Two Small Scale Problems Formulated and Solved in GENETICA

© Lefteris Virirakis left@genetica-informatics.org

Contents

1 The Traveling Salesman Problem (TSP) 1
1.1 Formulation 1
1.2 Solution method 1
1.3 Implementation and results 1
2 Optimal guillotine cut 5
2.1 Formulation 5
2.2 Solution method 5
a) Definitions and formal representations 5
b) The algorithm 6
2.3 Implementation and results 7

TWO SMALL SCALE PROBLEMS FORMULATED AND SOLVED IN GENETICA

1 The Traveling Salesman Problem (TSP)

1.1 *Formulation*

Given a set **S** of points on the Cartesian Plane, find the shortest route that visits all the points exactly once and returns to the starting point.

1.2 *Solution method*

A variation of the "farthest insertion" method was used :

1. Let **p** be a point in **S**. The list **L = (p, p)** defines a zero-length closed route.
2. Select randomly a point **p′** from the points of **S** that do not belong to **L**.
3. Find two successive points $\mathbf{p_1}$, $\mathbf{p_2}$ in **L** minimizing the sum of distances $\mathbf{p_1}$-**p′** and $\mathbf{p_2}$-**p′**.
4. Insert **p′** in **L** between $\mathbf{p_1}$ and $\mathbf{p_2}$.
5. Repeat the procedure from step 2 until all points belong to **L**.

The magnitude under maximization is $\mathbf{10^6 / (R + 1)}$, where **R** is the route length.

The algorithm described above can solve the problem because the "random" selections performed at step 2 evolve due to the computational system within GENETICA's environment.

1.3 *Implementation and results*

The algorithm was tested in the 49 point configuration presented in Table 1. The population size was set to 50, while the number of tests per computational cycle was set to 5. The rest variables of the computational system were modified during the computational process, such that global search turned to local search in the last computational cycles.

1	(143.11774	61.16726)	17	(30.00000	34.57807)	33	(90.97019	70.50531)
2	(186.86711	197.63012)	18	(78.32744	39.09550)	34	(150.84891	125.62405)
3	(143.99697	97.67054)	19	(43.97676	30.90955)	35	(119.98484	264.63365)
4	(161.67256	253.17332)	20	(79.20667	86.45275)	36	(72.59727	45.58363)
5	(137.32693	171.88984)	21	(85.02779	129.41385)	37	(47.85750	30.00000)
6	(52.01112	34.66902)	22	(153.51693	80.35877)	38	(119.13593	41.21779)
7	(74.50733	35.73017)	23	(121.01566	107.70591)	39	(101.39970	159.70187)
8	(148.72663	45.06822)	24	(102.00606	215.82112)	40	(122.44063	75.20465)
9	(130.02021	56.19505)	25	(116.77110	134.74987)	41	(182.92572	114.19404)
10	(125.04800	231.61698)	26	(156.67004	244.86609)	42	(139.60081	207.24103)
11	(108.16069	99.55028)	27	(41.06620	36.27590)	43	(38.64073	44.61344)
12	(100.06569	83.75442)	28	(66.47297	36.70035)	44	(88.96918	37.67054)
13	(106.09904	122.65286)	29	(164.24962	169.00960)	45	(106.03840	265.02779)
14	(125.47246	125.50278)	30	(50.55584	42.43052)	46	(96.42749	58.95402)
15	(105.82618	72.90045)	31	(102.24861	36.57908)	47	(88.39313	108.13037)
16	(166.88732	80.17686)	32	(86.96817	166.94795)	48	(128.29207	173.67863)
						49	(96.60000	154.60000)

Table 1

The set of points for the TSP application. Each point is defined by an identity number and coordinates in brackets

The content of the main window, which presents the best fitness function, the innovation index function and the evolution diagram (see *GENETICA_Documentation.PDF*, § 5.4.3, 5.6) at the end of the computational process, is presented in Figure 1. Solutions created in different phases of the computational process are presented in Figure 2. The best solution found (Figure 2: Phase 5) is the best known for the particular point configuration.

