Faculty of Mathematics and Information Science
Warsaw University of Technology

# Scalable Approaches to Bi-level and Dynamic Optimization Problems

**Jacek Mańdziuk**

September 9, 2019

# Main Research Interests

- Games – all aspects
  - Human-like Game Playing (problem solving)
  - Multigame Playing
- Dynamic Optimiztion Problems
- Bi-level Optimizaton Problems

# **Human-Like** Problem Solving & **Human-Machine** Collaborative Problem Solving

- **Human-Like** Problem Solving
  - Intuition-based approaches (Knowledge patterns)
  - Focusing on goals /plans rather than particular actions
  - Multitasking
  - Highly selective search (efficient action preselection)
  - Search-Free or Shallow-Search based methods
  - Methods that rely on knowledge transfer (Transfer Learning)


- **Human-Machine** cooperation in problem solving
  - On-line steering of the algorithm
  - Human-Machine loop (trust building)
- **Human-Machine** co-learning
  - Human learning from collaborative problem solving

# Human-Machine Collaborative Problem Solving

- MCTS/UCT – simulation based approach
- Human player cooperates with UCT and has a budget of simulation batches

•Before each batch they can interfere with the algorithm:

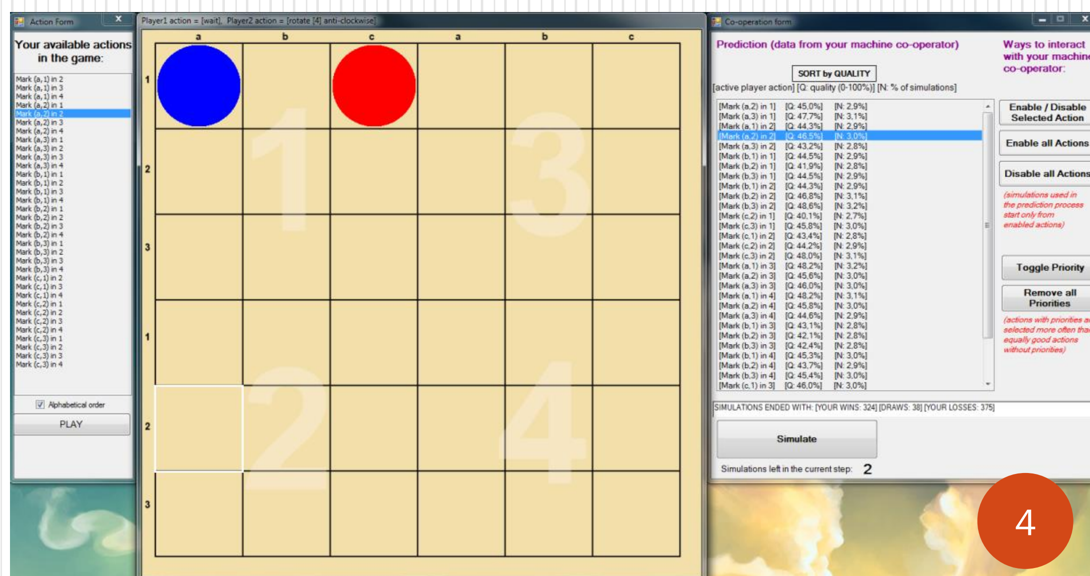
•**Enable/Disable actions** which simulations can start from (*narrowing the search*)

•**Toggling priorities of actions** making them simulated more often than the UCT formula says

•Humans **observe statistics** and make choices based on these statistics. In particular, they decide about the **move** to be played.

