Eighth European Conference on Evolutionary Computation in Combinatorial Optimisation – EvoCOP 2008

Naples, Italy – March 26-28, 2008

Optimization of menu layouts by means of Genetic Algorithms

Luigi Troiano+, Cosimo Birtolo*, Roberto Armenise*, and Gennaro Cirillo*

+ Engineering Department - University of Sannio, Benevento, Italy
* Poste Italiane – Chief Information Office (CIO) – Innovative Services Development Centre, Naples, Italy

Generative User Interface Design

- It offers new opportunities for experiencing creativity in engineering problems, by including meta-heuristics in the research into the solution space and producing new and unexpected artifacts

In contrast to traditional design ⟶ *Role of the designer is to explore a solution space in order that a direct relationship between designers' intentions and artifacts*

⇩

Generative design involves the use of assisting techniques and systems for refining and completing the design task

➔ Designers’ Role

- The designer is no longer responsible for meeting a set of recommendations, guidelines and requirements in order to evolve the original idea
- The designers responsible for shaping the constraints on the dynamic process and its behavior
- The human designer free to drive the designing process, keeping him/her focused on creative thought

- Meta-heuristics used in Generative Design include
  - Self-organization
  - Evolutionary Techniques (GA, GP)
  - Interactive Evolution
  - Swarm Systems
  - Ant Colonies
  - Generative Grammars

- *Generative UI design is a new and innovative approach*
- It has been investigated only recently relatively to some aspects

  - Quiroz et al. encodes user interfaces as individuals in an Interactive Genetic Algorithms (IGAs), and run through a number of generations to help explore the space of UI designs.
  - Ichikawa et al. describe re-working of Web page color for Color-Deficient Viewers

➔ Benefits of the generative approach

- ❑ A larger number of alternative can be explored thus pro-actively supporting human creativity and decision-making
- ❑ Different quality attributes and guidelines can be considered one at a time thus facilitating the trade-off between conflicting criteria
- ❑ Designers are free to focus on more value-adding tasks, leaving algorithms to fine-tune their choices
- ❑ Interfaces can be automatically adapted to a larger set of devices, and a more specific set of user preferences

Optimization of menu layouts by means of Genetic Algorithms

Context

- The role of menus in an attractive and usable GUI
- Accessibility, guidelines *(Apple's Human Interface Guidelines, Sun's Java Look and Feel Guidelines*) and preferences in menu system design
- Optimization of menus and menu items through a Genetic Algorithm

Aim

- Design of a Genetic Algorithm able to build a menu system which satisfies a set of preferences and constraints

Real-world applications

- Traditional desktop applications and Web applications
- Design of hierarchical structures
- Optimization of menus on computer, mobile phone such as toolbar, menu bar, context menu, side bar, list box

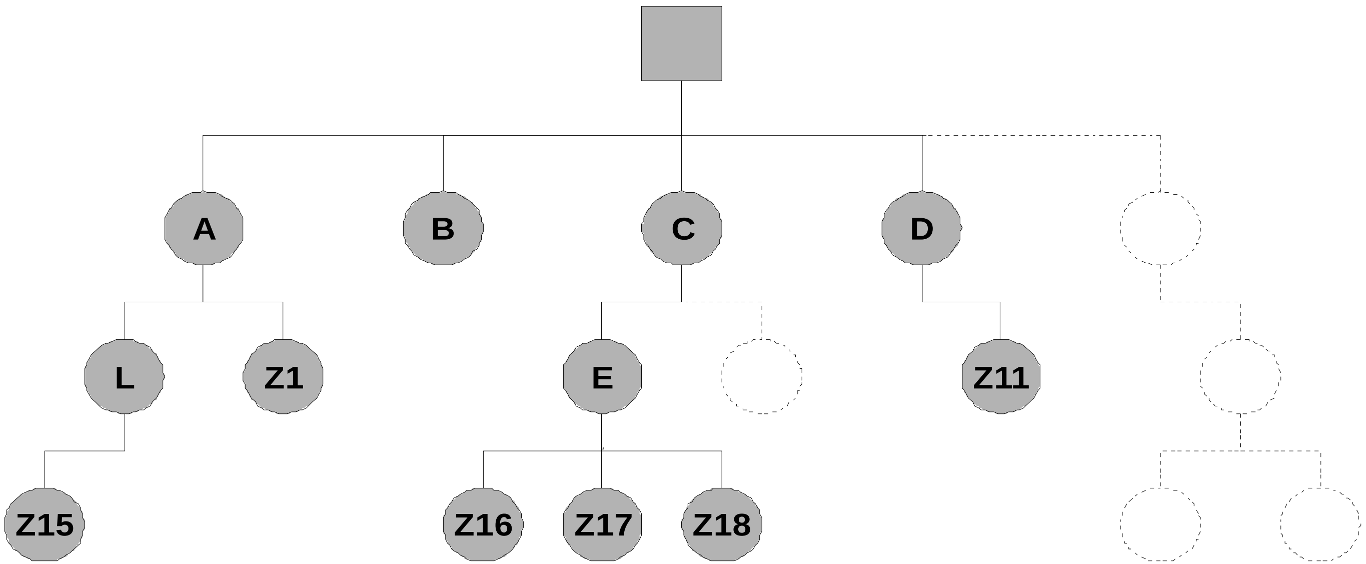
- Designers have to consider many aspects including:
  - How effectively functionalities are retrieved and activated
  - What standard guidelines suggest
  - What are the preferences of users

