

# An Enhanced Genetic Algorithm of the BLF2G Guillotine Placement Heuristic for the Orthogonal Cutting-Stock Problem

Slimane Abou-Msabah<sup>1</sup>

<sup>1</sup> Department of Computer Science, University of Science and Technology Houari Boumedienne, Bab Ezzouar, Algiers, Algeria  
slma1g@yahoo.com

**Abstract.** The orthogonal cutting-stock problem tries to place a given set of items in a minimum number of identically-sized bins. As a part of solving this problem with the guillotine constraint, we propose to combine the new BLF2G Bottom Left Fill 2 direction Guillotine placement heuristic with an advanced genetic algorithm. According to the item order, the BLF2G routine makes a direct placement of items on bins to give a cutting format. The genetic algorithm exploits the search space to find the supposed optimal order. Other methods try to guide the evolutionary process by introducing a greedy heuristic to the initial population to enhance the results. We propose to enrich the population by qualified individuals, without disturbing the genetic phase, by introducing a new enhancement to guide the evolutionary process. When we observe that there are no improvements, we inject a qualified individual to the population to avoid premature convergence to a local optima. We generate a set of order-based individuals, to enrich the evolutionary process. Our method is compared with other methods found in literature on existing data sets.

**KEYWORDS:** Cutting and Packing; Guillotine Constraint; Combinatorial optimization; Genetic algorithms; Heuristics; premature convergence; local optimum.

## 1 Introduction

The cutting or packing problem is a combinatorial optimization problem. The objective is to determine a suitable arrangement of various items within a wider set of bins. The main objective is to maximize the use of raw material, thus minimizing losses. This problem is interesting because it is applicable to several fields. For example, in the wood or steel industries, it is necessary to consider how to cut rectangular pieces from large sheets of material. In the transportation and logistics fields, objects of different sizes have to be packed in larger containers of standard size. In floor planning, it is necessary to consider the very-large-scale integration (VLSI) design. If we assign each piece a cost equal to its area, this problem can be formulated as a knapsack problem.

This paper, considers the orthogonal cutting-stock problem, which treats a strip of fixed width and supposed infinite height to generate items of rectangular shape. The production machines can be the guillotine shears, which impose the cut from edge-to-

---

Proceedings of the 21st RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion (RCRA 2014).  
Vienna, Austria, July 17-18, 2014.

edge (guillotine constraint). We keep the original orientations of items to be cut in decorated plates or for the draft layout of pages of newspapers (orientation constraint).

Several placement heuristics are used to solve this problem. We find that the new BLF2G guillotine placement heuristic introduced by Abou Msabah and Baba-Ali (2011) is the most adaptive and promising heuristic. The BLF2G routine packs items in levels to ensure the guillotine constraint and attempts to exploit intra-level residues vertically and horizontally by checking the guillotine constraint.

In this paper, we will combine this routine with an enhanced genetic algorithm. According to the item order, the BLF2G routine makes a direct placement of items on levels to give a cutting format. The genetic algorithm exploits the search space to find a supposed optimal order. We control the evolutionary process, and when we observe that there is no improvement, which means a locally optimal solution has been found, we say that the evolutionary process is stable. After stability is detected, we propose to inject an ordered list of items into the population based on a greedy heuristic to diversify the search in another area of the solution space. We will also propose a set of order-based heuristics to be injected into the population to enhance the ability of the genetic algorithm to find good solutions.

After this introduction to the problem, in section 2 we will discuss the placement routines found in the literature. In section 3, we present the genetic algorithm, propose some order-based greedy heuristics and describe the stability controlled genetic algorithm. The computational experiments will be discussed in section 4, followed by a comparison of our method to other heuristics on data sets found in the literature. We shall end our article with the conclusion, which is presented in section 5.

## 2 State of the art

In this section, we are interested in investigating guillotine placement heuristics from existing methods found in the literature to propose a routine that fits our case.

Lodi et al. (1999) present a new variant of the "Floor-Ceiling" FC routine to check the guillotine constraint. This variant performs the cuttings from edge to edge.

Ben Messaoud et al. (2004) presented a new Shelf Heuristic Filling heuristic (SHF) to apply the FC algorithm by injecting items placed in the ceiling below.