- **Human learns from collaborative problem solving**

- Creating a hybrid super-player
- Solving problems more effectively
- Improving the MCTS/UCT algorithm from the human input



# Security Games (SG)
# Stackelberg Equilibrium (Leader/Follower)

- A type of Pursuer-Evader game in a certain environment (usually represented as a graph)
- Two players: **Defender /Leader** (secures targets) and **Attacker /Follower** (attacks them)
- There are some targets to be defended/captured → **payoffs**
- **Imperfect information** (players reveal partial information about their strategies/behavior)
- The goal is to find **optimal Defender's strategy**
- **Non-zero-sum** game
- Both sides play a **mixed strategy**
- **BI-LEVEL: Follower knows Leader's strategy**
- **Follower is fully rational** and plays strategy that optimizes his reward
- **Follower breaks ties in his strategies in favor of leader's reward**

- Can be computed by solving continuous-discrete problem: **MILP**
- **Problems with scalability** of MILP solution

$for\ i \in \{D, A\}$

$\Sigma_i$ - a set of all pure strategies

$\Pi_i$ - a set of all mixed strategies, i.e. probability distribution over $\Sigma_i$

$$SE = (\pi_D, \pi_A) \in \Pi_D x \Pi_A$$

$$BR(\pi_D) = \arg\max_{\pi_A \in \Pi_A} U^A(\pi_D, \pi_A)$$

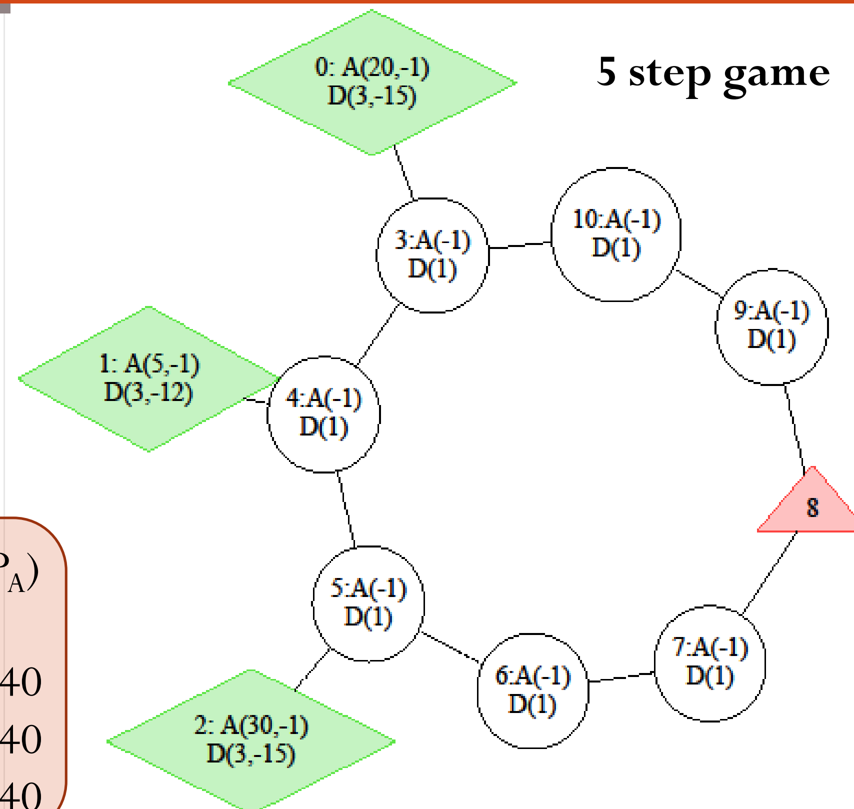
$$\arg\max_{\pi_D \in \Pi_D} U^D(\pi_D, BR(\pi_D))$$

# Security Games (SG)
# Stackelberg Equilibrium

Starting points: 0 & 8; Assets: 0, 1, 2

Rewards: non-targets and targets

Attacker can threaten either 0&1 or 1&2 ➔ Defender must follow *mixed strategy*




GAME EQUILIBRIUM:

| Prob. | Sequence | $E(P_D)$ |
|---|---|---|
| 0.4 | 0, 0, 0, 0, 0 | -15.00 |
| 0.6 | 3, 4, 5, 2, 2 | 3.00 |

| Prob. | Sequence | $E(P_A)$ |
|---|---|---|
| 1/3 | 7, 7, 6, 5, 2 | 11.40 |
| 1/3 | 7, 6, 6, 5, 2 | 11.40 |
| 1/3 | 8, 7, 6, 5, 2 | 11.40 |

