

# Zmodyfikowany algorytm genetyczny dla dwuwymiarowego nieregularnego problemu optymalnego rozkroju

Sławomir Żak

Politechnika Krakowska

Międzywydziałowy Kierunek Informatyka Stosowana, Rok IV

slawomir.zak@gmail.com

## Streszczenie

Problemy optymalnego rozkroju występują niezmiennie często w różnych dziedzinach techniki. Opracowano liczne algorytmy pozwalające na rozwiązywanie szczególnych wariantów tychże problemów (rozkrój gilotynowy, niegilotynowy itp.) W artykule zaproponowano zmodyfikowany algorytm genetyczny pozwalający na optymalizację wykroju elementów o nieregularnych kształtach z powierzchni dwuwymiarowych. Przedstawiono wyniki symulacji, w trakcie których zbadano wpływ doboru standardowych operatorów genetycznych. Pokazano również możliwość praktycznego użycia algorytmu w zastosowaniach inżynierskich dzięki integracji z programem AutoCAD 2007.

**Słowa kluczowe:** *problem rozkroju, algorytm genetyczny, automatyzacja AutoCAD*

## 1 Wstęp

Problem rozkroju to zagadnienie, dla którego istnieje szereg możliwych wariantów (rozkrój gilotynowy, niegilotynowy, nieregularny itp.). Najogólniej można go scharakteryzować w następujący sposób: dysponując jednolitym płaskim elementem (płat blachy, kartka papieru) o określonym wymiarze oraz zbiorem kształtów, poszukujemy takiego ich rozmieszczenia na tym elemencie, aby zminimalizować ilość powstałych ścinków podczas ich wycinania.

Pierwsze badania nad problemem rozkroju zostały zapoczątkowane w roku 1940, gdzie w pracy [1] rozważano optymalny podział prostokąta na kwadraty. W późniejszych latach wykazano NP-zupełność problemu rozkroju jak i przedstawiono szereg różnych jego odmian często wraz z dedykowanymi dla nich algorytmami. W niniejszym artykule rozpatrywano minimalizację powierzchni użytego płata materiału, mającego zadeklarowaną z góry wysokość i nieograniczoną długość, wycinając z niego elementy o nieregularnych kształtach [2]. Dla tak postawionego problemu został zaproponowany oraz przebadany, pod względem efektywności działania, uniwersalny algorytm genetyczny z dodatkowym parametrem pozwalający na optymalizację ułożenia zarówno elementów regularnych jak i nieregularnych. Przeanalizowano wpływ doboru operatorów genetycznych

na jakość otrzymywanego rozwiązania celem doboru ich najlepszej kombinacji. Dodatkowo została przedstawiona możliwość praktycznego użycia algorytmu w zastosowaniach inżynierskich dzięki integracji z programem firmy Autodesk - AutoCAD 2007.

## 2 Charakterystyka zastosowanego algorytmu

Ze względu na nieregularny charakter wycinanych elementów składowane są one w pliku XML w postaci wektorowej. Takie podejście jest bardzo wygodne, jeżeli chodzi o łatwość manipulowania geometrią kształtu i pozwala w prosty sposób definiować dowolną liczbę otworów (ang. holes) wewnątrz figur. Niesie jednak ze sobą trudność w określeniu stanu, kiedy dwie figury nachodzą na siebie wymagając zastosowania skomplikowanych technik badania przecinania się krawędzi [3]. Problem ten został rozwiązany poprzez zastosowanie rasteryzacji figur i porównywanie, czy poszczególne piksele dwóch figur nie nachodzą na siebie. Została tu również użyta technika zmniejszająca rozmiar macierzy rastra, poprzez zgrupowanie pikseli i przechowywanie ich jako 32-bitowej zmiennej typu Integer [4].

