Notes on the possibility of using Genetic Algorithms (GAs) combined with Nelder-Mead (NM) for mechanism optimization

Marcello Sanguineti

20 April 2020

- Let the structure of a mechanism (in the following, "structure" for brevity) be modelled as graph, where nodes=joints and edges=links. Note that a structure contains a set of non-optimized parameters, which will be optimized via Nelder-Mead (NM), thus getting a mechanism. So, in tewh following:

mechanism = structure with optimized parameters

- We model a structure as a simple (=with no multiple edges), connected, labelled graph G(V,E) with at most n vertices labelled as 1, ..., n. So, |V|=n.

- The number d_n of simple, connected, labelled graphs with at most n nodes grow super-exponentially with n. More precisely, d_n satisfies the recursion

$$\sum_k \binom{n}{k} k d_k 2^{\binom{n-k}{2}} = n 2^{\binom{n}{2}}$$

See https://mathworld.wolfram.com/LabeledGraph.html for its value. If we have to search in such a set of admissible solutions, for n>>1 brute force (=explicit enumeration) cannot be used. Hence, we have to seek suboptimal solutions in an efficient way. Using a GA combined with NM is one possibility to avoid explicit enumeration and compute with a reduced computational effort (sub)optimal structures.

- The number of admissible solutions to our problem is even larger than the one of simple, connected, labelled graphs, as we have to model also the facts that:

a link can be absent / present with no piston / present with piston and

a joint can be absent / present with motor / present with no motor.

- The very complex problem of finding an optimal mechanism can be divided into two subproblems and for each of them suboptimal solutions can be searched for. It is a two-level optimization:

outer level: generation of a new structure via a GA;

inner level: optimization via NM of the parameters of the new generated structure, thus getting a new mechanism.

- Basic idea of GAs+NM applied in this context: selecting the best current mechanisms (in terms if workspace) within a set of mechanisms (suitably initialized; see below) and combine them is such a way to make them evolve towards a (sub)optimal mechanism.

- Choosing a representation for a structure is a nontrival problem. The most immediate idea is torepresent it via an nxn adjacency matrix A such that

a_{ij} = 0 if there is no link between i and j,

a_{ij} = 1 if there is a link with no piston, and

a_{ij} = 2 if there is a link with piston.

Hence, to represent the structure we need the alphabet {0,1,2} of cardinality 3. If we suppose to have always n nodes, so the data structure has always the same dimension. In such a case, if a node is "pending" (i.e., connected to no other node), it means that the corresponding joint does not exist. By concatenating the rows of the adjacency matrix we get a vector of dimension n^2. We need also another vector v of n elements, such that

v_i= 0 if the node has no motor and

v_i = 1 if the node has a motor (if a node is pending, hence can be eliminated from the structure, already comes out in the adjacency matrix).

OPEN PROBLEM. It is not clear if such a representation is ok. Usually, GAs have a binary representation of the individuals (i.e., the binary alphabet {0,1} is exploited). However also alphabets with larger cardinality can be used. More importantly: one has to be sure that the representation is such that via crossover and mutation (see below) one gets from two feasible structures a new structure that is still feasible. One possibility may be using Ken Stanley's NEAT (NeuroEvolution of Augmenting Topologies) algorithm: https://www.cs.ucf.edu/~kstanley/neat.html. It is designed to evolve Neural Networks (NN) with arbitrary topologies, but NN those are just directed graphs. In our application graphs are not directed, but it may be possible to adapt NEAT to it. A major feature of NEAT is that it provides a way to perform meaningful crossover between two networks that have different topologies. See Section 2.5 at p. 174 of "Advances in Robotics and Virtual Reality", Springer 2012.

- In the mechanism optimization problem, we have:

Individual: mechanism (i.e., structure with optimized parameters)
Population: set of mechanisms.
Initial population: set of mechanisms designed by a human operator.
Genes: variables that characterize a structure (how the nodes are connected, presence/absence of a link with/without piston, presence/absence of a node with/without motor).
Chromosome: string of symbols of the alphabet; it represents a structure.

Once a new structure is obtained via the GA, its parameters are optimized via NM. The fitness function is the associated performance in terms of workspace.

Fitness function. The performance of a mechanism in terms of workspace. E The fitness function determines how well a mechanism performs and represents its score. Once fixed the structure by the GA, the fitness function is a function of the values parameters, to be optimized via NM.

Selection. The idea of selection phase is to select as parents the best mechanisms in the present population and let them pass their genes to the next generation. A typical criterion for selection is Holland's criterion: the selection probability of a mechanism is directly proportional to its fitness score.

To create new mechanisms, two procedures inspired by biological evolutionism are exploited: crossover and mutation with probabilities Pcross ("crossover rate" = crossover probability per pair of mechanisms) and P_{mut} ("mutation rate" = mutation probability per gene), respectively.

Crossover. Crossover is the most significant phase in a GA. It can be done in various ways (one-point crossover, standard crossover, ordered crossover). For instance, in the one-point crossover, in each parent a crossover point is chosen at random and the genes to the right of that point are swapped between the two parents.

Mutation. With a low probability Pmut, some genes of the in the structures obtained after the crossover are changed and the new structures are obtained.

Parameter optimization and fitness evaluation. The parameters of the new structures are optimized via NM and so their fitnesses are computed in terms of workspace and the offsprings are thus obtained.

Population update. Different schemes exist:
-produce as many offspring as parents and replace all parents by the offspring (pure reinsertion).
-produce less offspring than parents and replace parents uniformly at random (uniform reinsertion).
-produce less offspring than parents and replace the worst parents (elitist reinsertion).
-produce more offspring than needed for reinsertion and reinsert only the best offspring (fitness-based reinsertion).
Pure Reinsertion is the simplest reinsertion scheme. It is used in the simple GA. However, it is very likely, that very good mechanisms are replaced without producing better ones and thus good information is lost.

Termination. The algorithm terminates when does not produce mechanisms whose performances are significantly better. Then best mechanism at that generation is chosen as a (sub)optimal solution.

```
Algorithm GNM (Genetic Nelder Mead)
{
        Initialization;
        Evaluation;
        while termination criterion has not been reached
        {
                Selection_and_Reproduction;
                Crossover;
                Mutation;
                Parameters optimization and evaluation via Nelder Mead;
                Reinsertion.
        }
}
```

In the Initialization phase one has to set

- the genes value (Genes) , i.e., the number of variable slots on a chromosome;
- the codes value (Codes) , i.e., the number of possible values for each gene, i.e., the number of symbols used to represent individuals;
- the population size (PopSize), i.e., the number of chromosomes in each generation;
- the crossover probability (Pcross), i.e., the probability that a pair of chromosomes will be crossed;
- the mutation probability (Pmut), i.e., the probability that a gene on a chromosome will be mutated randomly;
- termination criterion 1: the maximum number of generations (MaxGen), i.e., the maximum number of chromosome populations that will be generated before the top scoring chromosome will be returned as the (sub)optimal solution;
- termination criterion 2: the generations with no change in highest-scoring (elite) chromosome (GensNoChange), i.e., the number of generations during which the elite chromosome has not changed before that it will be returned as the (sub)optimal solution.