Coevolution

Petr Pošík

Dept. of Cybernetics
CTU FEE

What is "coevolution"?

Coevolution in EAs:

- ✔ The fitness of individuals in a population
  - ✘ is not given by the characteristics of the individual (only), but
  - ✘ is *affected by the presence of other individuals in the population*.
- ✔ It is closer to the biological evolution than ordinary EAs are.

Coevolution can help in:

- ✔ dealing with increasing difficulty of the problem
- ✔ providing diversity in the system
- ✔ producing not just high-quality, but also robust solutions
- ✔ solving complex or high-dimensional problems by breaking them into nearly decomposable parts

Types of coevolution

By relation type:

- ✔ cooperative (synergic, compositional)
- ✔ competitive (antagonistic, test-based)

By the entities playing role in the relation:

- ✔ 1-population
  - ✘ intra-population
  - ✘ individuals from the same population cooperate or compete
- ✔ N-population
  - ✘ inter-population
  - ✘ individuals from distinct populations cooperate or compete

1-population competitve coevolution

Example: The goal is to evolve a game playing strategy

- ✔ successful against diverse opponents!!!

How would you proceed in an ordinary EA?

Problem: fitness evaluation

- ✔ by playing several games against human player? Against conventional program?
  - ✘ Problem: No learning gradient! Needle in a haystack. All randomly generated players will almost surely loose against any advanced player.
- ✔ by playing several games against internet players?
  - ✘ A bit better. . . but beware (Blondie24)

Solution: Intra-population competitive coevolution

- ✔ by playing several games against other strategies in the population.
- ✔ All individuals of the same type.
- ✔ In the beginning, all are probably quite bad, but some of them are a bit better.
- ✔ The fitness (the number of games won) may not rise as expected since your opponents improve with you.

2-population competitive coevolution

Example: The goal is to evolve a sorting algorithm

- ✔ able to sort any sequence of numbers
- ✔ correctly and quickly.

How would you proceed in an ordinary EA?

Problem: fitness evaluation

- ✔ Test all possible input sequences? Slow, intractable.
- ✔ Test only a fixed set of sequences? Which ones?

Solution: Inter-population competitive coevolution

- ✔ 2 populations, 2 species:
  - ✘ sorting algorithms
  - ✘ test cases (sequences to sort)
- ✔ Fitness evaluation:
  - ✘ Algorithm: by its ability to sort. How many sequences is it able to sort correctly? How quickly?
  - ✘ Test case: by its difficulty for the current sorting algorithms. How many algorithms did not sort it?
- ✔ Predator-prey relationship

N-population cooperative coevolution

Example: The goal is to evolve a team consisting of

- ✔ a goalie, back, midfielder, and forward
- ✔ so that they form a good team together.

How would you proceed in an ordinary EA?

Fitness evaluation:

- ✔ by simulating a number of games between teams

Problem: Evolution

- ✔ Represent all 4 strategies in 1 genome, evolve them all in 1 population.
- ✔ Theoretically possible, but the space is too large.
- ✔ May result in a team of players which wouldn't perform well if substituted to another team.

Solution: N-population cooperative coevolution

- ✔ 4 separate populations
- ✔ Evolve players which would play well with any other team members

Cooperation:

- ✔ symbiotic relationship
- ✔ good performance of the team ⇒ high contribution to fitness of all members

1-population cooperative coevolution

Example: **Niching** methods for

- ✔ diversity preservation
- ✔ maintaining several stable subpopulations in diverse parts of the search space

Examples of niching methods:

- ✔ fitness sharing
- ✔ crowding

Principle:

- ✔ better individuals similar to others already in population are thrown away in favour of worse, but diverse individuals
- ✔ the selection process is affected by the presence of other individual in the neighborhood

Fitness in coevolution

Some important classifications of fitness

- ✔ by its time-dependence:
  - ✘ **static**: does not change with time
  - ✘ **dynamic**: changes with time
- ✔ by the stochastic element:
  - ✘ **deterministic**: generates the same ordering of a set of individuals
  - ✘ **stochastic**: can generate different orderings of the same set of individuals
- ✔ by the role of other individuals in evaluation:
  - ✘ **absolute**: measured independently of other individuals
  - ✘ **relative**: measured with respect to individuals in the current population
- ✔ by its role in the EA:
  - ✘ **internal**: optimization criterion used by selection
  - ✘ **external**: used to measure the progress of the algorithm

Ideally, external fitness

- ✔ should be **static**, **deterministic** and **absolute**
- ✔ can easily be used as internal fitness

External fitness in coevolution:

- ✔ impossible (hard) to define
- ✔ often, it is **relative**, but measured with a carefully chosen, large enough set of other individuals (**static**) sufficiently many times (almost **deterministic**)

Internal fitness in coevolution:

- ✔ **relative**: affected by other individuals
- ✔ **dynamic**: affected by evolving individuals (needs re-evaluation)
- ✔ **stochastic**: usually evaluated against a smaller number of individuals

"Fitness" in competitions

Football league:

- ✔ all teams play against all others
- ✔ points awarded for win, draw, and loss
- ✔ teams sorted by the earned points

Tennis players:

- ✔ tournaments divided to various levels, with different point amounts
- ✔ points awarded to players by their final standings in tournament

Golf players:

- ✔ tournaments have different prize money to distribute to tournament winners
- ✔ highly paid tournaments attract more players and are harder to win
- ✔ players sorted by the won prize money

Chess Elo ratings:

- ✔ each player is assigned a level, based on historic results
- ✔ matches between players of different levels
- ✔ the player's level increases (decreases) if she recently won more (less) matches than expected

None of these systems is static:

- ✔ Is Pete Sampras better than Roger Federer?
- ✔ Is Arnold Palmer better than Tiger Woods?
- ✔ ...

