AMO–Advanced Modeling and Optimization, Volume 7, Number 1, 2005

Generalized particular covering problem with genetic algorithms*

László Illyés [1], László Pál [2]

[1] Sapientia University, Department of Mathematics and Informatics
E-mail: illyeslaszlo@sapientia.siculorum.ro

[2] Sapientia University, Department of Mathematics and Informatics
E-mail: pallaszlo@sapientia.siculorum.ro

Abstract
This paper proposes to solve the Area Problem with generalization and to find a good structure of chromosome. The problem with a particular Large Object was presented in [IL04]. We use the 2 biggest square heuristics to restrict the area of possibilities and to find the length of chromosome.
We have a single square Large Object and square Little Objects, where each size may appear only once. The problem is to cover with these squares the Large Object, minimizing the waste.

This paper can be categorized cf. [WHS04] as Single Knapsack Problem. (SKP)

Key words: rectangle placement, knapsack, genetic algorithm

1. Introduction

Rectangle layout is an important problem with applications in the glass cutting and paper cutting industries. There are many variants of the problem but the essential task is to place a given set of rectangle, or pieces, in a given larger rectangular area, so that the wasted space in the resulting layout is minimized. There are some popular placement algorithms, such as the BL strategy [J96], IBL [LT99], MERA [ABH04]. In this paper we propose a problem which is a particular case of the rectangular placement problem.

1.1 Notation and terminology

We have a single square Large Object with size $L \in N^*$ and square Little Objects, with sizes m, where each size may appear only once, $m \in [1, L-1]$. The problem is to cover with these squares the Large Object, minimizing the waste.

l_i - the size of side of square i
L – the stock size
$S=LxL$ – the whole area of the Large Object
s_i – The area of square i – in our case $l_i^2=i^2$

We have two problems to solve in this kind of problems:
1. Selection of pieces – The Area Problem
2. Putting on them on the pattern in a not overlapping form – The Covering Problem

* Supported by the Research Programs Institute of Foundation Sapientia, Grant Nr. 1326/2004

This paper proposed to discuss only the first problem. The Covering Problem will be discussed in next papers.

The mathematical model for the problem 1 is:

$$\begin{cases} Max\left(\sum_{i=1}^{m} s_i * x_i\right) \\ \sum_{i=1}^{m} s_i * x_i \le S \end{cases}$$

where $x_i \in \{0,1\}$ - $x_i=1$ when we use the $\boldsymbol{i}^{th}$ square otherwise 0

We assume, that we try to found a non-trivial solution, better than that solution in that we put the *(L-1)* length sided square and the 1 square on board:

$$(L-1)^2 + 1^2 \le \sum_{i=1}^{m} s_i * x_i \le L^2$$

and in other terms the waste proportion $\mathrm{w_p} \le 2*(L-1)/L^2$, but the exact solution is also a perspective.

The main thing in this kind of problems are the following:

A. We have to restrict the area of the possible solutions.
B. We have to find the complexity of the problem.

2. The Area Problem – hypothetical aspects

We use the 64x64 Large Object in our model, and we try to generalize this in parallel.

We have to calculate the maximum number of squares that we can put on the Square Large Object.

Lemma 1. *Maximum number of different squares m possible to put on the Stock satisfies the following inequalities:*

$$1^2 + 2^2 + \ldots + m^2 \le L^2 \le 1^2 + 2^2 + \ldots + m^2 + (m+1)^2 \ (1.1)$$

$$m(m+1)(2m+1)/6 \le L^2 \le (m+1)(m+2)(2m+3)/6 \ (1.2)$$

This Lemma is obvious, no need to demonstrate.

In the next graphics we can see the maximum number of squares that we can put on the given large square.