**-4.20**

**-7.98**

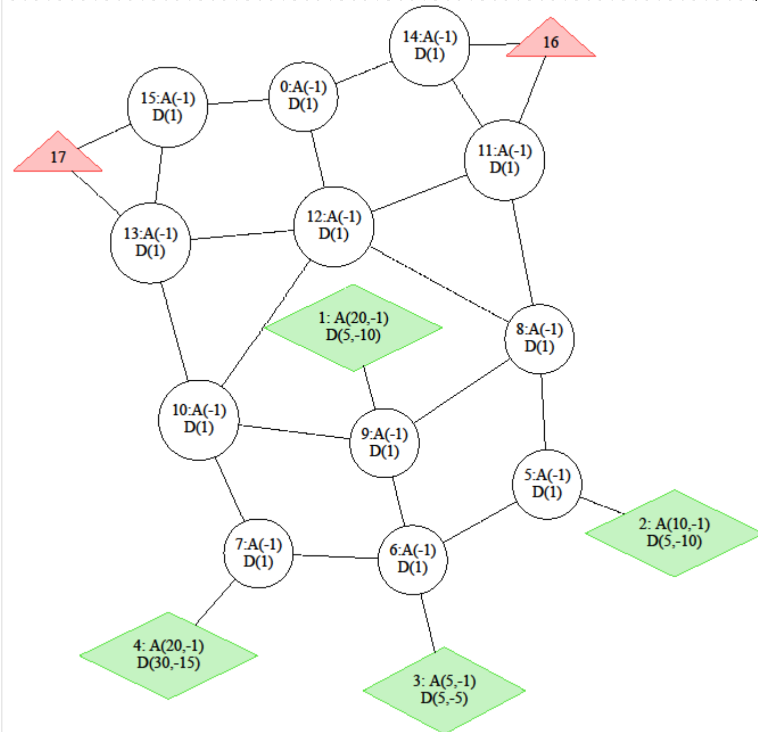
| Prob. | Sequence | $E(P_D)$ |
|---|---|---|
| 0.39 | 0, 0, 0, 0, 0 | -15.00 |
| 0.61 | 3, 4, 5, 2, 2 | 3.00 |

| Prob. | Sequence | $E(P_A)$ |
|---|---|---|
| 1/3 | 9, 9, 10, 3, 0 | 11.81 |
| 1/3 | 9, 10, 10, 3, 0 | 11.81 |
| 1/3 | 8, 9, 10, 3, 0 | 11.81 |

# Mixed-UCT method
# (Mixed strategy algorithm involving UCT)

**Idea**: iterative strategy improvement → equilibrium point

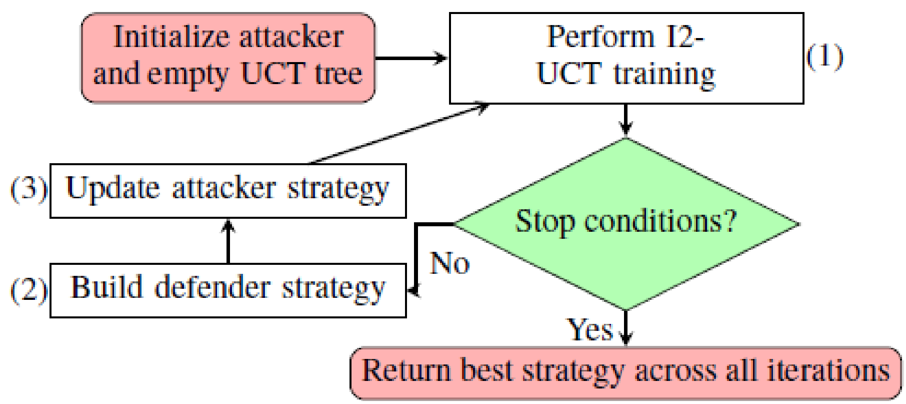
- 1. For the current Defender strategy calculate optimal Attacker strategy
- 2. **Perform UCT training against the currently calculated Attacker**
- 3. *** Store partial knowledge *** about current UCT solution
- 4. Calculate new Defender strategy based on collected knowledge
- 5. If not STOP then GOTO 1

