CZASOPISMO TECHNICZNE M
WYDAWNICTWO POLITECHNIKI KRAKOWSKIEJ

z. 3-M/2008
ISSN 0011-4561
ISSN 1897-6328

SŁAWOMIR ŻAK*

GENETIC ALGORITHM WITH ADDITIONAL PARAMETER FOR 2-DIMENSIONAL IRREGULAR SHAPE CUTTING PROBLEM

ALGORYTM GENETYCZNY Z DODATKOWYM PARAMETREM DO WYKROJU ELEMENTÓW O NIEREGULARNYCH KSZTAŁTACH Z POWIERZCHNI DWUWYMIAROWYCH

A b s t r a c t

The optimum cutting problems occur very frequently in various area of industry and technology. A number of algorithms were proposed, targeting solving particular variants of the aforementioned problems (guillotine cut, non-guillotine cut, etc.). In the article, we proposed and examined a genetic algorithm implementation with an additional parameter, allowing for optimization of the placement of individual elements with irregular shape profile on dimensional surface. The simulation results were presented, examining the impact of the combination of generic operators and the proposed additional parameter on the quality of the obtained results. An example of practical implementation of the examined algorithm in engineering applications is presented, thanks to the integration with the AutoDesk software package – AutoCAD 2007.

Keywords: *2d irregular-shape cutting problem*, *modified genetic algorithm*, *work automation with AutoCAD 2007*

S t r e s z c z e n i e

Problemy optymalnego rozkroju bardzo często występują w różnych dziedzinach techniki. Opracowano liczne algorytmy pozwalające na rozwiązywanie szczegółowych wariantów tych problemów (rozkrój gilotynowy, niegilotynowy itp.). W artykule zaproponowano i przebadano algorytm genetyczny z dodatkowym parametrem pozwalający na optymalizację wykroju elementów o nieregularnych kształtach z powierzchni dwuwymiarowych. Przedstawiono wyniki symulacji, w trakcie których zbadano wpływ doboru standardowych operatorów genetycznych, jak i zaproponowanego dodatkowego parametru, który ma wpływ na jakość uzyskiwanych wyników. Przedstawiono również możliwość praktycznego użycia algorytmu w zastosowaniach inżynierskich dzięki integracji z programem AutoCAD 2007.

Słowa kluczowe: *nieregularny problem rozkroju*, *zmodyfikowany algorytm genetyczny*, *automatyzacja pracy w AutoCAD 2007*

*Sławomir Żak, IV year student, Institute of Applied Informatics, Cracow University of Technology.

1. Introduction

The cutting problem has a number of possible variants, e.g. guillotine, non-guillotine or irregular type cutting problem. In the most general case, it can be characterised in the following way: taking into account a uniform, flat material sheet (e.g. a sheet of paper or metal) with a pre-defined size and a number of elements, we are trying to find their placement in such a way that it minimises the amount of the material shreds resulting from their cutting process.

The research on this particular topic was initiated in the 1940s, when the authors of [1] were considering an optimum of any rectangle into squares. In the following years, the cutting problem was proven to be NP-complete and a number of its variants were presented, often with dedicated solution algorithms.

In the present article we investigate the area minimisation process for the processed material sheet, with a pre-defined height and unlimited length, comprising cutting from it a number of elements with irregular shapes [2]. A universal genetic algorithm with an additional parameter is proposed for this particular problem and is then examined in detail in terms of operational efficiency. The impact of the selection of genetic operators on the quality of the obtained solution was also analysed in detail, targeting the utilisation of the best set of parameters. Additionally, an example of practical implementation of the examined algorithm in engineering applications is presented, thanks to the integration with the AutoDesk software package – AutoCAD 2007.

2. General characteristics of the genetic algorithm employed

Due to the irregular shapes of the cut-out elements, their description is stored in an XML file in a vector form. Such an approach is very convenient when it comes to easier manipulation in terms of shape geometry and additionally allows to define an arbitrary number of drill holes inside of each profile. It comes, however, with a challenge of determining whether two given profiles overlap or not, requiring the utilization of complex techniques for edge collision detection [3]. This particular problem was resolved via rasterizing the element profile and then comparing whether the individual pixels of the two given elements do not overlap. The raster matrix size compression technique was also applied here, featuring grouping individual pixels and storing them as 32 bit wide integer variables [4].

The proposed algorithm utilizes the generic element packing algorithm, commonly referred to as Bottom-Up-Left-Justified [5]. It provides the solution in the form of a sequence of packed elements, starting from the top left-hand corner of the material sheet towards its bottom part, moving gradually in the right direction, providing that the elements do not fit in the given column. Additionally, all the placed elements can be rotated by 0°, 90°, 180°, 270°, providing more points of freedom for the final solution.