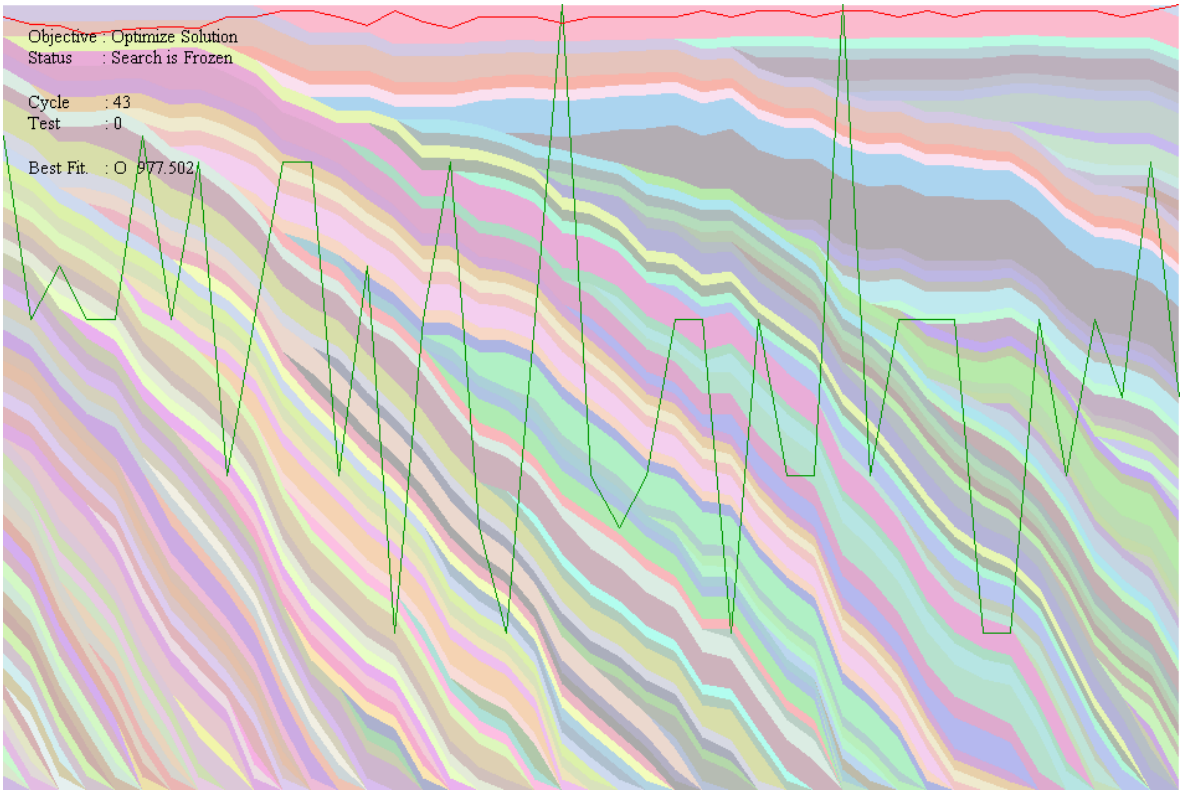

Figure 1

The content of the main window at the end of the computational process of the TSP application. The best fitness function is represented by a red line while the innovation index function is represented by a green line. The evolution diagram appears in the background.