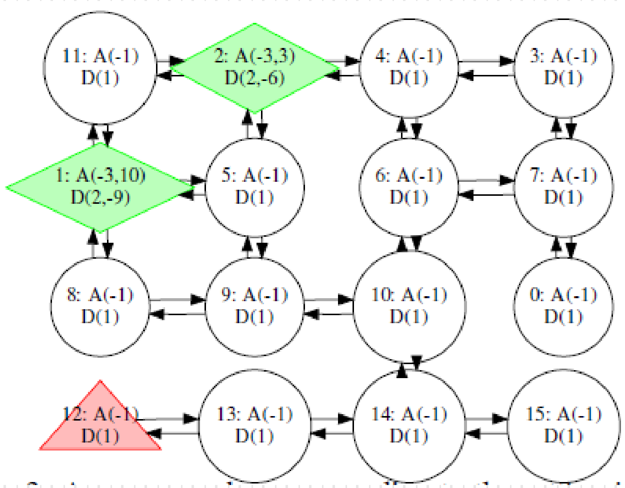


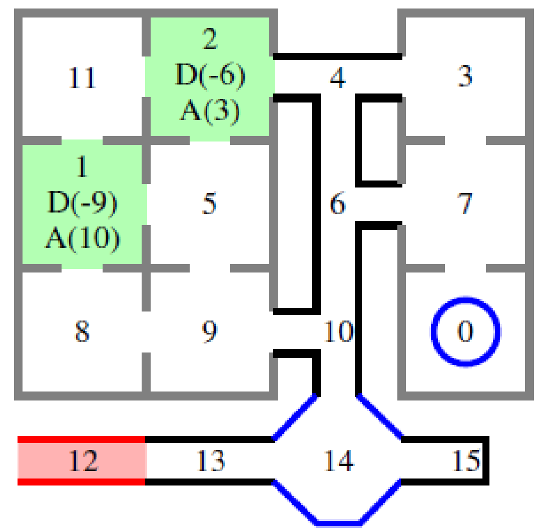

J. Karwowski, J. Mańdziuk, (2016), *Mixed strategy extraction from UCT tree in Security Games*, **22nd European Conference on Artificial Intelligence (ECAI'2016)**, The Hague, The Netherlands, 1746-1747

J. Karwowski, J. Mańdziuk, (2019), *A Monte Carlo Tree Search approach to finding efficient patrolling schemes on graphs*, **European Journal of Operational Research**, vol. 277, 255-268, Elsevier

# O2UCT - Improved UCT-based approach




**J. Karwowski, J. Mańdziuk, (2019),** *Stackelberg Equilibrium Approximation in General-Sum Extensive-Form Games with Double-Oracle Sampling*, ***International Conference on Autonomous Agents and Multiagent Systems (AAMAS-19)***, 2045-2047

# UCB1 (k-Armed Bandit)

- UCB = Upper Confidence Bounds (UCT = UCB applied to Trees)
- k-Armed Bandit Problem or k (one-arm) Bandits Problem
- Distributions of pay-offs $\{X_{j,t}\}_{t=1,2,\ldots}$ **$j=1,2,\ldots,k$** are fixed, but unknown
- How to maximize the total (long-term) reward?
- Exploration vs. exploitation balance

- First, play each arm once
- Then, use the following rule:

$$A^* = \arg\max_{i=1,\ldots,k} \left\{ \overline{X}_i + C\sqrt{\frac{\ln n}{n_i}} \right\},\ \ C = \sqrt{2}$$

# UCT – multiple simulations

Perform multiple simulations according to the following pattern:

Choose an action not yet selected (if exists)

