right]\right)$$

$i = 1,2,\ldots,m. j = 1,2,\ldots,n \qquad k = 1,2,\ldots.$

Generally, the COP problem is subjected following constrains.

1) The maximum number of garments g_{max}allocated for a cut template. This figure is determined as per the length of the fabric cutting table.

$$\sum_{i=1}^{m} c_i^{(k)} y_i^{(k)} \leq g_{max}$$

where

$$c_i^{(k)} = \begin{cases} 0 & if\ Size_i\ is\ not\ used \\ 1 & if\ Size_i\ is\ used \end{cases} \qquad \forall\, i,k$$

2) The size ratio should not generate total garment pieces greater than the cut order requirement.

$y_i^{(k)} \leq \min_j a_{ij}^{(k)} \qquad i = 1,2,\ldots,m. \quad \forall\, j$ where

y_i: Number of times a particular size (size i) appeared in the cut template

1.3 Conventional method of solving COP problem

In conventional approach, the COP problem is solved by calculating the possible size ratio under a prescribed heuristic procedure. The heuristic algorithm starts with suggesting a maximum configuration of possible size ratio that could be allowed to generate the first cut template. The suggested size ratio denoted in $Y_{suggest}^{(k)}$ is predicted according to the minimumnumber of quantity available in the existing cut-order and the constrained explained in section 1.1.

$$Y_{suggest}^{(k)} = \left[\left(\frac{a_{ij}^{(k)}}{\min\limits_j a_{ij}^{(k)}}\right)\right]$$

When algorithms proceeds, the number of the sizes in the modified cut order $A^{(k)}$is gradually being reduced (referexample 01) until the whole cut order is processed.

Numerical example 01:

Cut order

$A^{(1)} = [64 \;\; 94 \;\; 105 \;\; 68 \;\; 25]$

$$Y_{suggest}^{(1)} = \left[\frac{64}{25}\frac{94}{25}\frac{105}{25}\frac{68}{25}\frac{25}{25}\right]$$

$Y_{suggest}^{(1)} = [2 \;\; 3 \;\; 4 \;\; 2 \;\; 1]$

If the maximum number of garments that can be included in the cut template$g_{max} = 4$, then the corresponding size ratio would be$Y^{(1)} = [2 \;\; 0 \;\; 0 \;\; 1 \;\; 1]$and $b_{max,j=1}^{(1)} = 25$ and select $b_j^{(1)} = 25$in way that$b_j^{(1)} \leq b_{max,j}^{(1)}$is satisfied. Likewise, the first cut template is decided based on $Y^{(1)}$with corresponding fabric spread of $b_j^{(1)}$. In addition, the modified cut-order for second cut template is calculated as follows.

$A^{(2)} = [64 - 50 \;\; 94 - 0 \;\; 105 - 0 \;\; 68 - 50 \;\; 25 - 25]$

1.4 Genetic Algorithm

Genetic Algorithm (GA) is a probabilistic search algorithm which follows the natural rules of biological evolution. GA specialises in application of optimization problems is searching optimal or near optimal solutions [*7*,*8*]. Though researchers proved its efficiency in finding quality solutions, sometimes, it may computationally slower reaching optimal or near optimal solution [*9*,*10*]. In GA, each potential solution of the problem is encoded as a chromosome and selected amount of chromosomes will be treated as the initial population of chromosomes. These chromosomes will be processed in a cyclic manner, under the basic operators of GA; reproduction, crossover and mutation. Algorithm updates the chromosomes of the population pool while the evolution cycle proceeds in generation-wise. When the GA process continues for several number of generations (cycles), solutions become healthier and improved if the algorithm searches better fitness valued chromosomes in each generation.

This study introduces a hybrid GA approach to solve the problem of COP generation. Initially, a traditional heuristic method is used to generate several feasible prospective solutions. The aim of including heuristics solutions in this population is to speed up the algorithm execution time by reducing the search space size. The experiments of the study were carried out in two phases. First, the problems were solved by the GA method alone without incorporating any heuristic method. Second, the solutions were generated adjoining the conventional heuristic method with standard GA.

The rest of this paper is organized as follows: Section 2 describes the existing problems of COP generation using evolutionary optimization techniques. Section 3 explains the proposing hybrid approach of traditional heuristic and GA. Section 4 considers the experiments involving the proposed method with major findings. Finally the paper concludes the investigation with prospective research regarding the study.