Recently, Abou Msabah and Baba-Ali (2011) proposed a new placement heuristic, which takes care of the guillotine constraint. An item can be laid out on the strip in three possible ways; placement in levels, in BLGsub-levels or in BLFGsub-levels.

## 3 Our contribution

The BLF2G guillotine placement heuristic is the most adaptable to our case. We will combine it with a genetic algorithm. According to the order of items, the BLF2G heuristic packs items directly on a strip to give a layout. The genetic algorithm investigates the solution space to find the best order of items that offers good results.

As introduced in Abou Msabah and Baba-Ali (2011), the genetic algorithm failed in a greedy heuristic for a large data set. They propose to guide the genetic algorithm by introducing sorted lists of items to the initial population; they called it BLF2G+GA<sub>imprv</sub>. There are several sorting policies. We will propose a new sorting policy that promotes characterized items to appear first in the laying out process, which will be discussed below. We will use these sorting policies in turn to guide the evolutionary process, when stability is detected.

### 3.1 The genetic algorithm

We used a real-coded genome (cf. Hopper and Turton 1999) in which the chromosome was defined by the items order. Based on the BLF2G policy, the appearance order of items in the laying out process determined the quality of every individual. We implemented a genetic algorithm approach with a population size of 100, and we fixed the number of generations by 20 times the number of items. Initially we generated a random population with a random ordering item in each individual. At each generation, our BLF2G policy gave the quality of each individual. The genetic operators were defined as follows (cf. Abou Msabah and Baba-Ali 2011):

**Crossover operator:** We used the partially matched cross-over with 1-point (PMX1). The cross-over rate is 0.8. We then made a correction to make the children valid. We corrected child 1 with missing genes according to their order of appearance in parent 2, and replaced the double genes in child 2 by the missing genes according to their order of appearance in parent 1.

|                    |              |                  |
|--------------------|--------------|------------------|
| Parent 1 : 123 456 | 123321       | Child 1 : 123654 |
| PMX1 →             | Correction → |                  |
| Parent 2 : 654 321 | 654456       | Child 2 : 654123 |

**Mutation operator:** This is a permutation between two randomly chosen sites at which the mutation rate is 0.15:

Child : 1|2365|4 → 143652

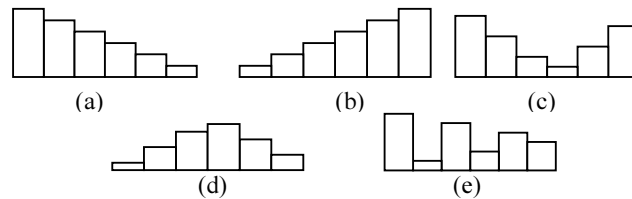
### 3.2 Order-based heuristics

Here, we will present some order-based heuristics ranging from a simple greedy heuristic to a more sophisticated one.

**DH heuristic.** The list of items is sorted according to the decreasing height of items. When two items have the same height we promote the widest one, figure 1 (a).

**IH heuristic.** The list of items is sorted according to the increasing height of items. When two items have the same height we promote the smallest one, figure 1 (b).

**DH-Reverse heuristic.** It is about applying the DH heuristic in two sides. We put the longest item in the right side then the next longest one in the left side, and so on, till the last item, then we concatenate the right side with the inversed list of the left side, to have a DH-Reverse sorted list, figure 1 (c).



**Figure 1.** The greedy routines

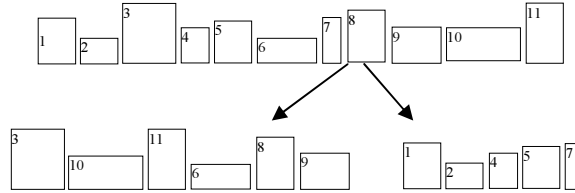
**IH-Reverse heuristic.** It is about applying the IH heuristic in two sides. We put the shortest item in the right side then the next shortest one in the left side, and so on, till the last item, then we concatenate the right side with the inversed list of the left side, to have a IH-Reverse sorted list, figure 1 (d).

**Harmonic heuristic.** It is about applying alternately the IH heuristic and the DH heuristic. The list of items is sorted alternately: take the highest item then the shortest one, and so on, figure 1 (e).