Stworzony algorytm wykorzystuje technikę układania elementów na taśmie zwaną Bottom-Up-Left-Justified [5], w której rozwiązanie jest sekwencją występujących kolejno po sobie kształtów układanych począwszy od górnego lewego rogu taśmy kolejno w dół, a następnie coraz bardziej na prawo, jeżeli figury nie mieszczą się już w danej kolumnie. Dodatkowo zastosowano możliwość obrotu figur o następujące kąty:  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ , co zapewnia większą swobodę w układaniu rozwiązania.

Zastosowany sposób rozmieszczania kształtów na taśmie uwzględniający kąty obrotu figur z góry odrzuca możliwość użycia binarnego kodowania z jakim najczęściej mamy do czynienia w przypadku algorytmów genetycznych. Z tego też powodu zastosowano chromosom o następującej budowie:

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

gdzie :

$(X_i, R_i)$  - określa położenie oraz kąt obrotu elementu

Funkcja przystosowania rozwiązania określona jest jako liczba pikseli na długość wymaganych do rozmieszczenia wszystkich układanych kształtów. Wartości bliższe zeru charakteryzują rozwiązanie lepsze.

## 3 Zastosowane operatory genetyczne

Zaimplementowano oraz sprawdzono efektywność działania dwóch klasycznych operatorów selekcji - *selekcja metodą koła ruletki* (R) oraz *selekcja turniejowa* (T). W przypadku pierwszej proporcjonalnie częściej wybierane są osobniki o funkcji przystosowania bliższej zeru. Natomiast selekcja turniejowa tworzy podpopulację o rozmiarze 10% populacji wejściowej z losowo wybranymi osobnikami i dopiero z niej dokonywany jest wybór kolejnego rodzica.

Specyficzna budowa chromosomu, w której zakodowane rozwiązania ma charakter permutacyjny bez powtórzeń, wymaga zastosowania specjalnych operatorów krzyżowania, które będą czuwały, aby nie dochodziło do generowania błędów. Takie ograniczenie

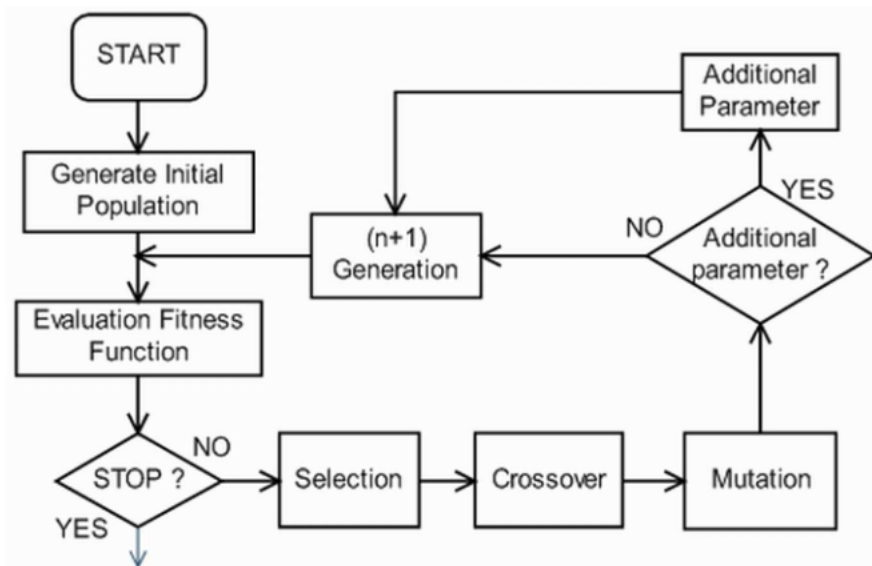
posiada większość operatorów krzyżowania znanych z problemu TSP (Traveling Salesman Problem). W toku badań wykorzystano następujące krzyżowania: *jednopunktowe* (1PX), *dwupunktowe* (2PX), *porządkowe* (DOX) [6], *liniowe* (LOX) [7], *pozycyjne* (PBX) [8], z częściowym *dopasowaniem* (PMX) [9], *cykliczne* (CX) [10] oraz *równomierne* (EX). Wszystkie powyższe operatory pracują na całym chromosomie przetwarzając zarówno kolejność układanych elementów jak i kąty obrotu.