- These aspects are often conflicting and make menu system design a combinatorial optimization problem as it depends on the arrangement of each item in different positions onto the menu structure

- In our work we show the application of GA as a viable approach to menu design

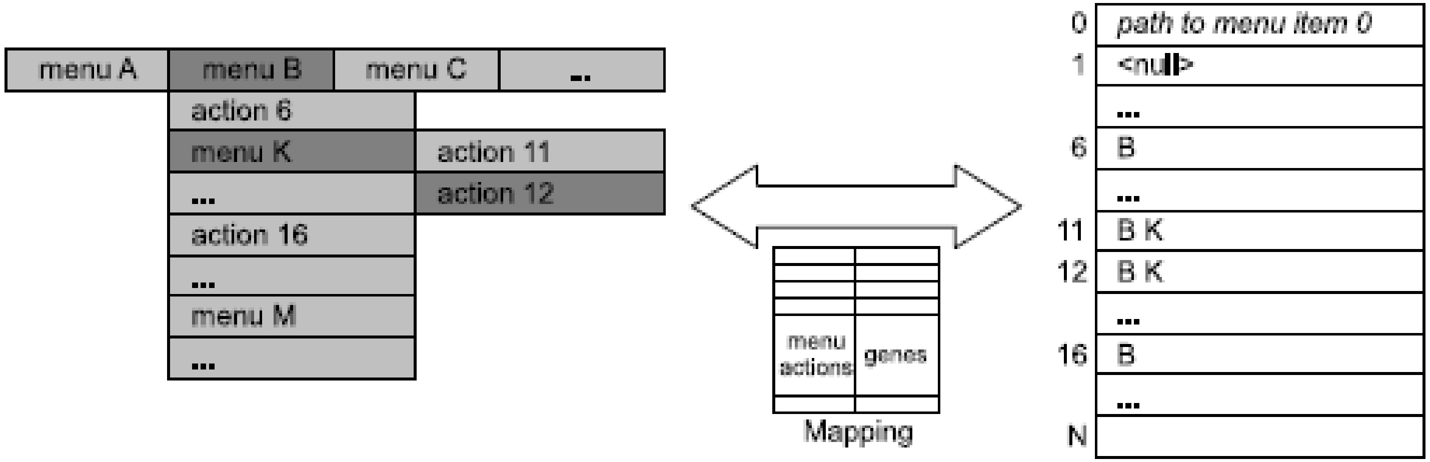
- Menu layout is a hierarchical structure by which the user gains access to application functionalities
  - A menu layout is made of menus
  - Each menu contains a list of items referring to submenus or to actions

We compose a menu layout made of 25 action items (*Z1..Z25*) and 12 submenus (*A …L*)



Chromosome Structure

- Each gene represents the path from root to a menu item
- The mapping between genes and actions is kept by an association table. When the path is empty, the action item is associated to the root (e.g. gene *N* in figure); if the path is null, that action item occurrence is not considered in the menu layout (e.g. gene 1)