2. PROBLEM DESCRIPTION

Though researches attempted optimistic heuristic and evolutionary search algorithms to solve the NP hard problems, existing methodologies still cause problems significantly in execution time [*11*,*12*,*13*]. In fact, efficiency of algorithm could be more problematic with more complex forms of COP problems. Most GA procedures require large population to find quality solutions with great effectiveness [*14*]. The proposed study investigates an efficient algorithm equipped with hybrid approach to generate quality solutions with improved efficiency.

3. APPLICATION OF HYBRID APPROACH OF CONVENTIONAL HEURISTIC AND GA

In simple genetic algorithm (SGA), initial population of solutions are generated randomly. In contrast, the propose hybrid structure utilizes the conventional heuristic technique in section 1.3 to reduce the execution time of the overall algorithm. In fact, the suggested size ratio will be acted as *encoding mask string* to determine the possible the size ratios as the chromosomes input for initial population of GA.

The proposed algorithm executes mainly in two steps to find the optimal size ratio of a cut template. Once a size ratio is finalized, the existing cut order will be modified to execute the algorithm to find the optimal size ratio for the second cut template configuration as mentioned in numerical example 01.

Step 1: Initialization and evaluation of possible solutions of the COP.

a) The algorithm begins with generation of suggested size ratio using the conventional heuristic algorithm

b) A chromosome represents a possible solution to the problem. It is encoded as a vector of integer number $Y^{(k)}$. Each solution chromosome isbounded to the encoding mask strings and the constrains.

c) Generation of initial set of chromosomes as per population size (PS).

Algorithm:

$Gen = 1$
$for\ N_{chr} = 1\ to\ PS$
$\quad k = Gen$
$\quad while\ a_{ij}^{(k)} \leq 0 \quad \forall\, i, j$
$\qquad Generate\ y_{suggest,\ i}^{(k)} \quad for\ a_{ij}^{(k)}$
$\qquad for\ i = 1\ to\ m$
$\qquad\quad if\ a_{ij}^{(k)} = 0 \quad \forall j$
$\qquad\qquad y_i^{(k)} = 0$
$\qquad\quad else$
$\qquad\qquad if\ \min_j a_{ij}^{(k)} > y_{suggest,\ i}^{(k)}$
$\qquad\qquad\quad y_i^{(k)} = round\left(\left(y_{suggest,\ i}^{(k)} - 0\right) \times rand()\right)$
$\qquad\qquad else$
$\qquad\qquad\quad y_i^{(k)} = round\left(\left(\min_j a_{ij}^{(k)} - 0\right) \times rand()\right)$
$\qquad\qquad end$
$\qquad\quad end$
$\quad y_{suggest,\ i}^{(k)} = y_{suggest,\ i}^{(k)} - y_i^{(k)}$
end

Step 2: Application of genetic operators to produce new set of chromosomes. At each generation, the algorithm uses the individuals in the current generation to produce the next population.

a. Evaluate each chromosome in the population with respective to the objective function $f(Y^{(k)})$. The evaluation value corresponds to each chromosome is known as fitness value [*15*] which determines the survival ability of the chromosome at each generation when evolution proceeds [*16*].
b. Select parent chromosomes proportional to their fitness value using roulette wheel selection technique [*17*,*18*]. Four chromosomes were selected as mating chromosomes in this study.
c. Generate offspring chromosomes from the parents by crossover operator.

In mating process, some chromosomes are undergone crossover operation at the defined crossover probability $P_{crossover}$[*16*].If a random number corresponds for a mating chromosome is less than the crossover probability [*19*] ($P_{crossover} = 0.6$), that particular chromosome is selected as a chromosomes to be crossover.The study being utilized theuniform order based crossover method with a major modification to exchange genes under the crossover operation [*20*]. Pursuing the standard methodology, a mask string is defined randomly to control the exchange of genes at each generation [*21*]. Instead of recommended gene swapping in uniform technique, genes are exchanged without violating the size ratio constrains mentioned in section 1.2.