Głównym zadaniem operatorów mutacji jest zapewnienie wyjścia algorytmu z lokalnego minimum. Na bazie ich klasycznych wersji, stosowanych zazwyczaj przy binarnym kodowaniu chromosomu, czyli mutacji *jednopunktowej* (1PM), *dwupunktowej* (2PM), *równomiernej* (EM) stworzono operatory, które nie pracują na całym rozwiązaniu (chromosomie) a jedynie na jego części (kątach obrotu kształtów). W ten sposób wyjście z lokalnego minimum zapewnione jest, dzięki możliwości obrotu figury o jeden z losowo wybranych kątów ( $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ ).

W ramach prób podniesienia efektywności algorytmu zaproponowano i przebadano dodatkowy operator genetyczny. Istota jego działania jest następująca: z pewnym prawdopodobieństwem część osobników w populacji (10%) zostaje usunięta, a na ich miejsce wprowadzane są nowe. Eliminowane osobniki wybierane są w sposób losowy, co ma zapewnić większą różnorodność populacji.

## 4 Implementacja algorytmu

Algorytm został zaimplementowany w środowisku Borland Developer Studio 2006 z wykorzystaniem języka C++. Większość kodu oparto o bibliotekę STL (Standard Template Library) celem zagwarantowania jak największej wydajności. Schemat blokowy algorytmu przedstawiono na Rys. 1.



Rys. 1: Diagram blokowy algorytmu

Fig. 1: Block diagram of the implemented algorithm

## 5 Wyniki przeprowadzonych testów

Przeprowadzono eksperymenty badające wpływ różnych kombinacji operatorów genetycznych na jakość uzyskiwanych wyników. Wykonano próby zarówno z zaproponowanym dodatkowym parametrem jak i bez niego. Testy zostały przeprowadzone według następującego planu Tab. 1:

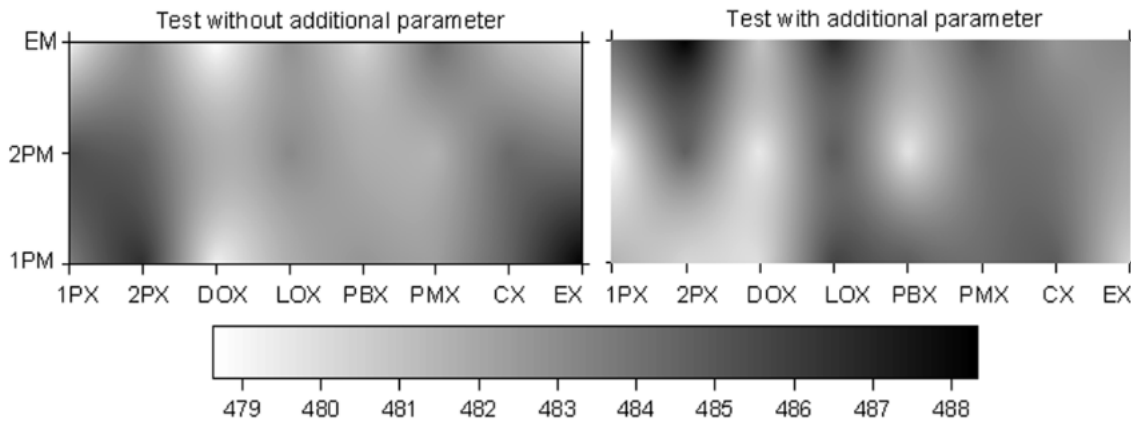
Tab. 1: Zastosowane kombinacje operatorów genetycznych

Test	Selekcja	Krzyżowanie	Mutacja	Test	Selekcja	Krzyżowanie	Mutacja
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