If all had already been tried select action $a^*$

$$a^* = argmax_{a \in A(s)} \left\{ Q(s,a) + C\sqrt{\frac{\ln N(s)}{N(s,a)}} \right\}$$

*Q(s,a)* – the average result for a pair (*state*, *action*)

*N(s)* – the number of visits in state *s*

*N(s,a)* – the number of times action *a* was selected in state *s*

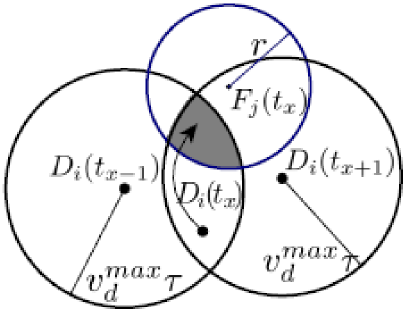
**Once the simulation is completed propagate the result back in the tree**

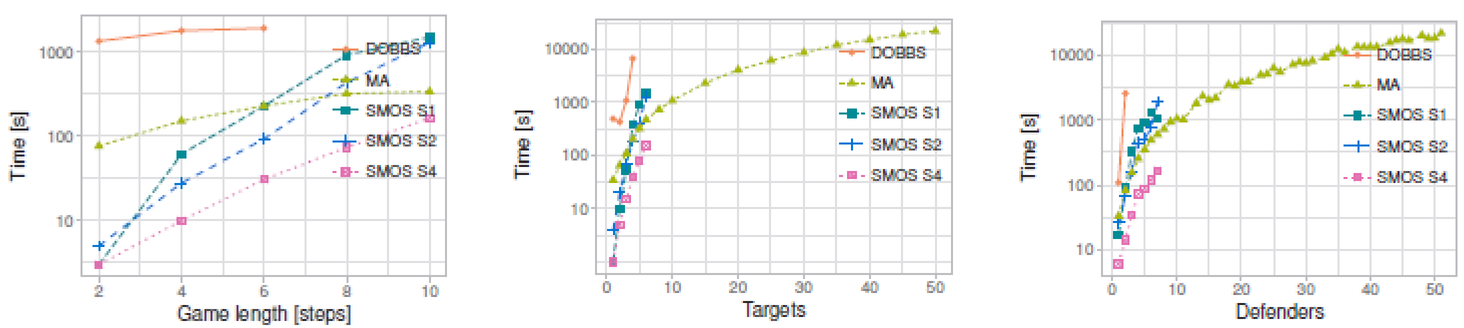
The saliency of C parameter

# SG with Moving Targets Evolutionary Approach

- $m \in \{2, 4, 6, 8, 10\}$
- $n_f \in \{1, 2, 3, 4, 5, 6, 8, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$
- $n_d \in \{max(1, n_f - 2), \ldots, n_f + 1\}$

- Real life transportation situations.
- Tourist harbour (ferry boats) in the Mediterranean Sea
- Number of ferries (targets) nf, patrolling boats nd
- Fixed ferry schedules on straingt routes
- Each chromosome encodes Defender's mixed strategy