The employed element placement algorithm, accounting for the possible, independent rotation of the individual elements, immediately sets aside the application of binary coding scheme, which is typically used in the generic implementations of generic algorithms. That is why the following chromosome structure was used

$$< (X_1;R_1),(X_2;R_2),(X_3;R_3), ...,(X_i;R_i), ...,(X_n;R_n) > \quad (1)$$

where (X_i; R_i) determines the location and rotation angle for the given profile.

The fitness function is hereby defined as a number of longitudinal pixels required to place all the target elements. Thus, the best solutions ought to tend to 0.

3. Genetic algorithm operators employed

Two classic selection operators were implemented and tested against operating efficiency, namely: *roulette type selection* (R) and *tournament selection* (T). In the case of the first method, the chromosomes with the fitness closer to 0 are selected with proportionally higher probability when compared with other elements. In the latter case, the tournament selection creates a population with 10% size of the initial population, containing randomly selected chromosomes, and only then selects a parent element.

The specific structure of the chromosome requires the application of specific crossing operators which will generate no encoding errors. It must be noted that the solution encoded by the chromosomes has a character of permutations with no repetitions. Such limitation is typically imposed on all crossing operators known from the TSP (Travelling Salesman Problem). In the course of our research, several crossing operators were employed, namely: *one-point* (1PX), *two-point* (2PX), *ordered* (DOX) [6], *linear* (LOX) [7], *position oriented* (PBX) [8], with *partial fitting* (PMX) [9], *cyclic* (CX) [10] and *uniform* (EX). All the aforementioned operators process the chromosome as a whole, operating on both the order of placed elements and their rotation angles.

The main task of the linear mutation operators is avoidance of local solution minima. Based on their classic versions, employed mainly with the binary data encoding schemes for chromosomes, namely: *one-point* (1PM), *two-point* (2PM) and *uniform* (EM), new operators were created, which process only part of the chromosome encoded information, e.g. the rotation angle for particular elements. This way the local minima avoidance is assured by the possible rotation of one of the elements by a randomly selected angle, chosen from the predefined set of values (0°, 90°, 180°, 270°).

In the framework of improving the overall operating efficiency of the genetic algorithms, a new genetic operator was proposed and examined. Its operation principle is as follows: with a certain pre-defined probability, part of the solution population (e.g. 10%) is replaced with a completely new set of chromosomes. The eliminated chromosomes are selected at random, which in return provides higher differentiation of the resulting population.

4. Algorithm implementation

The genetic algorithm was implemented in Borland Developer Studio 2006 environment, using C++ language. The code was based on the STL (Standard Template Library) to assure its highest performance. The block diagram of the implemented algorithm is presented in Fig. 1.

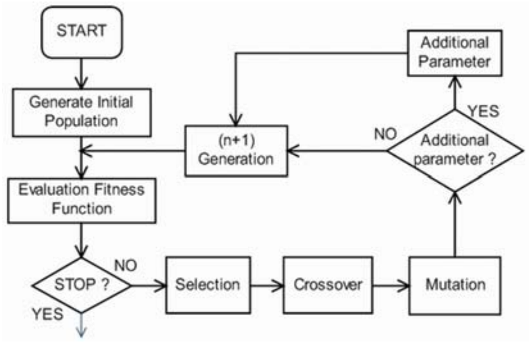

Fig. 1. Block diagram of the implemented algorithm
Rys. 1. Diagram blokowy algorytmu

5. Test results

A number of tests were conducted, examining the impact of various combinations of the genetic operators on the quality of the final results. The additional, proposed parameter was employed in part of the tests for comparative purposes. The said tests were carried out in the following manner (Table 1):

Table 1

Employed combinations of the genetic operators

Test number	Selection	Crossover	Mutation	Test number	Selection	Crossover	Mutation
1	R	1PX	1PM	25	T	1PX	1PM
2	R	1PX	2PM	26	T	1PX	2PM
3	R	1PX	EM	27	T	1PX	EM
4	R	2PX	1PM	28	T	2PX	1PM
5	R	2PX	2PM	29	T	2PX	2PM
6	R	2PX	EM	30	T	2PX	EM
7	R	DOX	1PM	31	T	DOX	1PM
8	R	DOX	2PM	32	T	DOX	2PM
9	R	DOX	EM	33	T	DOX	EM
10	R	LOX	1PM	34	T	LOX	1PM
11	R	LOX	2PM	35	T	LOX	2PM
12	R	LOX	EM	36	T	LOX	EM
13	R	PBX	1PM	37	T	PBX	1PM
14	R	PBX	2PM	38	T	PBX	2PM
15	R	PBX	EM	39	T	PBX	EM
16	R	PMX	1PM	40	T	PMX	1PM
17	R	PMX	2PM	41	T	PMX	2PM
18	R	PMX	EM	42	T	PMX	EM
19	R	CX	1PM	43	T	CX	1PM
20	R	CX	2PM	44	T	CX	2PM
21	R	CX	EM	45	T	CX	EM
22	R	EX	1PM	46	T	EX	1PM
23	R	EX	2PM	47	T	EX	2PM
24	R	EX	EM	48	T	EX	EM