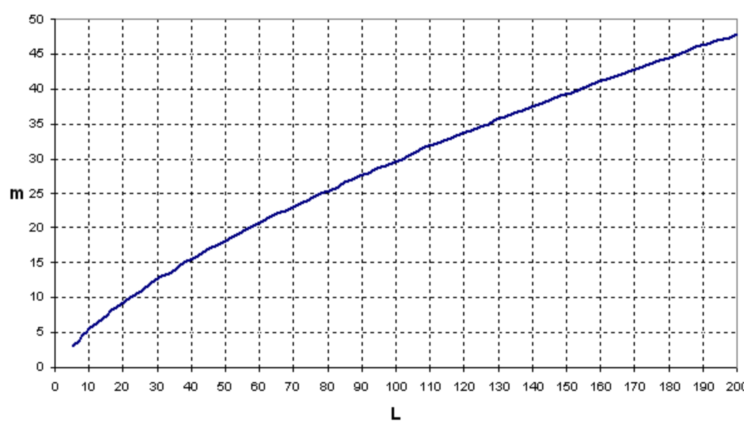

Fig. 1. Maximum number of squares

Considering, that only those squares can be situated side by side that satisfy the relation: $l_1+l_2<=L$ (where l_1 and l_2 are the sides of the squares), and calculating with these squares we have the next relations referring to their area (also 64x64):

$$58^2+\sum_{i=1}^{6} i^2 = 3455<4096 \qquad 54^2+\sum_{i=1}^{10} i^2 = 3301 \qquad 53^2+\sum_{i=1}^{11} i^2 = 3315$$

$$47^2+\sum_{i=1}^{17} i^2 = 3994 \qquad 46^2+\sum_{i=1}^{18} i^2 = 4225>4096$$

With these calculations we have the following figure, which gave us the maximum size of the biggest square when the Large Object has 64x64 size.

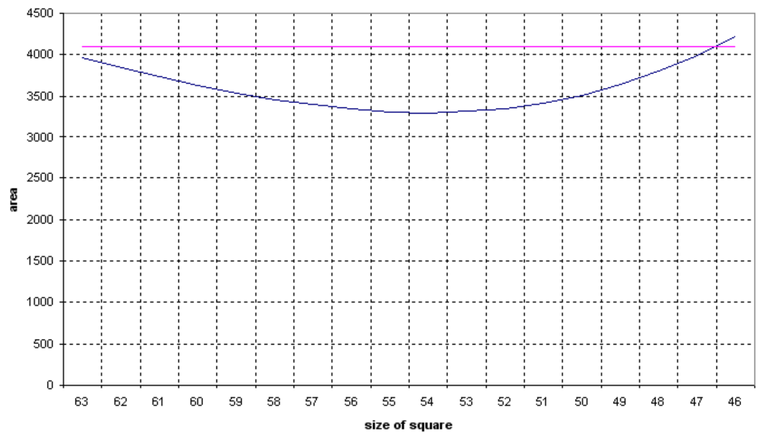

Fig.2. Possible covering area for 64x64

Lemma 2. *The minimum size L of the Large Object to have a possible solution to cover the whole area is* 19.

*Proof***.**

Let be k the size of the greatest square we put on the Large Object. We can put after we put this on, only squares smaller or equal with L-k. In case we put them all, we can write this in the following way:

$$k^2+\sum_{i=1}^{L-k} i^2 \geq L^2 \quad (2.1)$$

From this inequality we get:

$$(L-k)((L-k+1)(2L-2k+1)-6(L+k)) \geq 0 \text{, where } L>k \quad (2.2)$$

$$(L-k+1)(2L-2k+1)-6(L+k) \geq 0 \quad (2.3)$$

Critical points of this Lemma are in case when (2.1) extends the form (1.1), when $k \approx L/2$

a) In case, that $L=2n+1$
   The critical point is, in case when $k=(L+1)/2$
   Introduced this in the inequality we have:
   $L^2-17L-6 \geq 0$
   That means: $L_{min}=19$
b) In case, that $L=2n$
   The critical point is, in case when $k=L/2+1$
   If we introduce this is the inequality, we have:
   $L^2-19L-12 \geq 0$
   That means: $L_{min}=20$

The minimum of these two minims is 19 and we demonstrated the lemma.

In the next figure we can see the relation between the square Large Object and the biggest square.