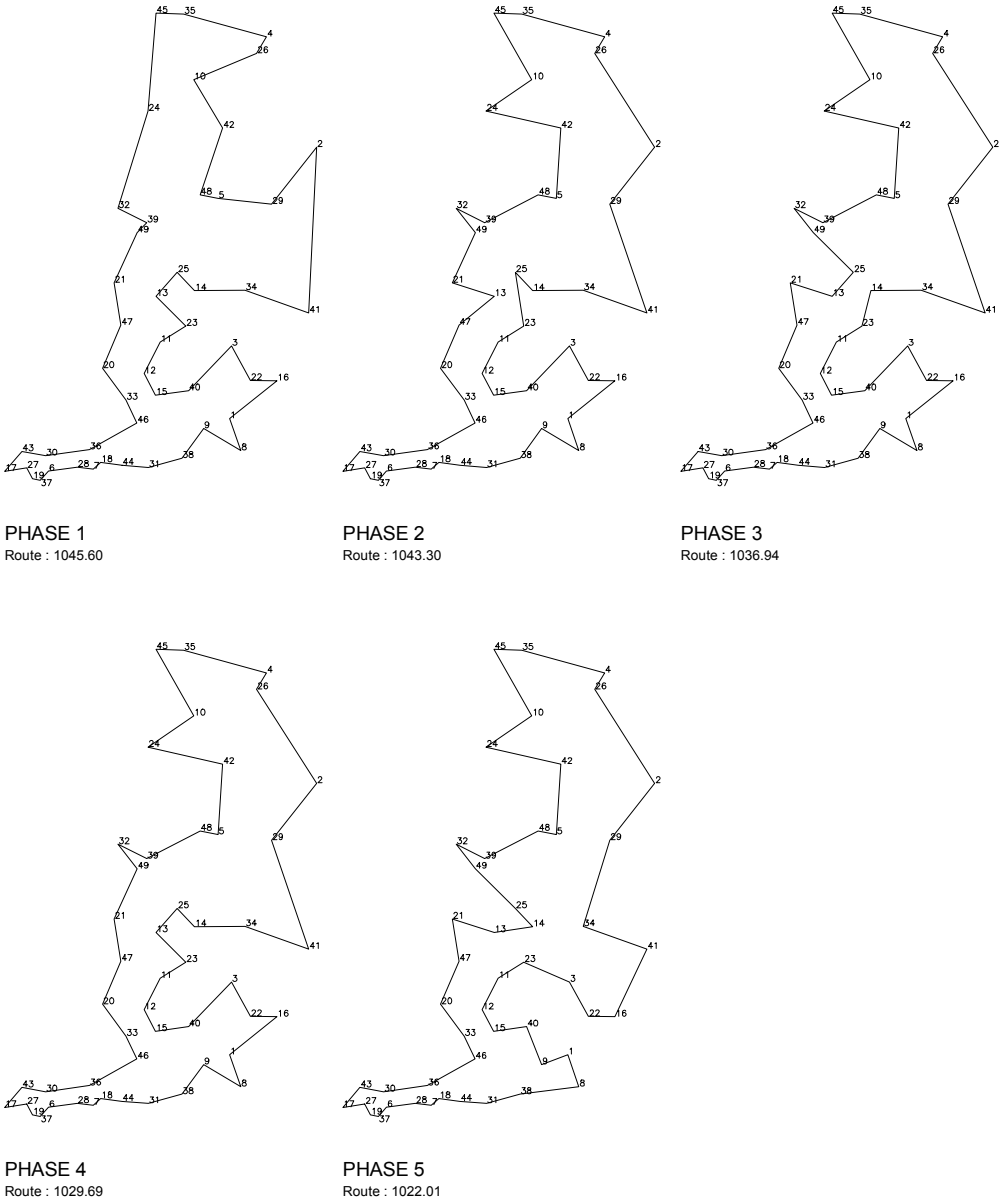

Figure 2

Results of the traveling salesman problem application. Phases 1 and 2 present solutions that represent species of the initial population. Phases 3 and 4 present best solutions produced in different computational cycles. Phase 5 presents the final solution.

2 Optimal guillotine cut

2.1 *Formulation*

Consider a set of rectangles, referred to as "panels", which have different dimensions, and the standard dimensions of other rectangles referred to as "boards". A board can be "cut" into two rectangular pieces by a line parallel to one of its sides (guillotine cut). Any piece derived by a cut either is a panel or is available to a new guillotine cut. A cut pattern is a succession of guillotine cuts, performed on one or more boards, resulting to the production of all the panels. Pieces resulting from a cut pattern without constituting panels, are considered as "waste". It is required a cut pattern that minimizes the total waste area.

2.2 *Solution method*

a) Definitions and formal representations

- Both "boards" and "panels" are considered as "rectangles". A "rectangle" is represented as a list having the form **(X, Y)**, where **X** and **Y** are the dimensions of the "rectangle".
- A "placed rectangle" (either "panel" or "board") is represented as a list having the form **$((x_0, y_0), (x_s, y_s))$**, where **x_0**, **y_0** are the coordinates of the lower left corner while **x_s** is the horizontal dimension and **y_s** is the vertical dimension of the "placed rectangle".
- If **$R = ((x_0, y_0), (x_s, y_s))$** is a "placed rectangle" and **p = (x, y)** is a "panel", then define the "potential placements" of **p** in **R** to be the lists:
  - **$((x_0, y_0), (x, y))$**, if **$x \le x_s$** and **$y \le y_s$**
  - **$((x_0, y_0), (y, x))$**, if **$y \le x_s$** and **$x \le y_s$**

  If there exists at least one "potential placement" of **p** in **R**, then **R** will be characterized as "appropriate for **p** production".
- For each "potential placement" of a "panel" in a "placed rectangle", define two "cutting methods" as shown in Figure 3. Let the "panel" **A** be placed in the "placed rectangle" represented by the external rectangle. Two alternative "cutting methods" produce the "placed rectangles" **B** and **C**. One "cutting method" is defined if either **B** or **C** has a zero area.