Table 2

Assumed values for control parameters in the algorithm

Parameters	Value
Crossing probability	0,6
Mutation probability	0,05
Population size	30
Number of generations	20
Number of algorithm repetitions	5
Additional parameter (when used)	0,1

Particular parameter values for this implementation of the genetic algorithm were defined a priori, based on the classic examples from literature and were presented in Table 2. The classic form of the genetic algorithm was used, with no adaptive mechanisms, which may constitute the basis for future developments. The tests were carried out for three problem space sizes, with 20, 40 and 60 elements, with the obtained results being averaged.

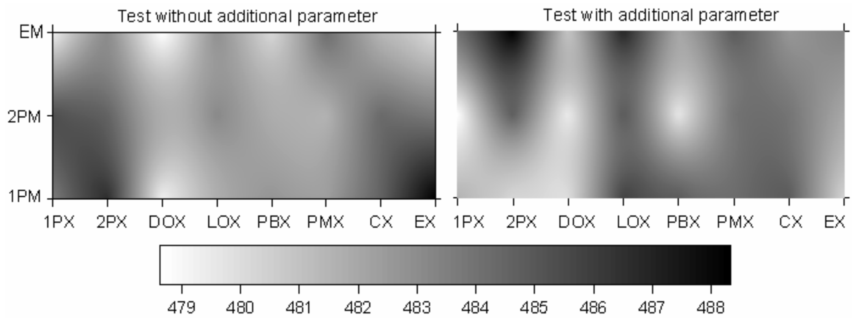

Fig. 2. Combinational efficiency for the genetic operators (lower values are better)
Rys. 2. Skuteczność kombinacji operatorów genetycznych (mniejsze wartości są lepsze)

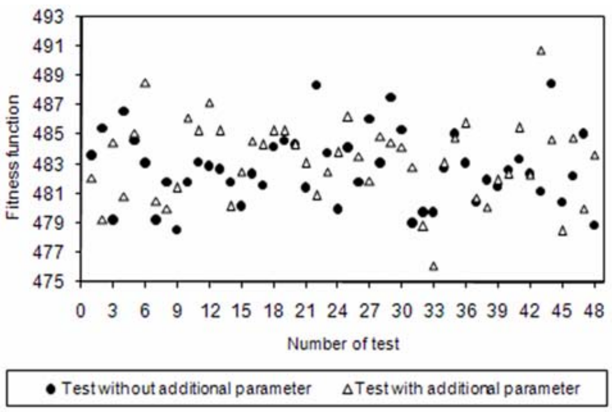

Fig. 3. Combinational efficiency for the genetic operators (lower values are better)
Rys. 3. Skuteczność kombinacji operatorów genetycznych (mniejsze wartości są lepsze)

The obtained results are depicted in Figs 2 and 3. Due to the significant result spread (Fig. 3), it is difficult to unanimously determine the impact of the individual genetic operators on the fitness function value. In the case of the algorithm with no additional parameter, the best results were exhibited by operator combination number 9, while parameter combination 33 performed best with the additional parameter enabled. The latter one also provided the globally best solution. Application of this particular parameter in conjunction with the roulette selection method in most cases lowers the algorithm efficiency, but improves it when applied with tournament selection.

All combinations of genetic operators where the one-point or two-point mutation was employed, providing that no additional parameter was used, are more likely to create solutions with lower fitness value when compared with the uniform mutation scheme. The situation is, however, the opposite when the additional parameter is employed, where the high efficiency is exhibited by single or double point mutation schemes, while the uniform mutation method underperforms (Fig. 2).

Unfortunately, it is impossible to determine which crossing operator outperforms the other ones. Special attention should be paid to their computational complexity and determine whether an improved solution is obtained at the cost of more extended computations and prolonged execution time.

It seems well justified to employ the proposed additional parameter in the case of tournament selection process while it is pointless when the roulette selection mechanism is used. The final decision on application of the additional parameter should therefore be correlated with the selection of the particular mutation scheme, in order to assure the highest efficiency of the implemented algorithm. For the tests collected in Table 2, it can be assumed that the average best results are generated by the T-DOX-EM scheme.