**Decreasing Height Optimization Width heuristic (DHOptW).** Abou Msabah and Baba-Ali (2011) propose this intelligently sorted policy by simulating the BLF2G guillotine placement routine to improve their genetic algorithm. The list of items is sorted alternately: take the longest item then the widest items according to the available width in the level, and so on.

**Divide Rule heuristic (DR).** The longest item will form a level; the largest items will determine the shape of the layout, and the smallest items are favored to fill gaps.

Another possible improvement is to subdivide the list of items into two parts such that the first part contains the largest items (alternately, longer item rather than wider item), and the second contains the other items. Thus, the large items are favored to be first. Figure 2.



**Figure 2.** The Divide Rule policy

|                          | Name            | # of Item | Plates dimension | Optimal height |
|--------------------------|-----------------|-----------|------------------|----------------|
| Abou Msabah and Baba-Ali | Msa17(a, b, c)  | 17        | 200 x 200        | 200            |
|                          | Msa35(a, b, c)  | 35        | 200 x 200        | 200            |
|                          | Msa75(a, b, c)  | 75        | 200 x 200        | 200            |
|                          | Msa150(a, b, c) | 150       | 200 x 200        | 200            |
| Hopper and Turton (2001) | C1(1, 2, 3)     | 16 or 17  | 20 x 20          | 20             |
|                          | C2(1, 2, 3)     | 25        | 40 x 15          | 15             |
|                          | C3(1, 2, 3)     | 28 or 29  | 60 x 30          | 30             |
|                          | C4(1, 2, 3)     | 49        | 60 x 60          | 60             |
|                          | C5(1, 2, 3)     | 73        | 60 x 90          | 90             |
|                          | C6(1, 2, 3)     | 97        | 80 x 120         | 120            |
|                          | C7(1, 2, 3)     | 196or197  | 160 x 240        | 240            |
| Burke et al. (2004)      | N1              | 10        | 40 x 40          | 40             |
|                          | N2              | 20        | 30 x 50          | 50             |
|                          | N3              | 30        | 30 x 50          | 50             |
|                          | N4              | 40        | 80 x 80          | 80             |
|                          | N5              | 50        | 100 x 100        | 100            |
|                          | N6              | 60        | 50 x 100         | 100            |
|                          | N7              | 70        | 80 x 100         | 100            |
|                          | N8              | 80        | 100 x 80         | 80             |
|                          | N9              | 100       | 50 x 150         | 150            |
|                          | N10             | 200       | 70 x 150         | 150            |
|                          | N11             | 300       | 70 x 150         | 150            |
|                          | N12             | 500       | 100 x 300        | 300            |
|                          | N13             | 3152      | 640 x 960        | 960            |

**Table 1.** Data sets found in the literature

## 4 Experimental results

For our experiments, we will compare our new sorting policies combined with the BLF2G guillotine placement heuristic which we call BLF2G+GA<sub>imprv+</sub> to BLF2G+GA and BLF2G+GA<sub>imprv</sub>.

### 4.1 Data sets found in the literature:

To assess the performance of our new algorithm, we use the data sets Msa of Abou Msabah and Baba-Ali (2011), data sets C of Hopper and Turton (2001) and the data set N of Burke et al. (2004). (Table 1).

Table 2 presents the height of the strip by applying our BLF2G+GA<sub>imprv+</sub> policy with regard to BLF2G+GA and BLF2G+GA<sub>imprv</sub>.

The results obtained in the table 2 show clearly that our new method BLF2G+GA<sub>imprv+</sub> outperforms the BLF2G+GA method in all cases except one N5, i.e. guide the evolutionary process by greedy routines improves the result in most cases. In front of the BLF2G+GA<sub>imprv</sub> method, our new method gives good results in few cases, but we lost in most of the cases.