➔ Fitness

$$Fitness(x) = \sigma \cdot H(x) + (1 - \sigma) \cdot C(x)$$

- H : Degree of Accessibility

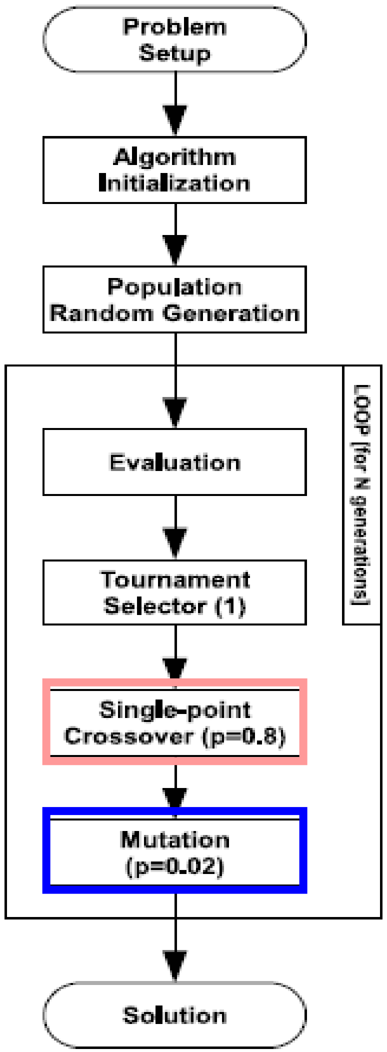
$$H(x) = e^{k(1 - \cdot H_{HAI}(x))} \qquad H_{HAI}(x) = \sqrt{\sum_{i=1}^{L} \sum_{j=1}^{N_i} \log_2(b_{ij}+1) \cdot \log_2(d_i+1)}$$

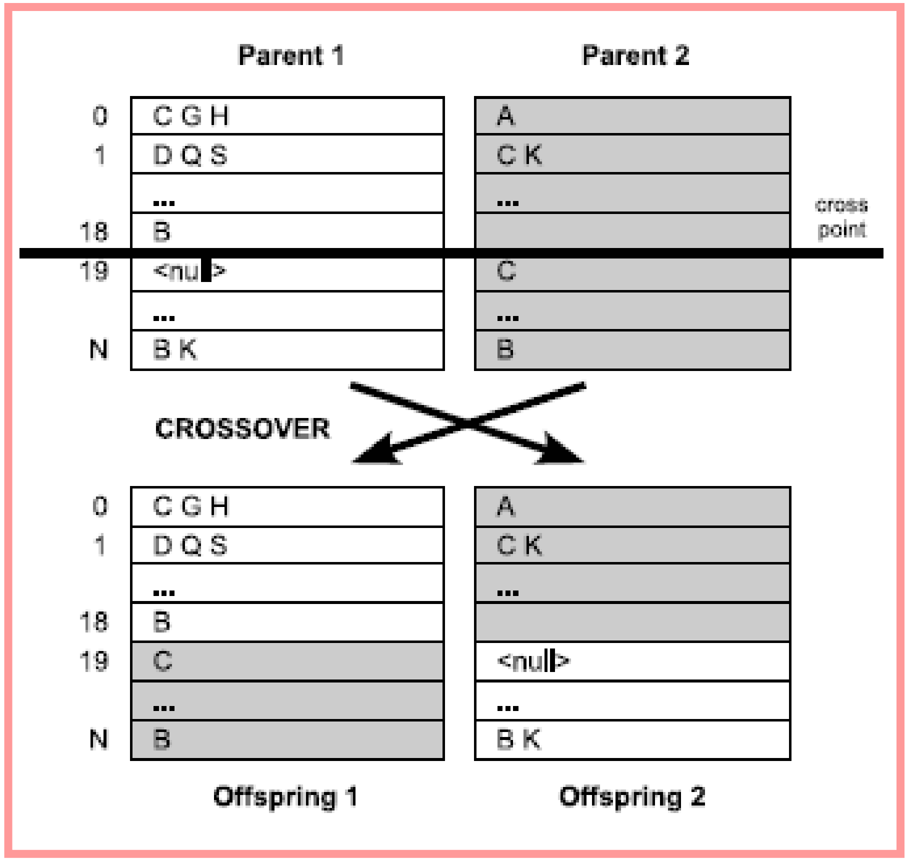
- C : Preferences and Constraints
  - ***Path Ordering*** *(ancestor, successor)* → defines an ordering relation between ancestor and successor along a path
  - ***Menu Ordering*** *(predecessor, follower)* → defines an ordering relation between predecessor and follower whenever they coexist within the same menu
  - ***Number of menu items*** *(menu, min, max)*→ defines the min and max number of items present in menu
  - ***Occurrence (item, min, max)→ defines the min and max number of occurrences of item***
  - ***Level*** *(item, min, max)*→ defines the min and max level for item
  - ***Menu belonging (item, menu)→ item should belong to menu***