As illustrated in figure 1, the solid line arrow indicates in what way a part of child 1 is composed, filling the genes from parent 1. The ***1***'s of the mask string guides to fill the set of genes of child 1 from parent 1.Similarly, rest of genes of child 1are filled from parent 2 making the generating chromosome to be satisfied the constrains of size ratio. In figure 1, when generating child 2, placing the value ***3*** in 2nd gene of parent 1 will not be possible, in such case, the maximum possible value will be placed (a value less than 3) by examining the size ratio constrains as well.

Volume 2 No. 4, April 2012 ISSN 2223-4985

International Journal of Information and Communication Technology Research

http://www.esjournals.org

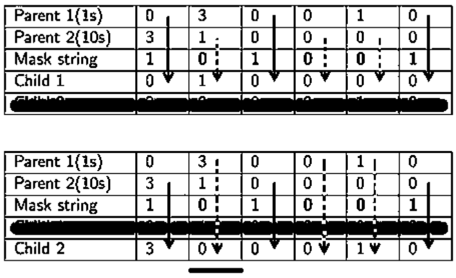

Figure 1: Crossover operation

d. Generate offspring chromosomes by mutation operators.

Number of chromosomes is also subjected to the mutation operation according to the mutation probability$P_{mutation}$to prevent solutions into local stream [*22,23*]. Following the uniform mutation technique [*24*], a gene of the mutating chromosome is selected according the mutation probability$P_{mutation} = 0.1$.The mutation algorithm proceeds along the chromosomes until it finds a gene which corresponding random number is less than$P_{mutation}$. Once a gene with chromosome is selected, the value of that gene is replaced with a uniform a random number between zero and (g_{max} - total of other gene values of the chromosome).

e. Evaluate children chromosomes with respective to the fitness function and retain higher fitted chromosomes in the population. Since proposed algorithm proceeds with fixed number of population, at each generation, low mating parents are removed from the population only when higher fitted children enters to the population.
f. Choose best fitness chromosome as elite and utilize the elite chromosome in the next population. Though elitism strategy is able to improve the performance of convergence [*15*], sometimes, it might tend to be premature convergence if elitism is excessively used [*25*]. However, the hybrid structure proposed in this paper demands a very low degree of elitism due to low selection pressure.

4. RESULTS AND DISCUSSION

4.1 Evaluation of Convergence for Optimization

The proposed hybrid approach can quickly find many feasible solutions for a given set of criteria. Figure 2 and figure 3 compare the average fitness values of 10 chromosomes in each generation, using both simple GA approach [*20*] and hybrid GA approach.

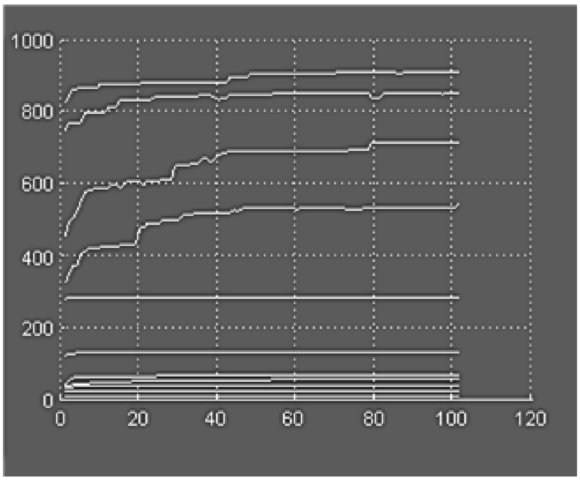

Figure 2: Convergence Performance with Simple GA (Population size = 10)

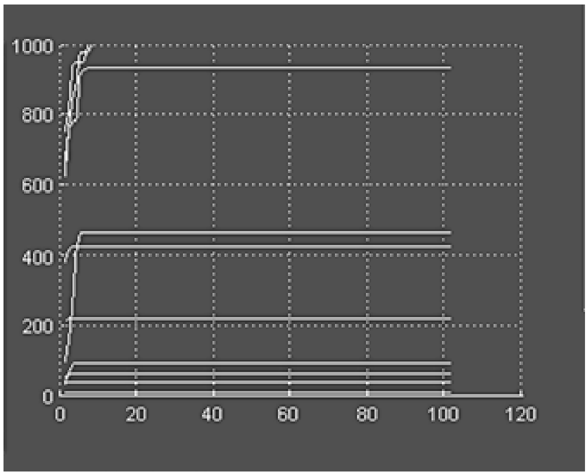

Figure 3: Convergence Performance with hybrid GA (Population size = 10)

Figure 3 evidences that proposed hybrid approach is able to achieve highest fitness values in great efficiency than simple GA alone methodology.