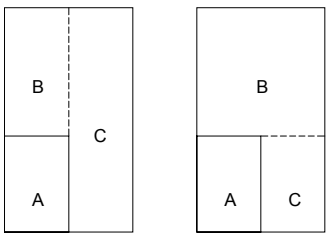

Figure 3

Alternative cutting methods

b) The algorithm

Initialize the following input lists:

(X, Y)	where **X**, **Y** are "board" dimensions
"panel list"	as a list of "panels"
"waste list"	as an empty list
"placed panel list"	as an empty list
"cut list"	as an empty list

1 If the "cut list" is empty, then append a "placed board" to the list

2 Define the first element of the "cut list" as the "cut piece"

3 If the "cut piece" is not "appropriate for the production" of each element of the "panel list" then :

{

4 remove the "cut piece" from the "cut list" and append it at the end of the "waste list"

}

else:

{

5 Identify the "panels" (elements of the "panel list") that can be produced by the "cut piece". Select one of them at random and define it as the "current panel"

6 Choose at random a "potential placement" of the "current panel" in the "cut piece"

7 Choose at random a "cutting method"

8 Append the "placed panel" derived by the cut at the end of the "placed panel list"

9 Remove the "current panel" from "panel list"

10 Append any other "placed rectangle" resulting from the cut—if it exists—at the beginning of the "cut list"

}

11 If the "panel list" is empty, then evaluate the sum of the areas of the "waste list's" elements and stop. Otherwise return to step 1.

1 / (W + 1) is defined as the magnitude to be maximized, where **W** is the total waste area

The algorithm described above can solve the problem because the computational system within GENETICA's environment evolves the "random" selections performed at the steps 5, 6 and 7.

2.3 Implementation and results

The algorithm was tested in the production of the panels presented in Table 2 from boards with dimensions 5 × 10 (abstract units). The population size was set to 300, while the number of tests per computational cycle was set to 20. The rest variables of the computational system were modified during the computational process, such that global search turned to local search in the last computational cycles.

Panel type	Dimensions	Quantity
A	1×2	2
B	1×3	3
C	1×4	1
D	1×5	3
E	1×6	1
F	2×3	2
G	2×4	3
H	2×5	4
I	3×3	2
J	3×4	1
K	4×4	2
L	4×6	1

Table 2

The set of panels for the "guillotine cut" application

The content of the main window, which presents the best fitness function, the innovation index function and the evolution diagram (see *GENETICA_Documentation.PDF*, § 5.4.3, 5.6) at the end of the computational process, is presented in Figure 4. Solutions created in different phases of the computational process are presented in Figure 5. The best solution found (Figure 5: Phase 6) has a zero waste area.