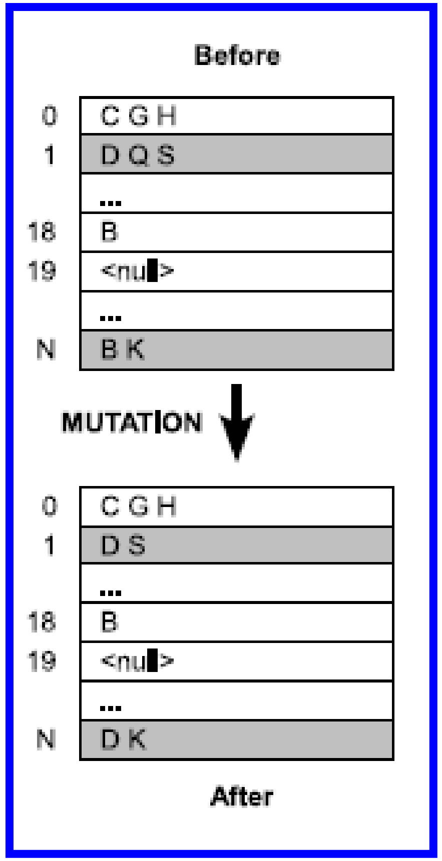
$$C(x) = \frac{\sum_{i=1}^{m} \bar{p}_i c_i(x)}{\sum_{i=1}^{m} \bar{p}_i}$$

m is the number of preferences
p_i is an index of the constraint importance
$c_i(x)$ is the compliance of *x* to the preference c_i