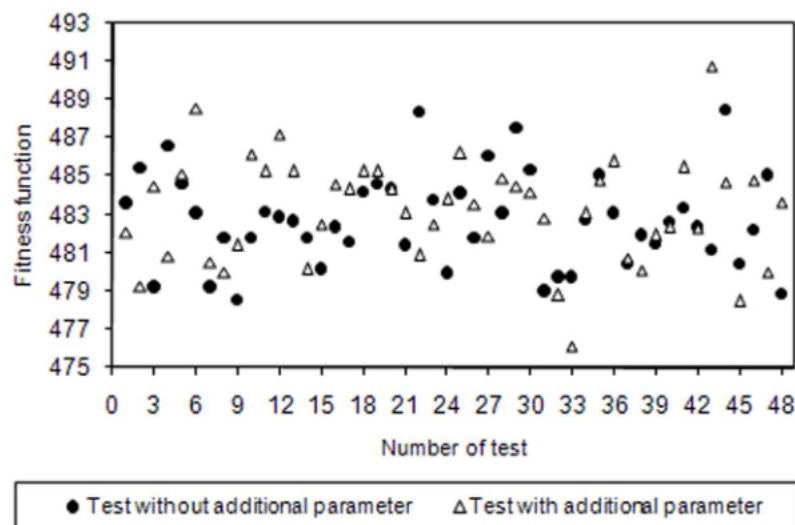
Tab. 2: Przyjęte wartości parametrów sterujących algorytmu

<i>Parametr</i>	<i>Wartość</i>
Prawdopodobieństwo krzyżowania	0.6
Prawdopodobieństwo mutacji	0.05
Rozmiar populacji	30
Liczba epok	20
Liczba powtórzeń algorytmu	5
Dodatkowy parametr (gdy był używany)	0.1

Wartości zastosowanych parametrów algorytmu genetycznego zostały dobrane a priori, w oparciu o klasyczne przykłady z literatury, i przedstawione w Tab. 2. Wykorzystano klasyczny algorytm, bez mechanizmów adaptacyjnych, co może stanowić podstawę do jego dalszej rozbudowy. Testy były prowadzone dla trzech instancji problemu rozkroju liczących kolejno 20, 40 i 60 elementów, a otrzymane wyniki zostały uśrednione.



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



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

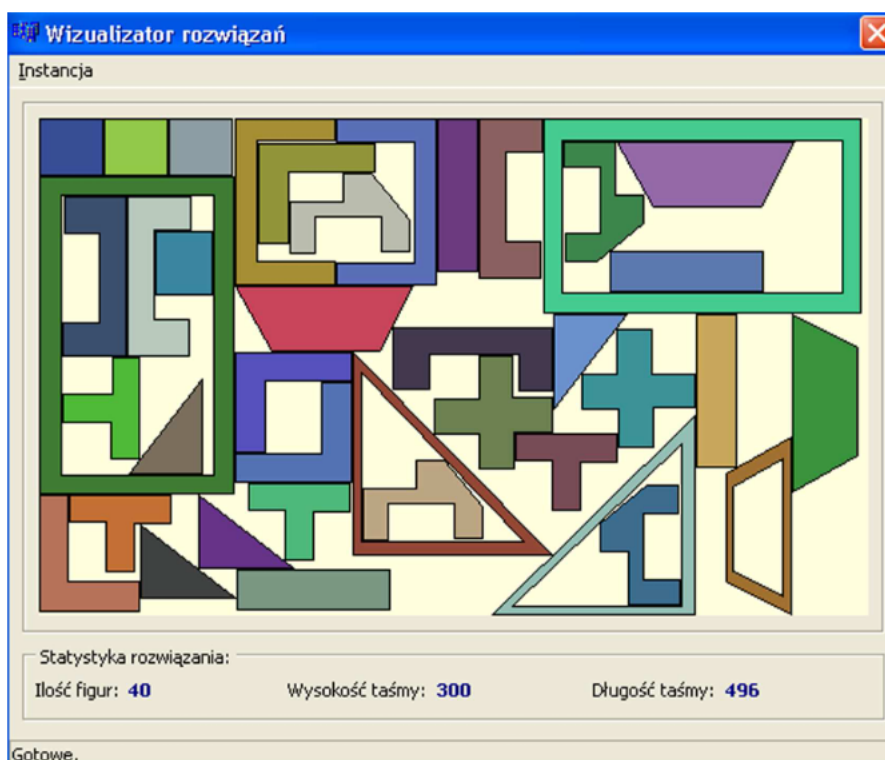
Otrzymane rezultaty zaprezentowano na Rys. 2 i Rys. 3. Ze względu na stosunkowo duże zróżnicowanie wyników (Rys. 3), trudno jest jednoznacznie określić jaki wpływ na wartość funkcji przystosowania mają poszczególne operatory. W przypadku algorytmu bez zastosowanego dodatkowego parametru najlepsze rozwiązanie zapewniła kombinacja numer 9, natomiast z dodatkowym parametrem kombinacja numer 33 dająca również rozwiązanie globalnie najlepsze. Użycie tego parametru, z selekcją metodą ruletki, w większości przypadków obniża efektywność algorytmu, a z selekcją turniejową przynosi poprawę wyników.