We can conclude that the evolutionary process was disrupted by the greedy routines. This means that the research was focused around these greedy routines (local optimum). To remedy that, we will present in the next section our improvements.

| Name    | BLF2G+GA   | BLF2G<br>+GA <sub>imprv</sub> | BLF2G<br>+GA <sub>imprv+</sub> |
|---------|------------|-------------------------------|--------------------------------|
| Msa17a  | <b>200</b> | <b>200</b>                    | <b>200</b>                     |
| Msa17b  | <b>200</b> | <b>200</b>                    | <b>200</b>                     |
| Msa17c  | <b>200</b> | <b>200</b>                    | <b>200</b>                     |
| Msa35a  | 220        | 215                           | <b>200</b>                     |
| Msa35b  | 215        | 210                           | 210                            |
| Msa35c  | 219        | <b>213</b>                    | 215                            |
| Msa75a  | 215        | <b>205</b>                    | 210                            |
| Msa75b  | 210        | <b>205</b>                    | <b>205</b>                     |
| Msa75c  | 218        | <b>210</b>                    | <b>210</b>                     |
| Msa150a | <b>205</b> | <b>205</b>                    | <b>205</b>                     |
| Msa150b | <b>205</b> | <b>205</b>                    | <b>205</b>                     |
| Msa150c | 219        | <b>212</b>                    | 214                            |
| C11     | 21         | <b>20</b>                     | <b>20</b>                      |
| C12     | <b>22</b>  | <b>22</b>                     | <b>22</b>                      |
| C13     | 21         | <b>20</b>                     | 21                             |
| C21     | <b>16</b>  | <b>16</b>                     | <b>16</b>                      |
| C22     | <b>16</b>  | <b>16</b>                     | <b>16</b>                      |
| C23     | 16         | <b>15</b>                     | <b>15</b>                      |
| C31     | 32         | 32                            | <b>31</b>                      |
| C32     | 34         | <b>33</b>                     | <b>33</b>                      |
| C33     | 34         | <b>30</b>                     | 33                             |
| C41     | 67         | <b>64</b>                     | 65                             |
| C42     | 68         | <b>66</b>                     | 67                             |
| C43     | 64         | <b>62</b>                     | 63                             |
| C51     | 97         | <b>94</b>                     | 96                             |

| Name | BLF2G+GA   | BLF2G<br>+GA <sub>imprv</sub> | BLF2G<br>+GA <sub>imprv+</sub> |
|------|------------|-------------------------------|--------------------------------|
| C52  | 102        | <b>97</b>                     | <b>97</b>                      |
| C53  | 98         | <b>94</b>                     | <b>94</b>                      |
| C61  | 133        | <b>126</b>                    | 128                            |
| C62  | 135        | <b>127</b>                    | <b>127</b>                     |
| C63  | 133        | <b>126</b>                    | 127                            |
| C71  | 263        | <b>250</b>                    | <b>250</b>                     |
| C72  | 266        | <b>248</b>                    | <b>248</b>                     |
| C73  | 267        | <b>248</b>                    | 249                            |
| N1   | <b>40</b>  | <b>40</b>                     | <b>40</b>                      |
| N2   | <b>50</b>  | <b>50</b>                     | <b>50</b>                      |
| N3   | 55         | <b>53</b>                     | <b>53</b>                      |
| N4   | 93         | <b>87</b>                     | <b>87</b>                      |
| N5   | 106        | <b>106</b>                    | 107                            |
| N6   | 106        | <b>103</b>                    | 104                            |
| N7   | <b>116</b> | <b>116</b>                    | <b>116</b>                     |
| N8   | 90         | <b>85</b>                     | <b>85</b>                      |
| N9   | 158        | <b>153</b>                    | <b>153</b>                     |
| N10  | 160        | <b>153</b>                    | 154                            |
| N11  | 158        | <b>153</b>                    | 154                            |
| N12  | 322        | <b>308</b>                    | 311                            |

Table 2 Computational results on data set Msa, C and N

## 4.2 Improvements

Injecting sorted individuals into the population may improve the results, but introducing additional sorted lists without a control may cause the search to converge on a local optimal solution, which disturbs the evolutionary process.

We propose a novel improvement to the genetic algorithm. We introduce a notion of the stability of the evolutionary process, such that there is no improvement after some iteration, which means we have obtained a local optimum solution. After stability is detected, we inject in turn a sorted list of items, from the above described, into the population and continue the evolutionary process, while controlling the stability of the evolutionary process, until the end of the treatment.