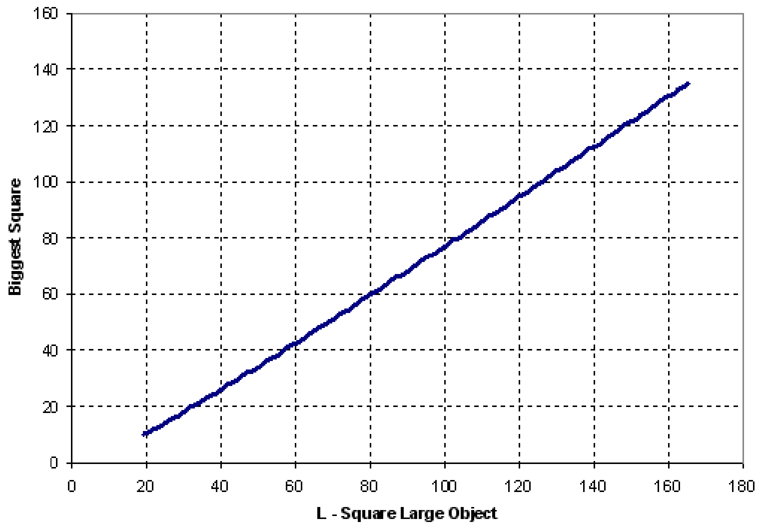

Fig.3. Relation Large Object-Biggest Square

We can use the (2.3) to have a relation between *L* the size of Square Large Object and *k*, size of the biggest possible square that we are allowed to put on, to satisfy the area covering option.
If $L=19$ than $k=10$, if $L=100.000$ than $k=99.227$, $k/L \rightarrow 1$, when $L \rightarrow \infty$ (2.4)

3. Genetic algorithm

3.1 Chromosome structure

The proposed length of chromosome is the size of the biggest square.

46	45	...	33	32	31	...	18	17	...	1
1	0	0	0	0		0	X	X	X	X

46	45	.....	33	32	31	...	18	17	...	1
0	0	0	1	0	X	X	X	X	X	X

We construct a haploid chromosome, we have a chromosome with length *k* bit, in this case 46 bit, where 1 represents the chosed square, and 0 represents the avoided square from the set.

k	k-1	.....	k-h+1	k-h	k-h-1	...		3	2	1
X	X	X	X	X	X	X	X	X	X	X

Like the 2 biggest square heuristic shows us, from the 46-th position of genes in chromosome (in general case *k*) to 32 (*L/2 or (L+1)/2*)-th chromosome we have only one of 1 allele, the others are 0. We can use this, when we generate the genes in a chromosome, to avoid the known bad solutions. We can proceed in the following way:

1. We generate the upper part of chromosome randomly (from *k* to *L/2*) to have one allele of 1 or all 0.
2. In case, we generate the upper part with one allele of 1 in position *n*, we complete with 0–s until position $m = L\text{-}k\text{-}n+1$
3. We generate the remaining part of chromosome randomly.

Point 2. has an inconvenient: the diversity of chromosomes is restricted also. Maybe is better to generate the chromosome with Point 1. and Point 3.

3.2 Fitness function. Genetic operators

There is no solution, when *S* > *LxL* (4096), because some parts of some squares are out of the square Large Object. When we perform operation on chromosomes we don't drop those solutions, because some part of the chromosome can be good, and we can use them when we make crossover operations, but we give them a fewer fitness, a fewer chance to be a chose parent.
The fitness function of the chromosomes we write like this:

$$f = \begin{cases} \left(0.7 \cdot \dfrac{\sum_{i=1}^{n} s_i}{S}\right)^2, & if \quad \sum_{i=1}^{n} s_i > L^2 \\ \left(\dfrac{S}{\sum_{i=1}^{n} s_i}\right)^2, & if \quad \sum_{i=1}^{n} s_i \le L^2 \end{cases}$$

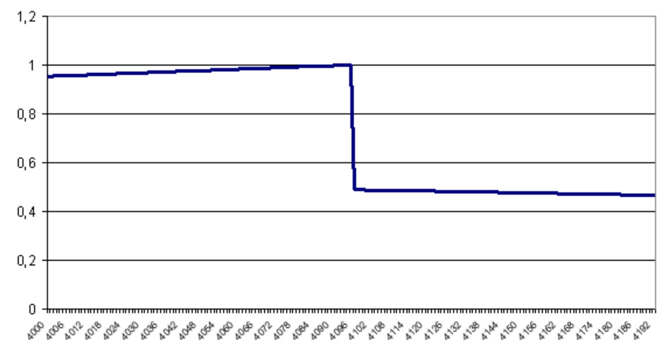

Fig. 4. Fitness function

We choose the quadratic function, to control the convergence of the population.