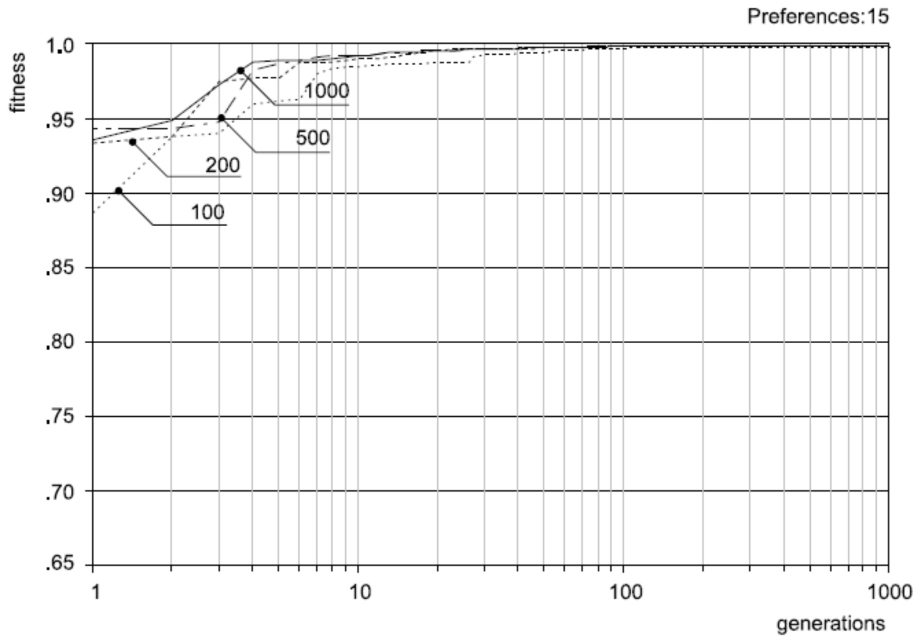
➔ Cross Over e Mutazione

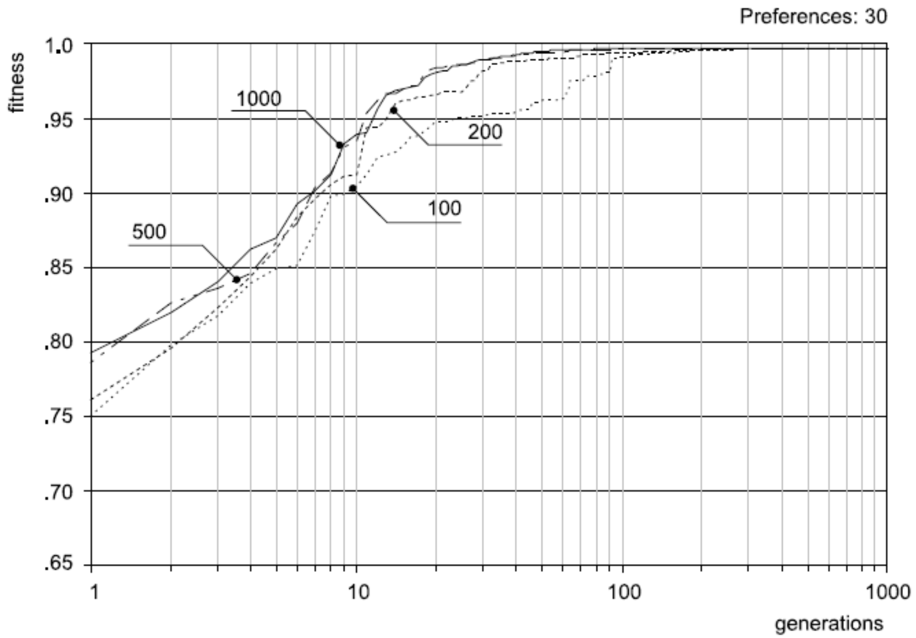


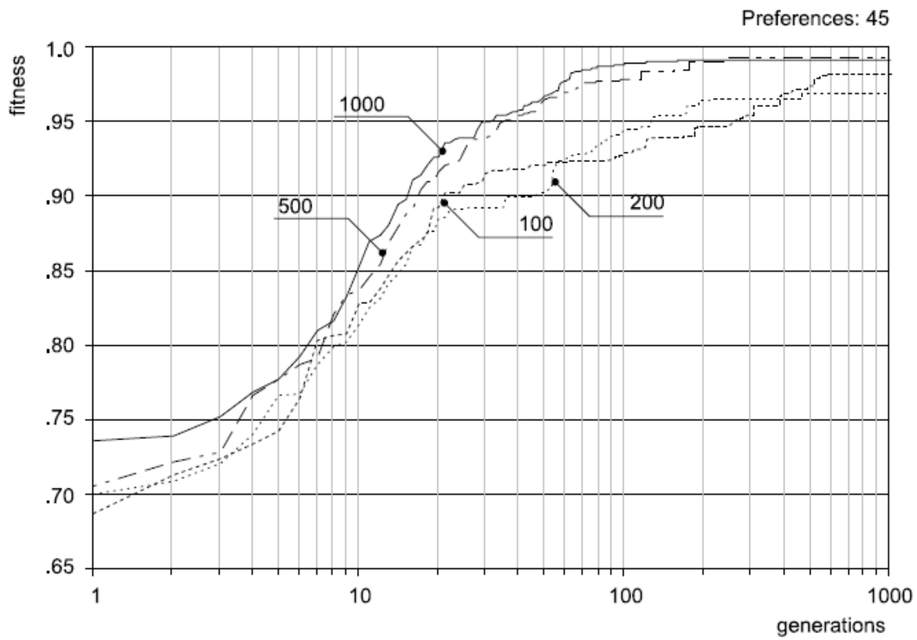


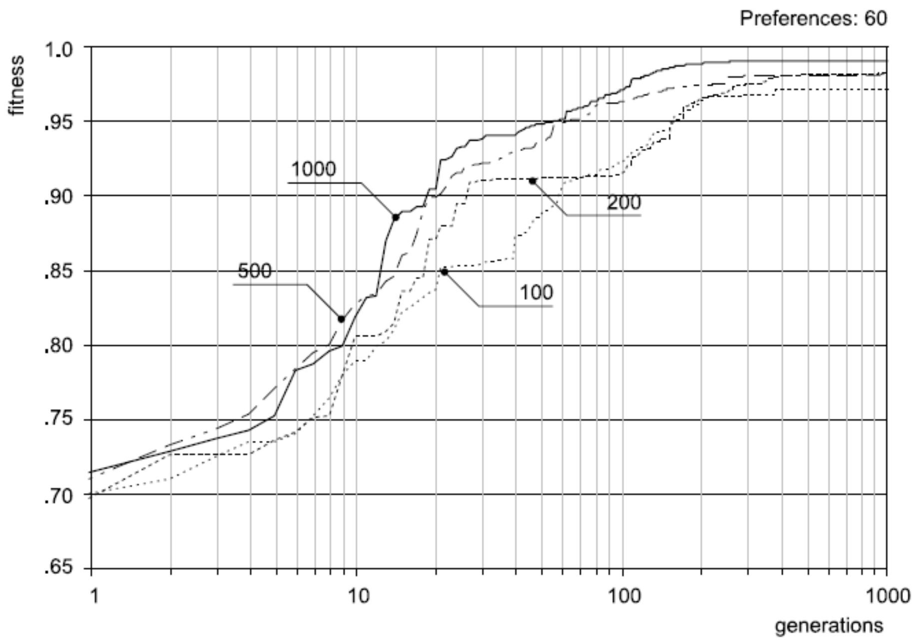


Experimental Results









- Example of Application

1. Given a set of 60 preferences with different priority, research of the optimized tree

2. Finding a menu layout which satisfy 20 preferences and 10 constraints explicitly defined by end-user

	A	B	C	D	E	F	G	H	I	J	K	L	z1	z2	z3	z4	z5	z6	z7	z8	z9	z10	z11	z12	z13	z14	z15	z16	z17	z18	z19	z20	z21	z22	z23	z24	z25
A										3		3	2				3					4															
B								3		2														2				3				4					