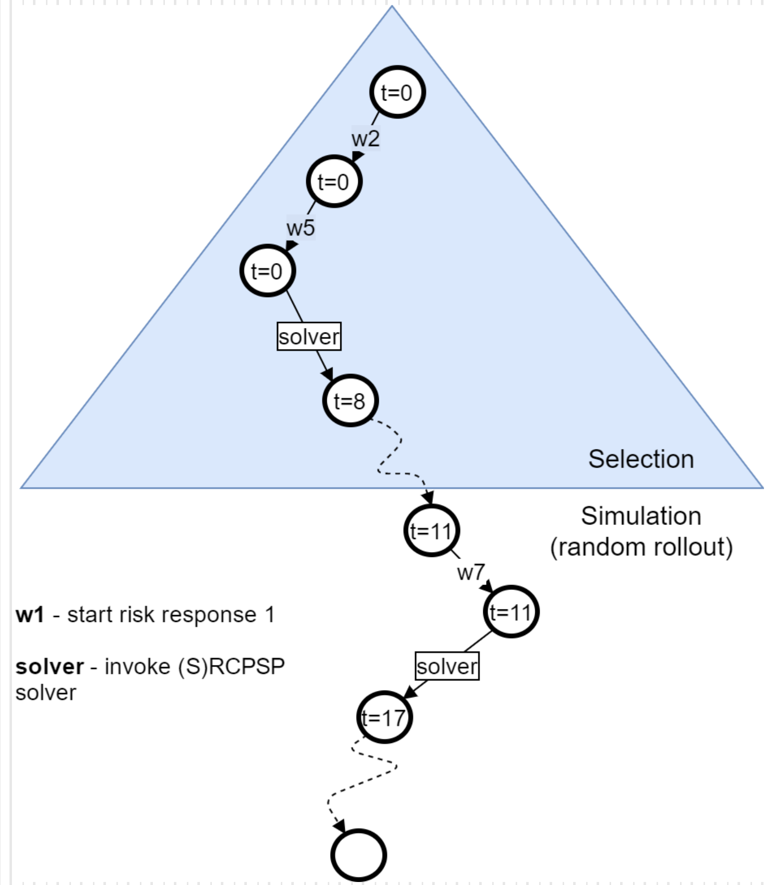

**J. Karwowski, J. Mańdziuk, A. Żychowski, F. Grajek, B. An (2019),** *A Memetic Approach for Sequential Security Games on a Plane with Moving Targets,* ***The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)***, 970-977

# Dynamic Optimization Problems

- Problems whose structure and / or parameterization change in time

- Risk-Aware Project Scheduling
- Dynamic Transportation Problems

- Monte Carlo Tree Search / UCT
- Evolutionary Algorithms
- Memetic Algorithms




**K. Walędzik, J. Mańdziuk, (2018)**, *Applying Hybrid Monte Carlo Tree Search Methods to Risk-Aware Project Scheduling Problem*, **Information Sciences**, vol. 460-461, 450-468, Elsevier

# Vehicle Routing Problem with Dynamic Requests Memetic Algorithm (EA + local optimization)

- *n* customers $v_1, v_2, \ldots, v_n$
- for each customer: demand, unload time (time required to unload cargo at customer's $v_i$), **arrival time $t_{vi}$** (time of arrival of the order from customer $v_i$)
- defined distance between each pair of customers
- fleet of m homogenous vehicles, each with identical capacity c
- speed of each vehicle is defined as one distance unit per one time unit
- depot with opening time $t_o$ and closing time $t_c$ ($0 <= t_o < t_c$)

**Goal:**

minimize the total routes' length of all vehicles according to the following constraints:

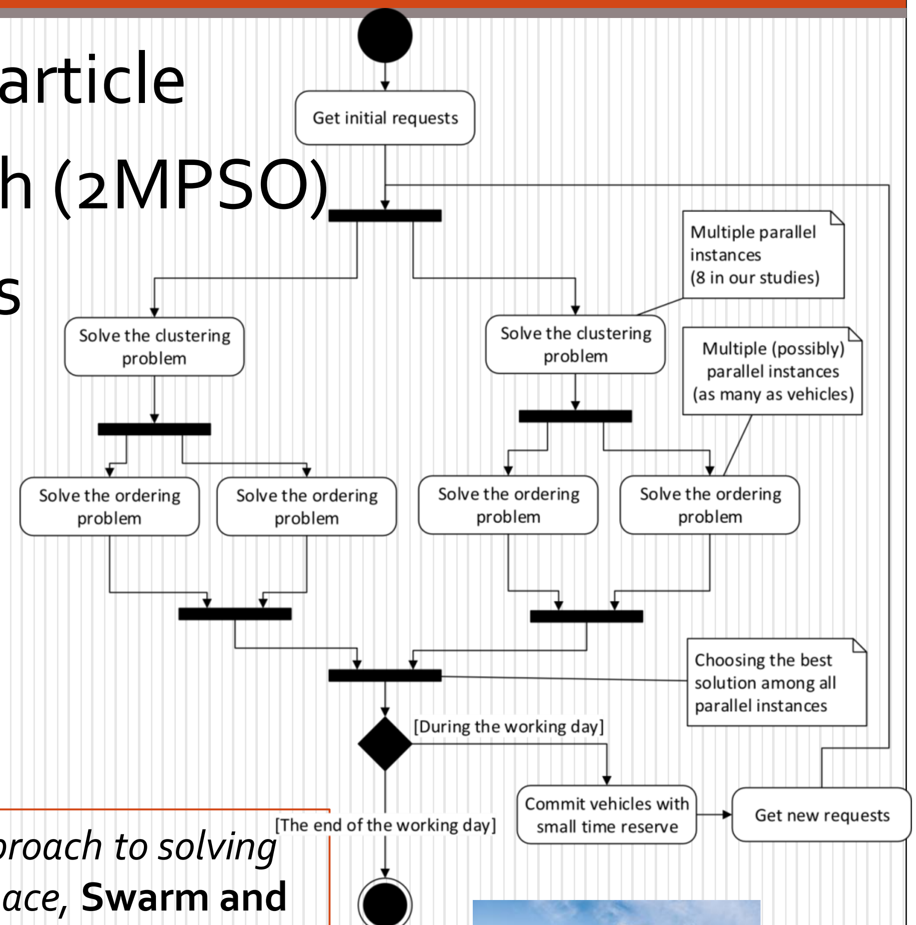
- each vehicle has to start from a depot after time $t_o$ and end its route in a depot before time $t_c$
- every customer has to be served exactly once and by one vehicle
- time of a vehicle arrival to customer $v_i$ has to be greater than $t_{vi}$ for all $i$
- the sum of customers' demands assigned to each vehicle must not exceed vehicle's capacity $c$



**J. Mańdziuk, A. Żychowski, (2016)**, *A Memetic Approach to Vehicle Routing Problem with Dynamic Requests*, **Applied Soft Computing**, vol. 48, 522-534, Elsevier.

# Vehicle Routing Problem with Dynamic Requests
# Particle Swarm Optimization with continuous coding

- Two-phase Multiple-Swarm Particle
- Swarm Optimization approach (2MPSO)
- Phase 1: clustering of requests
- Phase 2: routes optimization
- Continuous PSO coding
- Christofides / Fisher / Taillard dynamical benchmarks





**M. Okulewicz, J. Mańdziuk, (2019)**, *A metaheuristic approach to solving Dynamic Vehicle Routing Problem in continuous search space*, **Swarm and Evolutionary Computation**, vol. 48, 44-61, Elsevier

**M. Okulewicz, J. Mańdziuk, (2017)**, *The impact of particular components of the PSO-based algorithm solving the Dynamic Vehicle Routing Problem*, **Applied Soft Computing**, vol. 58, 586-604, Elsevier.