We calculate the average of the fitness function $ff = \sum_{i=1}^{k} f_i / k$, where *k* is the number of individuals in a population. In the case of the roulette wheel, fitness based selection, the individuals of the populations are chose to be parents according to this proportion: *f/ff*. Like *De Jong* proposed[AL02], every time, when we choose an individual to be a parent for the next population we decrease this probabilistic proportion with one, to avoid the early convergence of the population. If the proportion is 1, we don't decrease this.

We choose two-point crossover operator, and accept both of the children in the next generation. We can perform the operator over 46 bits (*k* bits) or over the 32 (*L/2*) last significant bit, to avoid the 2 bit of 1 in the significant part of the chromosome. This heuristically approach we discussed before for other reason. In general case 46→k and 32→L/2 (or (L+1)/2)

Complexity of the problem

With formula 2.5, we could establish the complexity of the first problem, an upper bound, and we consider again, that the sum of length of sides of 2 adjacent squares $l_1+l_2<=L$. It means that from k to $L/2$ (32 now) position in the chromosome, we have only one of 1 bit. (46-32=14, which means, that we have only one of 1 bit on 15 bits.)

$$2^{18}+2^{19}+...+2^{31}+2^{32}=2^{18}(1+2+...+2^{14})=2^{18}(2^{15}-1)\text{, if } L = 64$$

$$2^{19}+2^{20}+...+2^{31}+2^{32}=2^{19}(1+2+...+2^{13})=2^{19}(2^{14}-1)\text{, if } L = 65$$

Lemma 3. *The Area Problem in this case is NP complex.*

Proof.

One upper bound for this general problem we can calculate in the next way:

$$2^{(L-K)}+2^{(L-K+1)}+...+2^{(L-K+L/2)},\quad \text{if } L = 2n \qquad (3.1)$$

$$2^{(L-K+1)}+2^{(L-K+2)}+...+2^{(L-K+(L-1)/2)}\text{, if } L = 2n+1 \qquad (3.2)$$

in other terms,

$$2^{(L-K)}(2^{(L/2)+1}-1)\text{, if } L = 2n \qquad (3.3)$$

$$2^{(L-K+1)}(2^{(L-1)/2+1}-1)\text{, if } L = 2n+1 \qquad (3.4)$$

Because $2^{(L-K)}(2^{(L/2)+1}-1) > 2^{(L-K+1)}(2^{(L-1)/2+1}-1)$, we can consider the (3.3) an upper bound of the problem

We can approximate (3.3) by $2^{L-K+L/2+1}=2^{3L/2-K+1}$

If $L\to\infty$ and $k = L$ we have in worst case $O(L)=2^{L/2+1}$, so we can say, that in this approximation is NP complex problem.

In practice we found few entries upper than the 42 unit length square (1 piece of 44, 2 of 43).

3.3 Practical result for Large Object 64x64

Like we show in the previous paper [IL04], we have over 300,000 results for the Area Problem. We chose chromosomes that satisfy the $l_1+l_2=64$ where l_1 and l_2 are the sizes of the two biggest square, and the "almost" Divide et Impera (33+31) and (34+30). The best result until now is when 35 unit square is not covered**.**