C					2						3									3						4											
D						2				4		2						2					3										4				
E									3																			3	3	3							
F							2				4																										
G																																			3	3	3
H														2	2	2																					
I																			2																		
J																					2																
K																									2												
L																											3										

Table 1. Path Ordering and Menu belonging

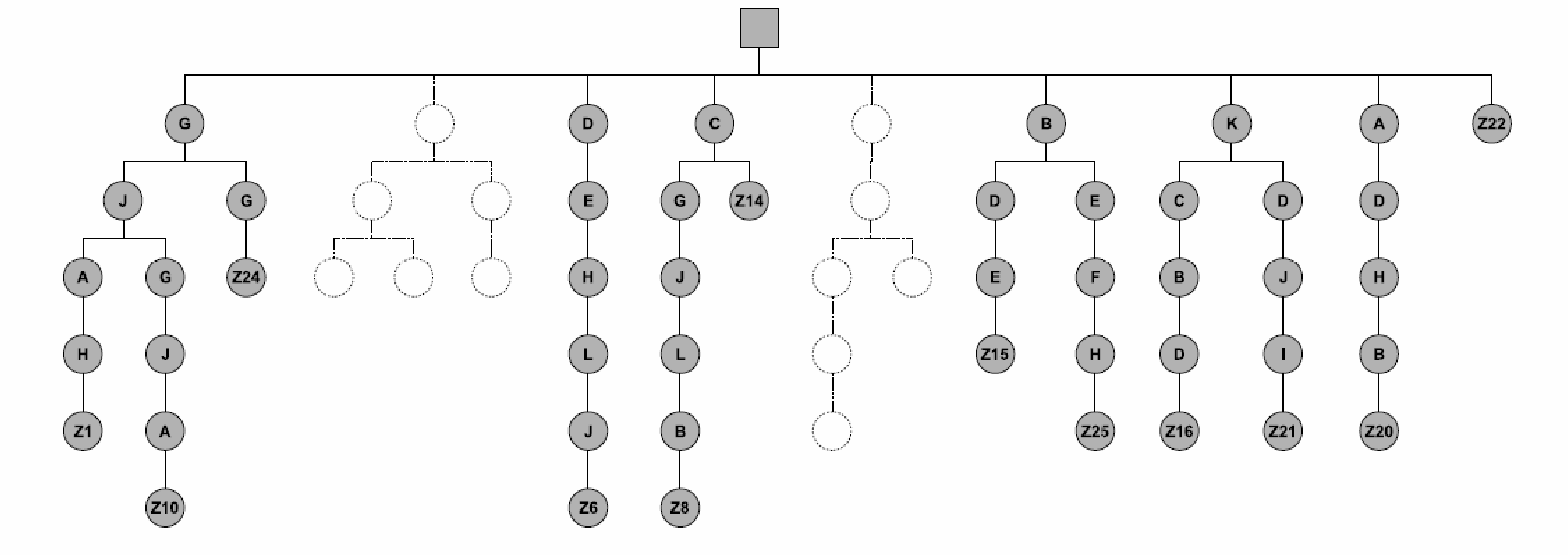
Children Number	Min - Max	Priority
Root	4-5	2
Level 1	2-4	3
Level 2	2-5	3
Level 3	1-2	4
Ordering	A-B-C-D	2
Ordering	F-G	3