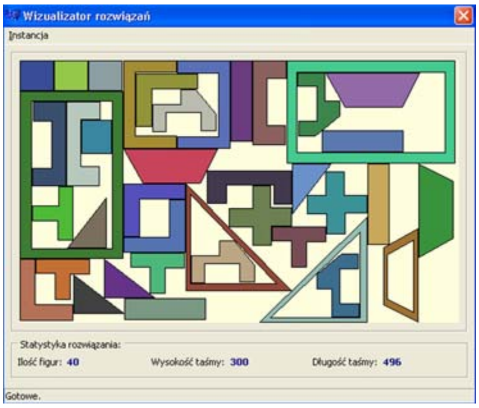

Fig. 4. Program interface with an example of a solution

Rys. 4. Interfejs programu z przykładowym rozwiązaniem

Figure 4 depicts the interface of the developed software package for optimization along with an example of the generated solution.

6. Practical application of the proposed algorithm

The proposed algorithm can in practice be employed in the industry. The areas of interest include furniture manufacturers, industrial tailors, etc., that is companies which use

CAD type software packages. Figure 5 depicts software integration with the AutoCAD 2007 package, where the sub-optimum solution was exported once completed. This operation was achieved via OLE (COM) technology and may constitute a base for successful integration of the algorithm with other CAD type packages.

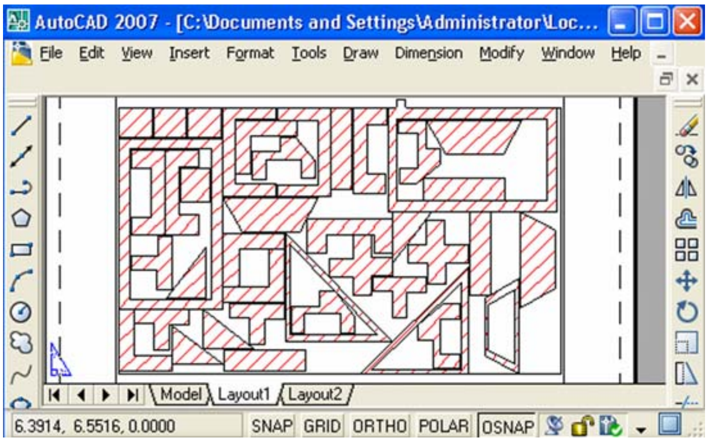

Fig. 5. Algorithm integration with AutoCAD 2007
Rys. 5. Integracja algorytmu z programem AutoCAD 2007

7. Conclusions

In the article we proposed and examined a universal genetic algorithm for 2-dimensional irregular cutting problem, allowing for effective placement of individual elements on the target material sheet. It was proven that for certain combinations of genetic operators it is possible to improve the overall efficiency by employing an additional parameter. The potential practical application of the proposed algorithm was also presented, in the form of integration with the CAD type packages.

Future works should focus on the selection of optimum crossing and mutation probabilities as well as the value for the additional parameter, which may result in the improvement of the quality of the obtained solutions.

References

[1] Brooks R., Smith C.A.B., Stone A.H., Tutte W.T., *The Dissection of Rectangles into Squares*, Duke Math. Journal, Vol. 7, 1940.

[2] Błażewicz J., Hawryluk P., Walkowiak R., *Using a Tabu Search Approach for Solving the Two-Dimensional Irregular Cutting Problem*, Annals of Operations Research, Vol. 41, 1993, 312-327.

[3] Burke E., Hellier R., Graham K., Whitwell G., *A New Bottom-Left-Fill Heuristics Algorithm for Two Dimensional Irregular Packing Problem*, Operations Research, Vol. 54, No. 3, May–June 2006, 581-601.

[4] B u d z y ń s k a L., K o m i n e k P., *Influence of Given Representation on Performance of an Evolutionary Algorithm for 2d Irregular-Shape Cutting Problem*, Symposium on Methods of Artificial Intelligence, Gliwice, November 5–7 2003.
[5] B u d z y ń s k a L., K o m i n e k P., *The Application of GLS Algorithm to 2 Dimensional Irregular-Shape Cutting Problem*, Lecture Notes in Computer Science, Vol. 3038, Springer, Berlin 2004, 1241-1248.
[6] D a v i s L., *Applying Adaptive Algorithms to Epistatic Domains*, Proceedings of the International Joint Conference on Artificial Intelligence, 1985.
[7] F a l k e n a u e r E., B o u f f o u i x, *Genetic Algorithm for Job Shop*, Proceedings of 1991 IEEE International Conference on Robotics and Automation, 1991.
[8] S y s w e r d a G., *Schedule optimization Using Genetic Algorithms*, Handbook of Genetic Algorithms, New York 1991.
[9] G o l d b e r g D.E., L i n g l e R., *Alleles, Loci, and Travelling Salesman Problem*, Proceedings of the First International Conference on Genetic Algorithms, Hillsdale, New York 1985.
[10] O l i v i e r L., S m i t h D., H o l l a n d J., *A Study of Permutation Crossover on the Traveling Salesman Problem*, Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, New York 1987.