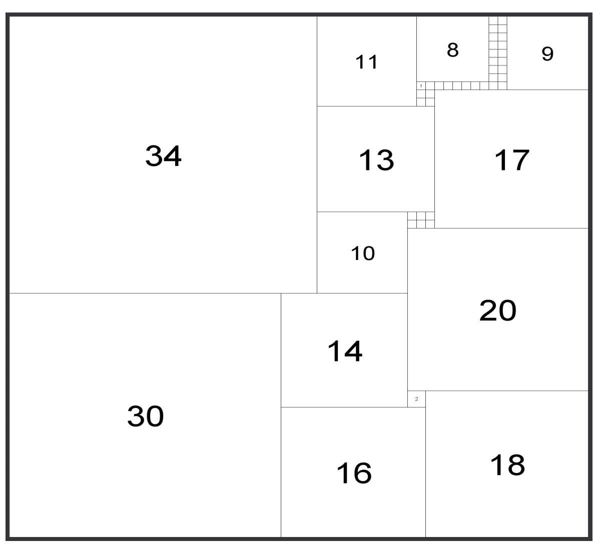

Fig. 5. Square of 64x64

Conclusions

We found the **length of chromosome** for the Area Problem in general case.
We found the **deep of a general graph tree**. (The maximum number of small squares.)
We found **19 the minimum size** of the Square Large Object for this problem.
We found, that the Area Problem is an **NP complex** problem.
We found the **constraints** between the Alleles of a chromosome, that describes the area of solutions.
We found **better solution**, than in the previous paper , where was 107 uncovered unit [IL04].
Our **conjecture** is that combining genetic algorithm with the **„Divide et Impera"** method gave good results, but **not** the **global optimum** in this problem.

References

[ABH04] Ahmad, A.R., Basir, O., Hassanein, K., (2004), "Improved Placement Algorithm for Layout Optimization", Proc. of the 2nd Int'l Industrial Engineering Conf. (IEEC'04), Dec. 2004, Riyadh, Saudi Arabia, *to appear*

[ABIH04] Ahmad, A.R., Basir, O., Hassanein, K., (2004), "A Placement Algorithm for Efficient Generation of Superior Decision Alternatives in Layout Design", Proc. of the 5th Int'l Conference on Operations and Quantitative Management , October 25-27, 2004.

[AL02] Álmos Attila et.al., (2002), Genetikus Algoritmusok, Typotex 2002

[BOL70] Boldur Gh., (1970), Unele aspecte ale problemei croirii elementelor dreptunghiulare, SNCCECE (Studii si Cercetari de Calcul Economice si Cibernetica Economica); Nr 6, 1970

[FAB01] Fabian,Csaba, (2001), Two-dimensional Cutting Stock Problems Using Evolutionary Algorithms, IE-2001 Proceedings, Inforec, Bucharest, 2001, pag. 513-517

[FAB02] Fabian, Cs. B., (2002), Generalized Simple and Crossover Mutations for Evolutionary Algorithms, International Conference on Economic Cybernetics Bucharest, 2002

[FAIL04] Fabian, Cs. B., Illyes, L., (2004), Solving Fuzzy Cutting Stock Problems Using Fuzzified Genetic Algorithms 1'st ESICUP Meeting –Lutherstadt-Wittenberg-Germany 18-20 march 2004

[HT01] Hopper, E., Turton, B.C.H. (2001) "An Empirical Investigation of meta-heuristic and heuristic algorithms for a 2D packing problem", Euro.J. of Op. Res., 128, pp. 34-57

[IL04] Illyés, L. (2004) "Genetic algorithms for a particular covering problem" International Conference on Economic Cybernetics Bucharest, April 23-25, 2004, *to appear*

[J96] Jakobs, S., (1996), "On genetic algorithms for packing polygons", Euro. J. of Op. Res., 88, pp. 165-181.

[LT99] Liu, D., Teng, H., (1999), "An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles", Euro. J. of Op. Res., 112, pp. 413-420

[WHS04] Wäscher, G.–Hauβner, H.–Schumann, H., (2004), An improved Typology of Cutting and Packing Problems – Working Paper – Faculty of Economics and Management Magdeburg

[VB04] Vyatkina, K., Bukhalova, V., (2004), On separating Boxes with Right-Angled Cuts, 1'st ESICUP Meeting –Lutherstadt-Wittenberg-Germany 18-20 march 2004