4.2 Performance of Proposed Algorithm

The proposed algorithm was further examined in terms of cost of processing with the help of cutting cost formula derived by Filipic and Fister [4,26] .Solutions obtained for 15 different industry-related COP problems (figure 4), depict that proposed algorithm has more cost advantage than commercially available COP generator and the algorithm based on simple GA. Although there was slight cost improvement in hybrid method than the simple GA method, efficiency has improved in a way that algorithm could allow to reduce the population size without harming the convergence and quality of the solutions. Thus, proposed algorithm provided significant minimization in execution time (figure 5).

Volume 2 No. 4, April 2012 ISSN 2223-4985

International Journal of Information and Communication Technology Research

http://www.esjournals.org

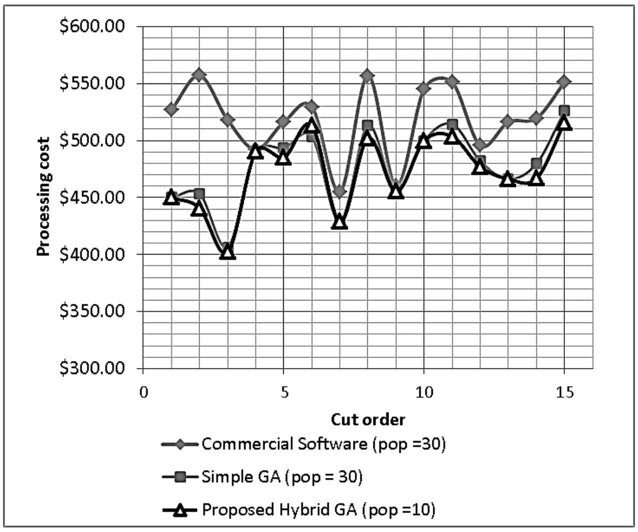

Figure 4: Comparison of cost values in different approaches of optimizing the COP solutions (Generations =100)

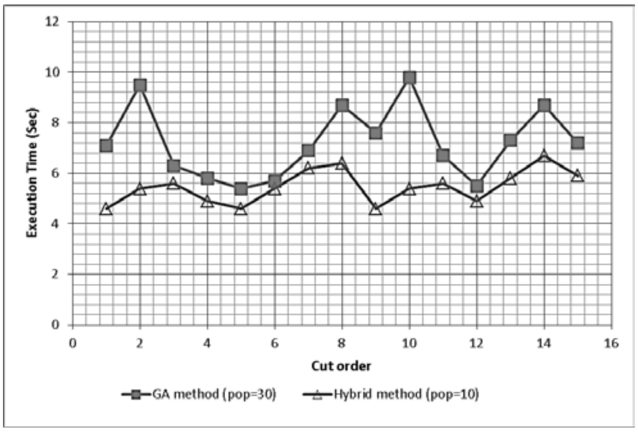

Figure 5: Execution Times of algorithms (Generations = 100)

5. CONCLUSIONS AND SUGGESTIONS

5.1 Conclusion

The proposed methodology is an improvement of simple genetic algorithm procedure used to optimize COP solutions. The newly developed algorithm is executed to search optimized size ratios of the cut templates with short execution time, preserving the quality of searching in simple GA approach. Nevertheless, the algorithm is capable of performing under small population size and smaller number of generations. Results of experiments evident that adjoining conventional heuristic approaches with GA accomplished an efficient searching of COP solutions than the simple GA procedure.

5.2 Future research

The proposed hybrid methodology improved the efficiency of search. However, investigation of other optimization techniques such as simulated annealing, tabu search would provide more effective and efficient solutions when they combined with genetic algorithm, in optimizing COP solutions.

REFERENCES

[1] W Wong and S Leung, "Real-time GA-based rescheduling approach for the pre-sewing stage of an apparel manufacturing process," *International Journal of Advanced Manufacturing Technology*, vol. 25, pp. 180-188, 2004.

[2] W K Wong, C K Chan, and W H Ip, "Optimization of spreading and cutting sequencing model in garment manufacturing," *Computers in Industry*, vol. 43, pp. 1-10, 2000.