Kombinacje operatorów, w których zastosowano mutację jednopunktową lub dwupunktową bez korzystania z dodatkowego parametru, częściej tworzą rozwiązania słabsze niż te, w których używano mutacji równomiernej. Zupełnie odwrotna sytuacja jest w przypadku wykorzystania dodatkowego parametru, gdzie ciężar operatywności zostaje przeniesiony z mutacji równomiernej na jednopunktową lub dwupunktową (Rys.2).

Niestety nie jest możliwe, określenie który operator krzyżowania charakteryzuje się największą wydajnością. Należałoby tutaj zwrócić uwagę na ich złożoność obliczeniową i sprawdzić czy uzyskujemy spodziewany zysk kosztem wydłużenia czasu obliczeń.

Wydaje się celowym stosowanie zaproponowanego dodatkowego parametru w przypadku wykorzystywania selekcji turniejowej i bezskuteczne w przypadku selekcji ruletką. Decyzja o użyciu dodatkowego parametru powinna być skorelowana z wyborem metody mutacji aby zapewnić największą sprawność algorytmu. Dla testów z Tab. 1 można przyjąć, że średnie najlepsze rozwiązania generuje kombinacja: T - DOX - EM

Na Rys. 4 przedstawiono interfejs programu do optymalizacji wraz z wygenerowanym przykładowym rozwiązaniem.



Rys. 4: Interfejs programu z przykładowym rozwiązaniem

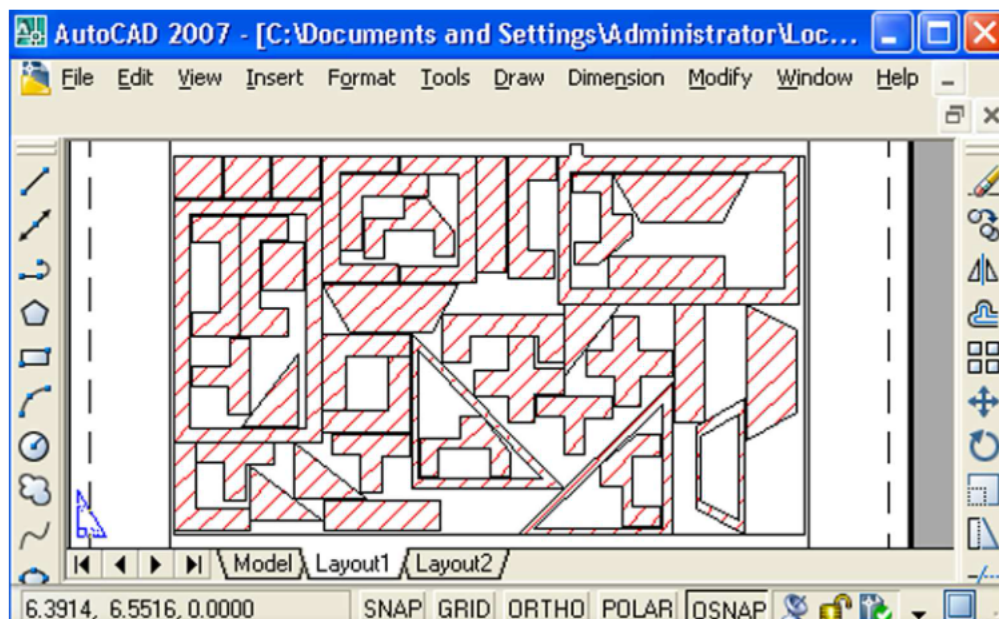
Fig. 4: Program interface with exemplary solution