Table 2. Menu Ordering and Number of Menu Items

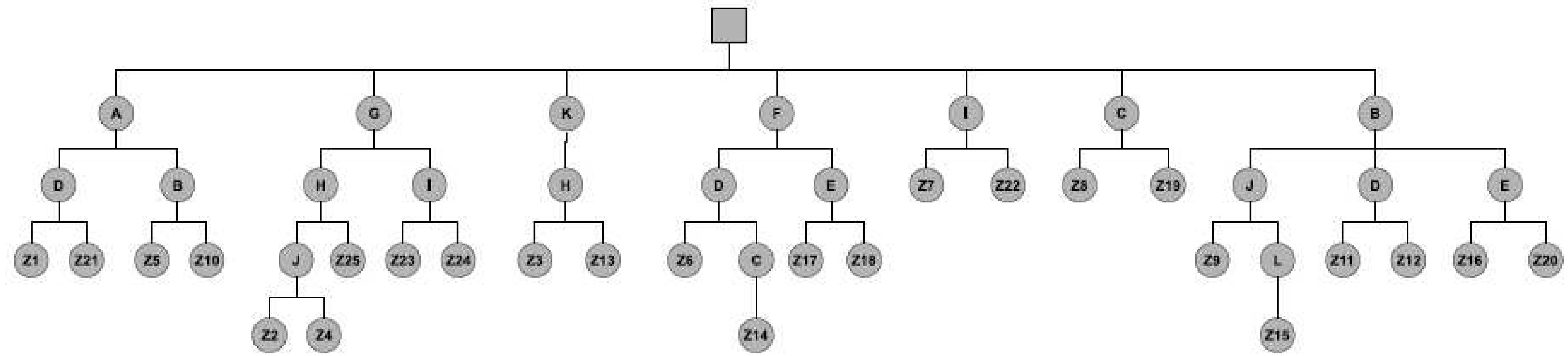
Menus	LEVEL Min - Max	Priority	Repetition Min - Max	Priority
A	1-1	2	1-1	2
B	1-1	2	1-1	2
C	1-1	2	1-1	2
D	1-2	3	-	-
E	1-3	4	-	-
F	-	-	0-1	3
G	-	-	0-1	3
H	2-2	3	0-1	3
I	-	-	0-1	3
J	2-4	3	1-3	2
K	-	-	1-2	4
L	3-4	4	1-2	4

Table 3. Level and Occurance

- Layout of the best individual after 10 generations (fitness 0.7949), given 60 preferences with fixed priority [1,3]



- Layout of the best individual after 1000 generations (fitness 0.9717), given 60 preferences with fixed priority [1,3]



➔ Example of Application

1. Given a set of 60 preferences with different priority, research of the optimized tree

2. Finding a menu layout which satisfy 20 preferences and 10 constraints explicitly defined by end-user

Experimental Results

Tabella 1. Path Ordering and Menu belonging

	A	B	C	D	E	F	G	H	I	J	K	L	z1	z2	z3	z4	z5	z6	z7	z8	z9	z10	z11	z12	z13	z14	z15	z16	z17	z18	z19	z20	z21	z22	z23	z24	z25
A										3		3	2				3					4															
B								3		2														2				3				4					
C					2						3									3						4											
D						2				4		2						2					3										4				
E									3																			3	3	3							
F							2				4																										
G																																			3	3	3
H														2	2	2																					
I																			2																		
J																					2																
K																									2												
L																											3										