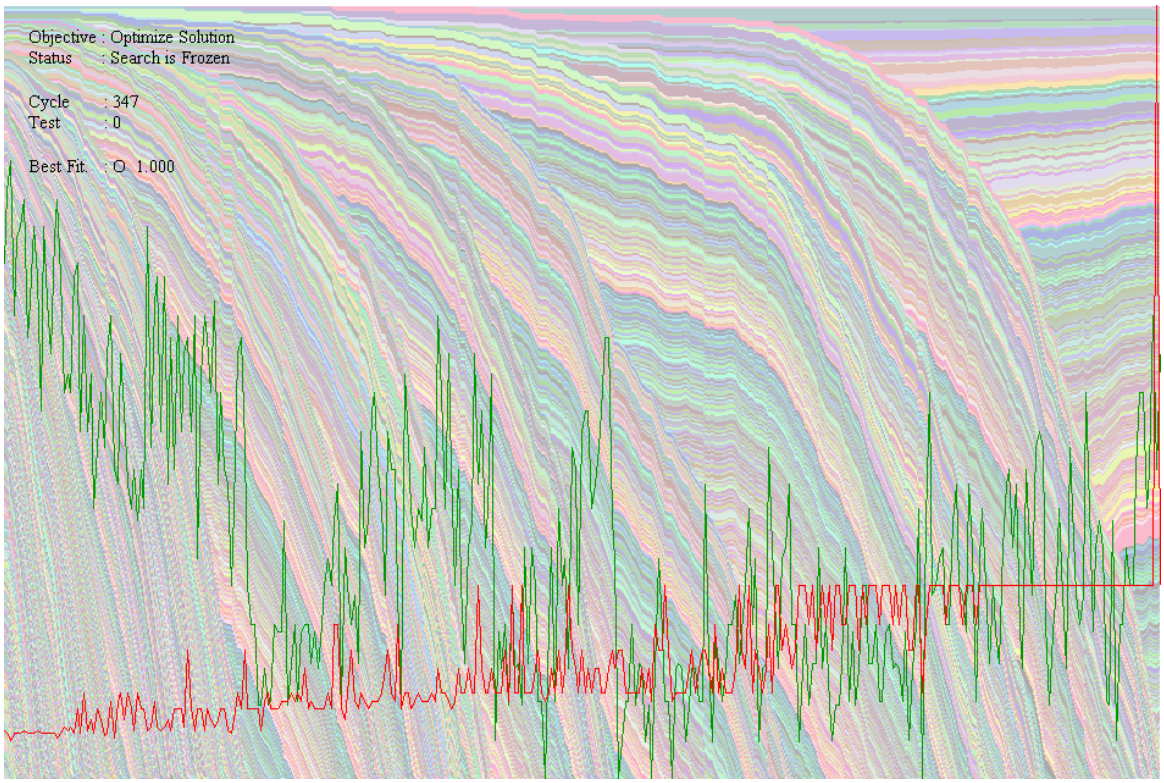

Figure 4

The content of the main window at the end of the computational process of the guillotine cut application. The best fitness function is represented by a red line while the innovation index function is represented by a green line. The evolution diagram appears in the background.

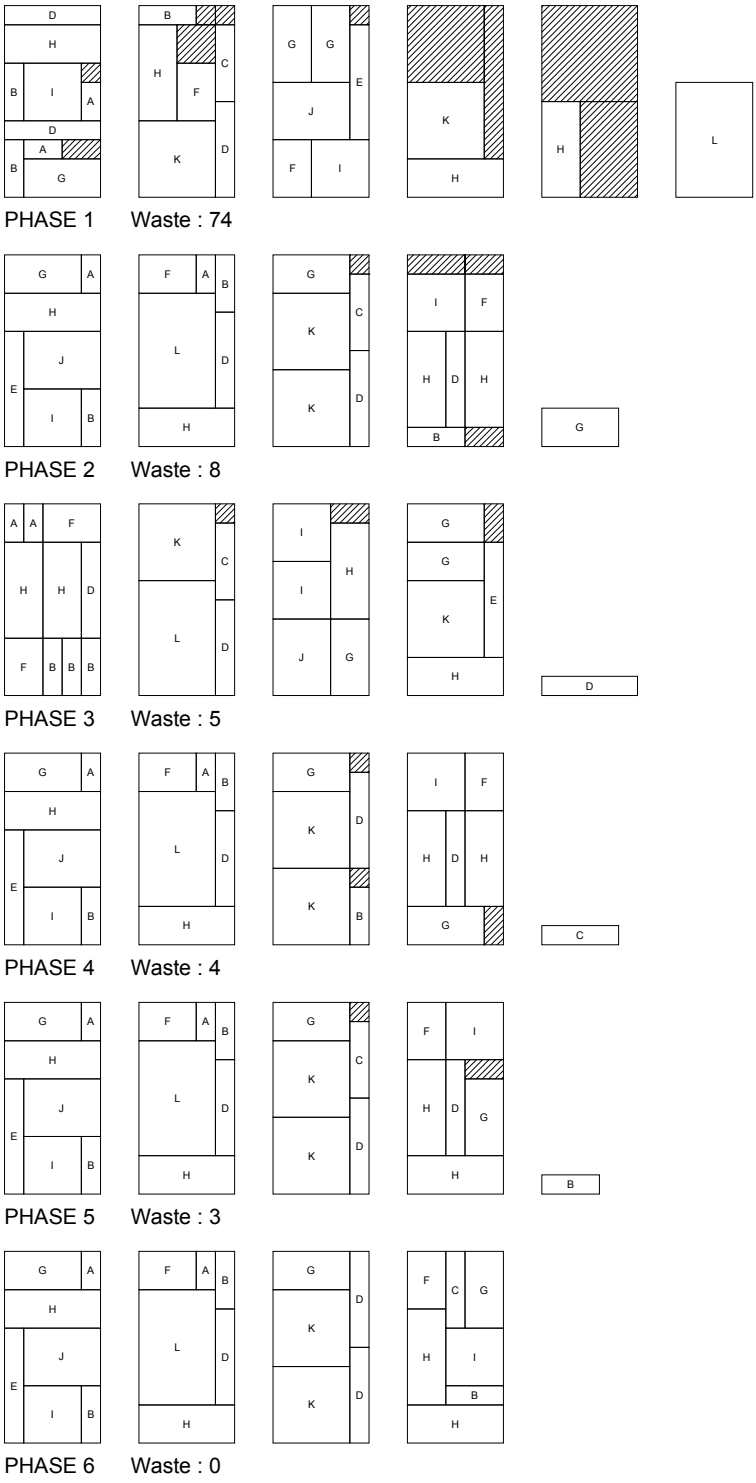

Figure 5

Results of the optimal guillotine cut application are presented with respect to the notation introduced in Table 2. Phase 1 presents a solution represented by a species of the initial population. Phases 2, 3, 4 and 5 present best solutions produced in different computational cycles. Phase 6 presents the final solution. Waste is presented as a shaded area.