## 6 Możliwość praktycznego wykorzystania algorytmu

Stworzony algorytm może być praktycznie wykorzystywany w przemyśle. Przykładem mogą tu być producenci mebli, zakłady krawieckie itp., czyli firmy które użytkują oprogramowanie typu CAD. Na Rys. 5 przedstawiono integrację algorytmu z programem



AutoCAD 2007, do którego po wykonaniu obliczeń zostało wyeksportowane suboptymalne rozwiązanie. Operacja ta jest realizowana dzięki wykorzystaniu technologii OLE (COM) i może stanowić podstawę do integracji algorytmu z innymi aplikacjami typu CAD.



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

## 7 Wnioski

W artykule przedstawiono i przebadano uniwersalny algorytm genetyczny dla dwuwymiarowego nieregularnego problemu rozkroju, pozwalający na efektywne układanie elementów na taśmie. Wykazano, że dla pewnych kombinacji operatorów genetycznych możemy uzyskać ich większą sprawność dzięki użyciu zaproponowanego dodatkowego parametru. Przedstawiono również możliwość ewentualnego praktycznego wykorzystania algorytmu w programach typu CAD. Dalsze prace powinny skupić się na doborze optymalnych wartości prawdopodobieństwa krzyżowania, mutacji oraz wartości dodatkowego parametru, co może przynieść wzrost jakości uzyskiwanych rozwiązań.

## Literatura

- [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, pp 312-327, 1993.

- [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, pp 581-601, May-June 2006.
- [4] Budzyńska L., Kominek 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] Budzyńska L., Kominek P., *The Application of GLS Algorithm to 2 Dimensional Irregular-Shape Cutting Problem*, Lecture Notes in Computer Science, Vol. 3038, pp. 1241-1248, Springer, Berlin 2004.
- [6] Davis L., *Applying Adaptive Algorithms to Epistatic Domains*, Proceedings of the International Joint Conference on Artificial Intelligence, 1985.
- [7] Falkenauer E., Bouffouix, *Genetic Algorithm for Job Shop*, Proceedings of 1991 IEEE International Conference on Robotics and Automation, 1991.
- [8] Syswerda G., *Schedule optimization Using Genetic Algorithms*, Handbook of Genetic Algorithms, New York 1991.
- [9] Goldberg D.E., Lingle R., *Alleles, Loci, and Travelling Salesman Problem*, Proceedings of the First International Conference on Genetic Algorithms, Hillsdale, NJ, 1985.
- [10] Olivier L., Smith D., Holland J., *A Study of Permutation Crossover on the Traveling Salesman Problem*, Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, NJ, 1987.

### Summary

The optimum cutting problems occur very frequently in various area of industry and technology. A number of algorithm were proposed, targeting solving particular variants of the aforementioned problems (guillotine cut, non-guillotine cut, etc.). In this 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 the 2 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 a 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*