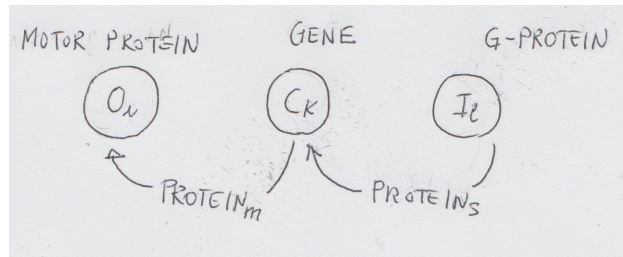I think that it is possible a genetic programming using an evolving graph with evolving edges that can connect evolving nodes.

The simplest method to write an evolution program is to copy the rna: a string of rna code (representable like a graph how we will see), interact with the environment with sensors (the G-proteins) that transmit the signals (here protein-byte), that give information of the environment (proteins representation of the world).

The genetic program (the rna) is a sequence of functions $f(p)$ that work in parallel on the information; so that $q = C_i(p)$ where $q \in C$ or $p \in C$ and C is a protein-byte signal (it is not a G-protein, and it is not a motor protein).

Each function work in parallel and give a byte to all the other functions (when it is activates); the actuators are motor protein that act on the environment $O_i(p) \in E$ where E is the environment, and the sensors are $I_i(p \in E)$ where G-protein read the environment.

It is possible change the function type, changing the domain, or the codomain: so that the input and output protein-byte types give the function types; there exist three possible function for different domains Input, Output and Control, $O_i : C \to E$, $I_l : E \to C$, and $C_k : C \to C$, the protein-byte has a structure that give the type: for example the first bits 01 for Input, 00 for Output and 10 for Control, and the genetic algorithm can change each bits; it is possible to change the edges, and to change the genetic receptors of each function with the evolution.

Each protein-byte from the the functions is passed to all the functions, and only some functions can operate the bytes; the function execute only a restricted number of elements in the domain (generally one) and give a restricted number of output (generally one).

It is necessary to insert a signal latency, after some time the protein is degraded (the node is active for some time, it continue to send protein-byte in the codomain, then it can restart to read the domain)

It is necessary to have an error function to optimize the genetic programming, favoring the optimal program (in this case robots).