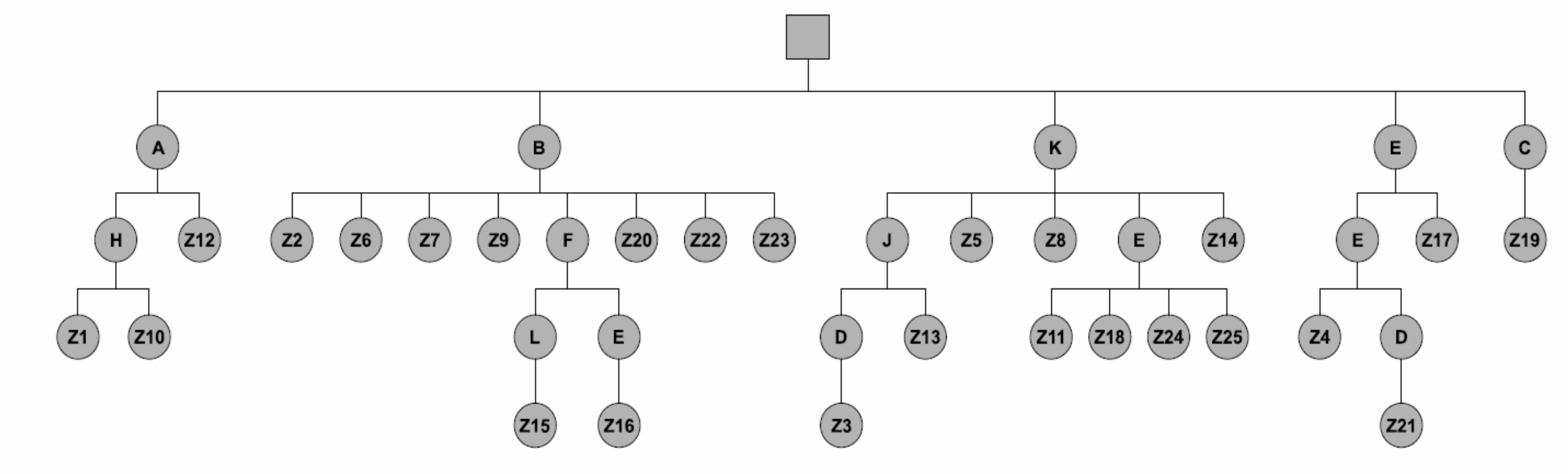
Tabella 2. Menu Ordering and Number of Menu Items

Children Number	Min - Max	Priority
Root	4-5	2
Level 1	2-4	3
Level 2	2-5	3
Level 3	1-2	4
Ordering	A-B-C-D	2
Ordering	F-G	3

Tabella 3. Level and Occurance

Menus	LEVEL Min - Max	Priority	Repetition Min - Max	Priority
A	1-1	2	1-1	2
B	1-1	2	1-1	2
C	1-1	2	1-1	2
D	1-2	3	-	-
E	1-3	4	-	-
F	-	-	0-1	3
G	-	-	0-1	3
H	2-2	3	0-1	3
I	-	-	0-1	3
J	2-4	3	1-3	2
K	-	-	1-2	4
L	3-4	4	1-2	4

- Layout with compatible constraint and preference sets (fitness = 0.9952)



Fitness = 0.9952 $H_{HAI} = 6.147$ H = 0.9498

Conclusions

- We have designed a genetic algorithm for optimizing the layout of a GUI menu system, keeping into the account:
  - ***Accessibility***, as the ease of reaching desired actions
  - ***Guidelines***, as a set of best practices in organizing the menu layout
  - ***Preferences***, as a wish list made by the user in implicit or explicit way

Future Work

- Integrating more *mandatory constraints* into the solution generation (there will be no illegal individuals to deal with)
- Exploring other *Evoutionary Tecnhiques* (e.g. GP)
- Finding out *implicit preferences* by analyzing the usage of generated menus, instead of forcing the user to make explicit all his/her preferences (*Interactive evolutionary computation*, e.g. IGA approach)