## 4.3 The controlled stability genetic algorithm (csGA)

This algorithm benefits from the order-based heuristics cited above and guides the genetic algorithm by introducing order-based heuristics to the population in turns, when stability is detected, in the evolutionary process. The factor of stability is defined by 5 times the number of items. Shown below is the genetic algorithm with controlled stability:

| Name    | BLF2G+GA | BLF2G+GA <sub>imprv</sub> | BLF2G+csGA |
|---------|----------|---------------------------|------------|
| Msa17a  | 200      | 200                       | 200        |
| Msa17b  | 200      | 200                       | 200        |
| Msa17c  | 200      | 200                       | 200        |
| Msa35a  | 220      | 215                       | 200        |
| Msa35b  | 215      | 210                       | 210        |
| Msa35c  | 219      | 213                       | 213        |
| Msa75a  | 215      | 205                       | 205        |
| Msa75b  | 210      | 205                       | 205        |
| Msa75c  | 218      | 210                       | 208        |
| Msa150a | 205      | 205                       | 205        |
| Msa150b | 205      | 205                       | 205        |
| Msa150c | 219      | 212                       | 212        |
| C11     | 21       | 20                        | 20         |
| C12     | 22       | 22                        | 22         |
| C13     | 21       | 20                        | 20         |
| C21     | 16       | 16                        | 16         |
| C22     | 16       | 16                        | 16         |
| C23     | 16       | 15                        | 15         |
| C31     | 32       | 32                        | 31         |
| C32     | 34       | 33                        | 33         |
| C33     | 34       | 30                        | 30         |
| C41     | 67       | 64                        | 63         |
| C42     | 68       | 66                        | 63         |
| C43     | 64       | 62                        | 62         |
| C51     | 97       | 94                        | 94         |

| Name | BLF2G+GA | BLF2G+GA <sub>imprv</sub> | BLF2G+csGA |
|------|----------|---------------------------|------------|
| C52  | 102      | 97                        | 96         |
| C53  | 98       | 94                        | 94         |
| C61  | 133      | 126                       | 125        |
| C62  | 135      | 127                       | 126        |
| C63  | 133      | 126                       | 126        |
| C71  | 263      | 250                       | 249        |
| C72  | 266      | 248                       | 246        |
| C73  | 267      | 248                       | 248        |
| N1   | 40       | 40                        | 40         |
| N2   | 50       | 50                        | 50         |
| N3   | 55       | 53                        | 53         |
| N4   | 93       | 87                        | 87         |
| N5   | 106      | 106                       | 105        |
| N6   | 106      | 103                       | 103        |
| N7   | 116      | 116                       | 103        |
| N8   | 90       | 85                        | 84         |
| N9   | 158      | 153                       | 152        |
| N10  | 160      | 153                       | 152        |
| N11  | 158      | 153                       | 153        |
| N12  | 322      | 308                       | 308        |

Table 3 Computational results on data set Msa, C and N

```

Begin
Generate initial population
Nbr of generation :=0;
Nbr of stability := 0;
While Nbr of generation < maximum generation && optimal not reached
do
    Selection();
    Crossing();
    Elitism();
    ++ Nbr of generation;
    If fitness = previous fitness
        Then ++ Nbr of stability;
        Else Nbr of stability := 0;
    End if
    If Nbr of stability = threshold
        Then Nbr of stability := 0;
        Inject a sorted list of pieces to the population;
    End if
End while
End

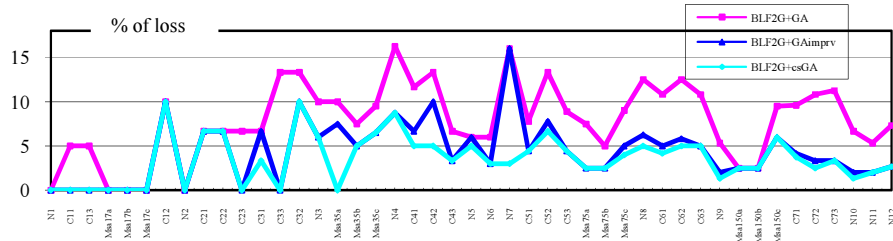
```

For our experiments, we compare our new csGA method combined with the BLF2G guillotine placement heuristic to BLF2G+GA and BLF2G+GA<sub>imprv</sub>, About Msabah and Baba-Ali (2011) with data sets found in literature.

Table 3 presents the height of the strip by applying our BLF2G+csGA policy with regard to BLF2G+GA and BLF2G+GA<sub>imprv</sub>.

With this improvement, our csGA maintains its superiority over the other methods.

