# 4.3. Reinforcement learning for forming coalitions: the DFG algorithm

Weiß (1995)

DFG: Dissolution and Formation of Groups

Basic Problems tackled:

- How can several agents learn what actions they can perform in parallel?
- How can several agents learn what sets of actions have to be executed sequentially?

# Reinforcement Learning (I)

Watkins (1989), Sutton (1987)

Let's use our single agent definition:

Then an agent Ag has in Dat for each pair (s,a) Sit × Act an evaluation e(s,a). Its decision function then selects always the action a in a situation s, for which e(s,a) is optimal.

Then learning is performed by getting a feedback after an action or action sequence and a learn function Q distributes the feedback among the evaluations.

# Reinforcement Learning (II)

The interesting part of reinforcement learning (often also called Q-learning) is how the learn function Q is defined. There are many possibilities and an important point is especially how the distribution of feedback is done after action sequences.

There are obvious similarities to learning in neural networks.

The basic agent architecture resembles Markov processes and their theory is used for proving properties of Q-functions.

From time to time random decisions have to be made to try out new situation action combinations
☞ exploration

# The DFG Algorithm - Scenario (I)

A set of organizations competes for furthering a given task. The general procedure is that for each occurring situation each organization is allowed to bid its next solution step and only the solution step of the best organization will be executed, thus generating the next situation.

An organization itself consists of compatible agents and smaller organizations. In the following, we call these organizations and agents units.

The units of a winning organization perform the actions that their decision functions suggest for the current situation.

# The DFG Algorithm - Scenario (II)

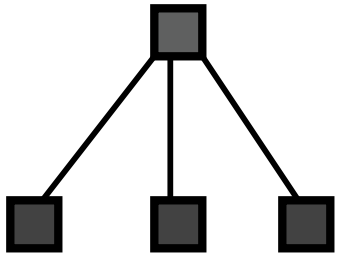
This is the reason why the units have to be compatible, i.e. no action of one unit can prevent the action of another unit.

In each organization there is one agent that is acting as leader and that computes the bids of the organization. It also receives the rewards (feedback) for the organization. It represents the whole organization.

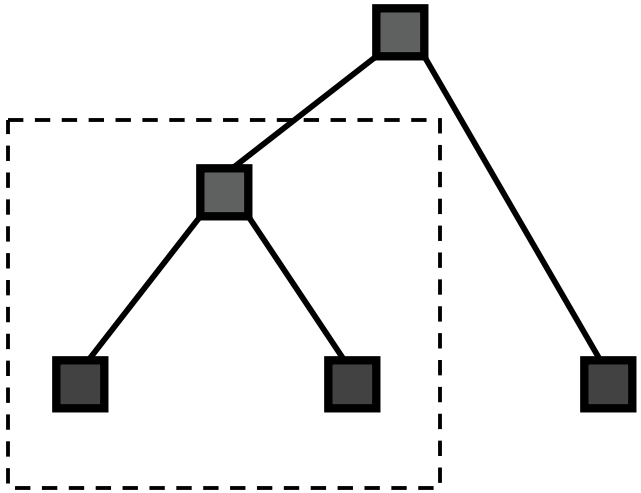
We want organizations to be dependent on situations!

# The DFG Algorithm - Examples for organizations

flat:



hierarchical:



# The DFG Algorithm - Rationale

Obviously, for each situation we want to find the organization whose units perform all possible actions that can be performed in parallel and that also are sensible, i.e. they should further the problem solution process.

The DFG algorithm tries to learn these organizations.

# The DFG Algorithm - The basic cycle

The DFG algorithm learns by extending, dissolving and forming of organizations.

Basic cycle:

1. Competition:

   Evaluation and selection of actions
2. Modification of evaluations:

   former and active organizations get rewarded
3. Development of organizations:

   Dissolving and forming of organizations

# Competition

$S_j$: actual situation

$U_i$: organization that could act in actual situation

$B_i^j = (a + b) \times E_i^j$: bid of $U_i$ for $S_j$, where

- a: learn factor
- b: random factor
- $E_i^j$: evaluation of the combined actions of $U_i$ for $S_j$ so far

# Modification of evaluations

Let $U_i$ be the organization winning in situation $S_j$ and $U_k$ the winning organization that led from situation $S_l$ to $S_j$

Modify the evaluations as follows:

$$E_i^j = E_i^j - a \times E_i^j + R^{extern}$$

$$E_k^l = E_k^l + a \times E_i^j$$

Where $R^{extern}$ is the extern feedback provided by the environment.

☞ this stabilizes successful action sequences and destabilizes unsuccessful sequences

# Development of organizations (I)

- After starting the system and as long as the evaluation of a unit is increasing, there is no need to look for alternative organizations, i.e. no extensions, no defects.
- An interest in alternative organizations starts, when the evaluation of a unit decreases or stagnates. In order to find this out, the leader (or the agent itself) computes a moving mean value of the last n modifications of the evaluation of the unit.

# Development of organizations (II)

- Organizations interested in alternatives form a new (combined) organization, if the modification mean value gets smaller than the evaluation before n+1 modifications (multiplied by a so-called formation factor).
  First the unit with the highest evaluation selects one cooperation partner, namely the compatible unit with the highest evaluation, then among the remaining ones this is repeated until all units found a new partner or there are no compatible units left anymore.

# Development of organizations (III)

- An organization is dissolved by its leader, if the mean value of its evaluation falls below its initial evaluation (from when it was formed) multiplied by a so-called dissolution factor.
- Whenever a unit has to bid the first time for its situation, it uses a predefined value $E^{init}$

# Characterization of the DFG algorithm

Each unit permanently does

- online learning
- with a teacher who specifies the quality of its behavior.

The learning is achieved by making experiences.

# Discussion

+ Good solution to problem scenario
+ Rather fine tuning of organizations to situations possible
- Only sensible for a small Sit and a small Mact
- In order to allow for learning, the same situations have to occur very often
- Big administrative overhead in agents