SANDBOX LEARNING: TRY WITHOUT ERROR?

Adaptivity

An adaptive system reacts to changing environmental conditions with different behaviors. In the simple case, the system selects from a set of predefined alternative behaviors. This limits the variability of possible responses. It is therefore desirable that the system learns additional alternatives.

Problem Statement

Learning can be achieved through observation of the outside world and subsequent correction of a world model as e.g. in an autonomous vehicle that laser-scans the environment to update its map. Learning can also be achieved through trial and error: The system carries out actions with a probability of success <1. A reward (or penalty) assigned to this action influences the probability for future re-enactment of the same action (reinforcement learning).

Natural systems learn on the collective (genotype) or on the individual level (phenotype). For collective learning, nature had very long time spaces available, and the populations involved are huge. Moreover, nature works with redundancy and with a total disregard of detrimental outcomes: failures simply die out. Individuals on the other hand are forced to react immediately, if they want to survive.

Learning in technical systems underlies different boundary conditions: They, too, have to react immediately (in the order of ms), but errors are not acceptable (4-way green of a traffic light). Genetic Algorithms (GA) mimic nature by trying out thousands of (pseudo random) alternative solutions and sorting out the less successful ones. This requires an assessment of the success of the tried solutions. If this assessment takes place in the real world then (1) we cannot avoid errors, and (2) the learning times become prohibitive (example traffic control: thousands of tries x reaction time of 15 minutes).

A first Solution

To overcome these problems we propose a 3-level architecture:

Level 0 is the so-called “productive system”. It is observable, parameterized and acts in real time.

Level 1 is responsible for short time adaptation. It supplies parameters adapted to the current situation to level 0. For that it selects from a given set of parameters. Of those we assume that they might not be optimal but at least they will not lead to illegal situations. In case of more than one acceptable parameter set we use a fitness value assigned through a reinforcement mechanism based on the previous success of the parameter sets. Level 1 is implemented as a Learning Classifier System (without the GA).

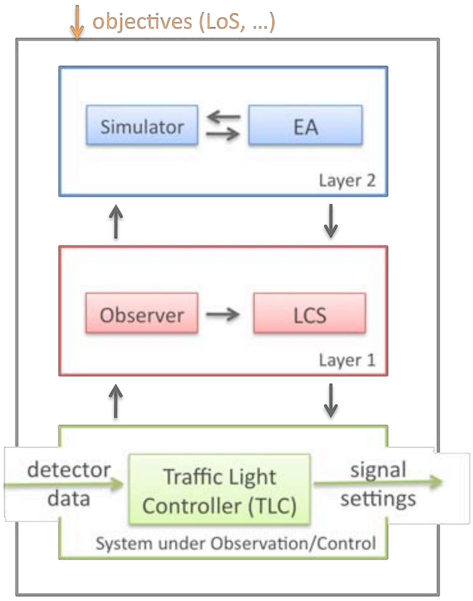
So far we have a fixed behavioral repertoire on level 1 limited by the pre-supplied parameter sets (by the designer – or the traffic engineer). Therefore level 2 has the task to invent new parameter sets for situations so far not adequately covered by level 1. If level 1 encounters a new situation, for which no adequate parameter set is available, it (1) enacts the closest parameter set (best effort – it must be capable of acting immediately), and (2) it asks the level 2 learning mechanism to come up with parameters better suitable for this situation.

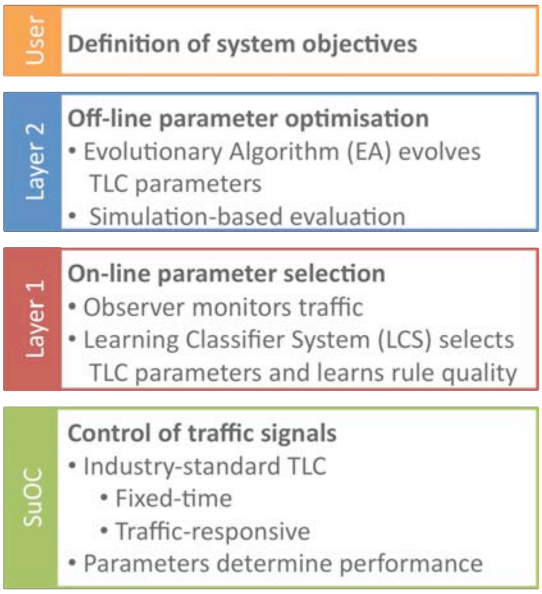
Level 2 comprises a “sandbox” where we can learn very fast and without negative effects in case of errors. It contains a GA, which learns against a simulation of the real world. This simulation can be much faster than real time (x1000 in case of our traffic simulations). Optimized parameter sets are then added to the behavioral repertoire on level1 for usage as soon as the same situation reoccurs.

We have implemented this 3-level architecture for a learning adaptive traffic light controller and for the node controller of a communication network.

The talk will present some results of both applications.

Dagstuhl Seminar Proceedings 09201
Combinatorial Scientific Computing
http://drops.dagstuhl.de/opus/volltexte/2009/2123




User
Definition of system objectives

Layer 2
Off-line parameter optimisation
- Evolutionary Algorithm (EA) evolves TLC parameters
- Simulation-based evaluation

Layer 1
On-line parameter selection
- Observer monitors traffic
- Learning Classifier System (LCS) selects TLC parameters and learns rule quality

SuOC
Control of traffic signals
- Industry-standard TLC
  - Fixed-time
  - Traffic-responsive
- Parameters determine performance

Open Problems and extensions

Although the results so far are quite promising there is a range of remaining problems leading to future work:

- Level 2 learning cannot be more accurate than the simulation. The model must be continuously adapted to reality in order to avoid unrealistic learning.
- Depending on the quality of the evaluation function on level 2 we might fail to catch illegal parameter sets. We are planning to “filter” parameter sets by verification mechanisms before implementing them into level 1.
- Level-2 simulation can take only the local environment into account. The non-local network has to be abstracted.
- So far we cannot handle multi-step problems. This is due to the nature of the Learning Classifier System, which maps a single situation to a single action (single rule). We will investigate different learning mechanisms for level 1.
- We are working on collaboration/cooperation schemes between the learning nodes.
- We are planning to derive a generic “sandbox learning architecture” and apply it to additional similar scenarios with a decentralized network structure (more examples from communication networks, different communication protocols, logistic systems etc.)

The work reported here is carried out in close cooperation between Leibniz University Hannover (Sven Tomforde) and KIT - University of Karlsruhe (Hartmut Schmeck, Holger Prothmann).