[3] C Jacobs-Blecha, J C Ammons, A Schutte, and T Smith, "Cut order planning for apparel manufacturing," *IIE Transactions*, vol. 30, pp. 79-90, 1998.

[4] B Filipic, I Fister, and M Mernik, "Evolutionary search for optimal combinations of markers in clothing manufacturing," in *GECCO*, Washington, 2006.

[5] Z Degraeve and M Vanderbroek, "A Mixed Integer Programming Model for Solving a Cutting Stock Problem in the Fashion Clothing Industry," 1995.

[6] YoungSu Yun, "Hybrid genetic algorithm with adaptive local search scheme," *Computers & Industrial Engineering*, vol. 51, pp. 128–141, 2006.

[7] M Yoshikawa, T Fujino, and H Terai, "A Novel Genetic Algorithm Routing Technique in 3-Dimetional Space," in *Proceedings of the 10th World Multiconference on Systemics, Cybernetics and Informatics*, vol. 1, 2006, pp. 70-75.

[8] H Kanoh and T Nakamura, "Knowledge based genetic algorithm for dynamic route selection," in *Proceedings of Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, vol. 2, 2000, pp. 616-619.

[9] B C Ricardo, Prudencio, and B L Teresa, "Neural Network Hybrid Learning: Genetic Algorithms &Levenberg-Marquardt," in *Principles and practice of constraint programming-CP 2002: 8th International Conference, CP 2002*, Ithaca, NY,

International Journal of Information and Communication Technology Research

http://www.esjournals.org

2002.

[10] R C Correa, A Ferreira, and P Rebreyend, "Scheduling multiprocessor tasks with genetic algorithms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 10, no. 8, pp. 825-837, August 1999.

[11] Imtiaz Ahmad and Muhammad K Dhodhi, "Task assignment using a problem-space genetic algorithm," *Concurrency: Practice and Experience*, vol. 7, no. 5, pp. 411-428, 1995.

[12] M Arakawa, "Hierachy Genetic Algorithm to Solve Multi-Objective Scheduling Problems involving Various Types of Assignments for Parallel Processing Systemsl ," in *Service Robotics and Mechatronics: Selected Papers of the International Conference on Machine Automation ICMA2008*.: Springer, 2009.

[13] Feng-Tse Lin, Cheng-Yan Kao, and Ching-Chi Hsu, "Applying the genetic approach to simulated annealing in solving some NP-hard problems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 6, pp. 1752-1767, 1993.

[14] Wolfgang Banzhaf and Colin R. Reeves, *Foundations of genetic algorithms 5*.: Morgan Kaufmann, 1999.

[15] Sue Ellen Haupt and Randy L. Haupt, *Practical Genetic Algorithms*.: Wiley-Interscience, 2004.

[16] D E Goldberg, *Genetic algorithms in search, optimization & machine learning*.: Addison Wesley, 1997.

[17] Mitchell Melanie, *An Introduction to Genetic Algorithms*, 3rd ed.: The MIT Press, 1998.

[18] UjjawalMaulik, S Bandyopadhyay, and A Mukhopadhyay, *Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics*.: Springer, 2011.

[19] C Karr, *Industrial applications of genetic algorithms*. Boca Raton FL: CRC Press, 1999.

[20] Ranga Prasad Abeysooriya and T G I Fernando, "Canonical Genetic Algorithm To Optimize Cut Order Plan Solutions in Apparel Manufacturing," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 2, pp. 150-154, February 2012.

[21] Mitsuo Gen and Runwei Cheng, *Genetic algorithms and engineering optimization*.: John Wiley & Sons, 2000.

[22] Yeo Su Yuin Melinda, Genetic Algorithm and Acceleration Techniques, 2003.

[23] Dang XuanThao, *Genetic Algorithms and Application in Examination Scheduling*.: GRIN Verlag, 2010.

[24] Masatoshi Sakawa, *Genetic algorithms and fuzzy multi-objective optimization*: Springer, 2002.

[25] Chang WookAhn, *Advances in evolutionary algorithms: theory, design and practice*.: Springer Science & Business, 2006.

[26] I Fister, M Mernik, and B Filipi, "A hybrid self-adaptive evolutionary algorithm for marker optimization in the clothing industry," *Applied Soft Computing*, vol. 10, no. 2, pp. 409–422, 2010.