To show the comparisons more clearly, we give the percentage of loss over optimal in figure 3. The optimal solution, if exist, equal to the sum of the areas of items.



**Figure 3.** Comparison of the BLF2G+csGA heuristic to BLF2G+GA, and BLF2G+GA<sub>imprv</sub>

Test sets sorted out according to the number of items

As seen in the graph presented in Figure 3, our method outperforms other methods for any size data sets.

## 5 CONCLUSION

We demonstrated the efficiency of our contribution to the rectangular cutting-stock problem. We used the new BLF2G guillotine placement heuristic combined with a genetic algorithm guided by sorted lists of items. Introducing several ordered lists to the evolutionary process might disturb the quality of the treatment to a local optimal solution. To remedy this issue, we propose to guide the genetic algorithm intelligent-

ly. In addition to the DH and DHOptW policies, we developed new ordered policies. We inject these sorted lists into the population when we observe that there is stagnation in the evolutionary process, i.e., we are in a local optimal solution. This will help diversify the search in another area of the solution space.

The comparisons between our csGA combined with the BLF2G routine and other methods are very encouraging. Our method repeatedly reaches the optimum, especially for Msa35a with 35 pieces.

Our method BLF2G+csGA could not be used with the data set N13 of spatial size, which was also the case for the data sets MT01, ..., MT10 of Burke et al. This state is explained by the use of GA, which requires the execution of the routine several times, causing saturation in memory and requiring a lot of computation time.

The BLF2G routine depends completely on the GA to find the order of items that gives the optimal solution. In the future, we intend to enhance the GA in the genetic phase without altering the evolutionary process.

## References

1. Abou Msabah, S.; Baba-Ali, A. R. (2011) A New Guillotine Placement Heuristic Combined with an Improved Genetic Algorithm for the Orthogonal Cutting-Stock Problem, *Proceedings of The IEEE International Conference on Industrial Engineering and Engineering Management 2011, IEEM11*, Singapore, pp. 482-486.
2. Baker, B. S.; Coffman, E. G.; Rivest, R. L. (1980) Orthogonal packings in two dimensions, *SIAM Journal of Computing* 9, 4, pp. 846-855.
3. Jakobs, S. (1996) On genetic algorithms for the packing of polygons, *European Journal of Operational Research* n° 88, pp. 165-181.
4. Liu, D.; Teng, H. (1999) An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles, *European Journal of Operational Research* 112, pp. 413-420.
5. Ramesh Babu, A.; Ramesh Babu N. (1999) Effective nesting of rectangular parts in multiple rectangular sheets using genetic and heuristic algorithms, *International Journal of Production Research* Vol. 37, n°7, pp. 1625-1643.
6. Lodi, A.; Martello, S.; Vigo, D. (1999) Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems, *INFORMS journal on computing* Vol 11, pp. 345-357.
7. Ben Messaoud, S.; Chu, C.; Espinouse, M. L. (2004) An approach to solve cutting stock sheets, *IEEE International Conference on Systems, Man and Cybernetics*, pp. 5109-5113.
8. Burke, E. K.; Kendall, G.; Whitwell, G. (2009) A Simulated Annealing Enhancement of the Best-Fit Heuristic for the Orthogonal Stock-Cutting Problem, *INFORMS Journal on Computing* Vol. 21, No. 3, pp. 505-516.
9. Berkey, J. O.; Wang, P. Y. (1987) Two dimensional Finite bin packing algorithms, *Journal of the Operational Research Society* 38, pp. 423-429.
10. Burke, E. K.; Kendall, G.; Whitwell, G. (2004) A New Placement Heuristic for the Orthogonal Stock Cutting Problem, *Operations Research* vol. 52 no. 4, pp. 655-671.
11. Hopper, E.; Turton, B. (1999) A genetic algorithm for a 2D industrial packing problem, *Computers and Industrial Engineering* vol. 37/1-2, pp. 375-378.
12. Hopper, E.; Turton, B. (2001) An empirical investigation of metaheuristic and heuristic algorithms for a 2D packing problem. *European Journal Operational Research* 128, pp. 34-57.
13. Goldberg, D. E. (1989) *Genetic algorithms in search, optimization, and machine learning*, Reading, MA: Addison-Wesley.
14. Michalewicz, Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs*, Third, Revised and Extended Edition, Springer.