The same holds for fitness assessment in coevolution!

Problems with fitness assessment: 1-pop. competitive coevolution

Cycles, etc.

- ✔ What if A beats B, B beats C, but C beats A?
- ✔ What if A beats B, but B beats far more individuals than A?
- ✔ The quality assessment depends on what we really want:
  - ✘ A player that beats the most other players?
  - ✘ A player that beats the most other "good" players?
  - ✘ A player that wins by the most total points on average?
- ✔ Often, other tests are executed.
- ✔ But, do you want to spend your fitness budget
  - ✘ on evaluating current individuals more precisely, or
  - ✘ on searching further?

2 competitive populations (illustration)

Lotka-Volterra model (Predator-prey population dynamics):

$$\frac{dx}{dt} = \alpha x - \beta xy$$
$$\frac{dy}{dt} = -\gamma y + \delta xy$$

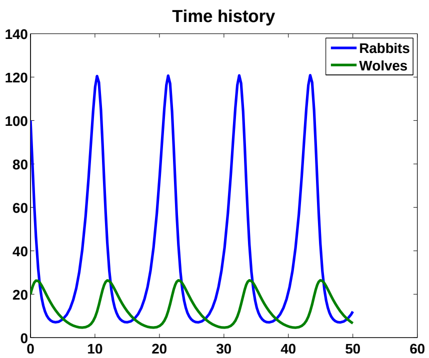
where x is the number of prey (rabbits) and y is the number of predators (wolves).

Assumptions:

1. The prey population has always food enough.
2. The predators eat only the prey.
3. The rate of change of population is proportional to its size.
4. The environment is static.

Meaning:

- ✔ The change of the prey population (dx/dt) is composed of
  - ✘ increase due to the newly born individuals (proportional to the population size, αx) and
  - ✘ decrese caused by the predation (which is proportional to the rate of predator-prey meetings, βxy).
- ✔ The change of the predator population (dy/dt) is composed of
  - ✘ decrease due to natural death (proportional to the population size, γy) and
  - ✘ increase alowed by the food suply (proportional to the rate of predator-prey meetings, δxy).



Problems with fitness assessment: 2-pop. competitive coevolution

Arms races

- ✔ one population learns a trick and forces the second population to learn a new trick to beat the first one...
- ✔ one population may evolve faster than the other:
  - ✘ all individuals from that population beat all the individuals from the other
  - ✘ no selection gradient in either population ⇒ uniform random selection
  - ✘ external fitness in both populations drops until the gradient re-emerges
- ✔ not exactly what was shown by Lotka-Volterra, but similar
- ✔ Solution:
  - ✘ detect such situation (but how?)
  - ✘ delay the evolution of the better population until the worse one catches up

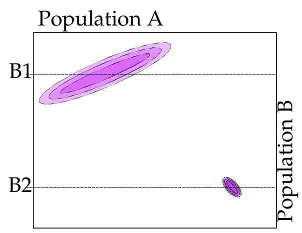
Problems with fitness assessment: N-pop. cooperative coevolution

Hijacking

- ✔ a really good forward takes over one population, any team will play well thanks to him
- ✔ members of all other populations have almost the same fitness ⇒ uniform random selection
- ✔ Solution: apply some form of *credit assignment*

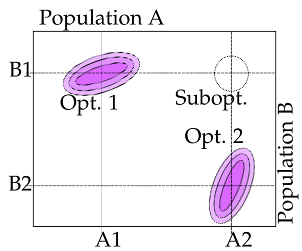
Relative overgeneralization

- ✔ when evaluated by average score, worse (but more robust) individual B1 will have higher score than better (but volatile) B2
- ✔ use maximum score (more tests needed)
- ✔ but again, the choice depends on what we want — a player able to get the highest score, or a player that would compare well with the most other opponents?



Miscoordination

- ✔ when the team components are not independent
- ✔ Pop. A evolved A2 (but not A1), pop. B evolved B1 (but not B2)
- ✔ Neither A2 nor B1 survives



Summary

Coevolution

- ✔ can be cooperative or competitive (or both)
- ✔ can take place in 1 population or in more populations
- ✔ fitness is not fixed during evolution
- ✔ introduces new unexpected dynamics to the system (new issues to be solved)

Appropriate when

- ✔ no explicit fitness function can be formed
- ✔ there are too many fitness cases
- ✔ the problem is modularizable (divide and conquer)