# Dynamic Vehicle Routing Problem with Traffic Jams

**Traffic is characterized by 3 distributions:**

Probability of encounter **p ∈ {0.02, 0.05, 0.15}** //for each edge

Length in steps **TTL := uniform<int>(2,5)**

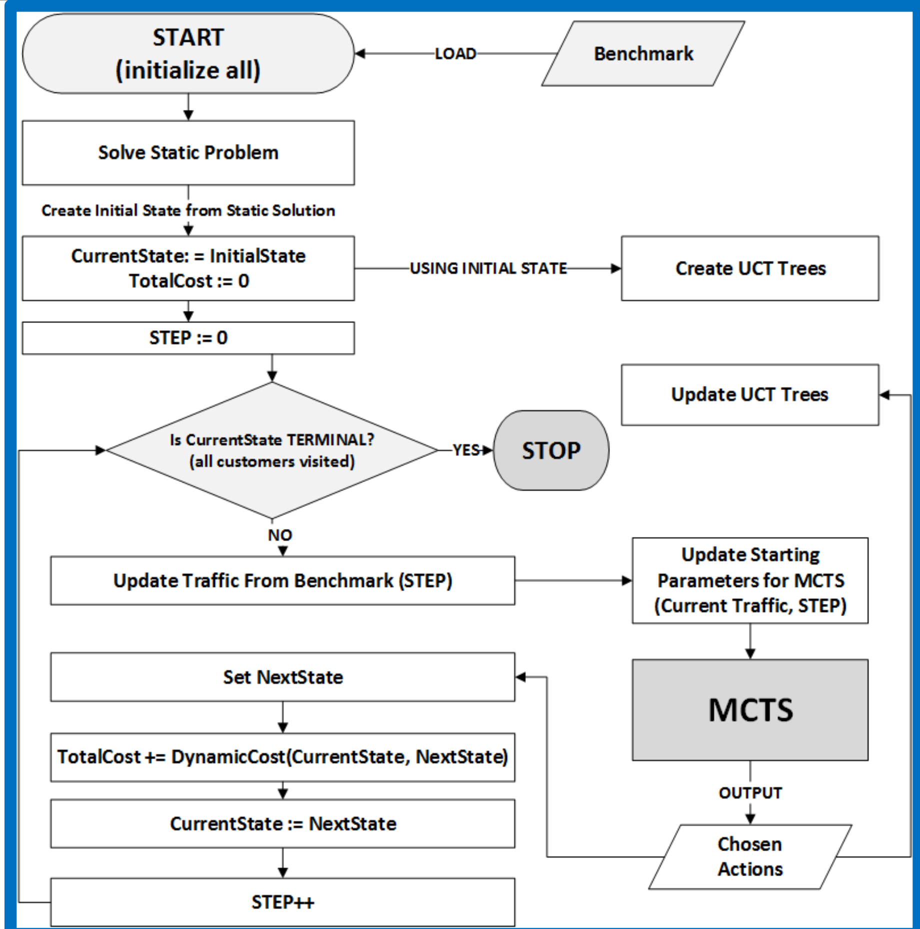
Intensity **I := uniform<int>(10,20)**

➔**The cost of an edge increases**

**Application of MCTS/UCT – UCT forest**

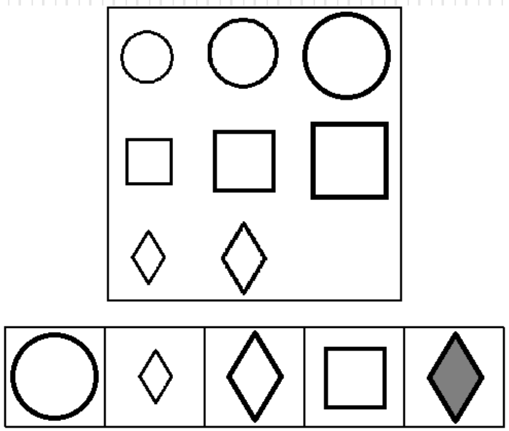
**One UCT tree per vehicle / Synchronization**

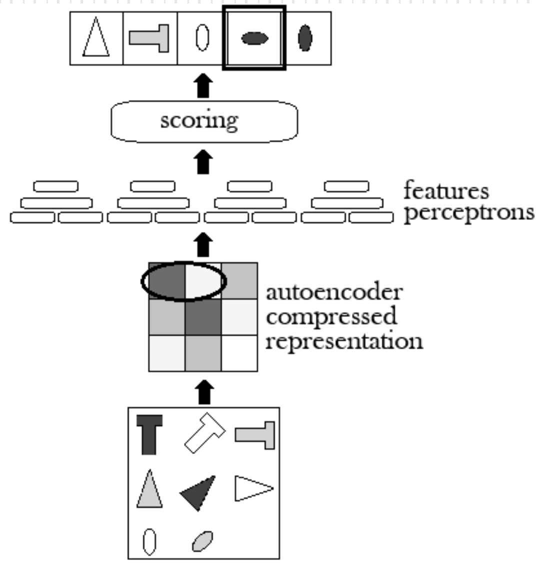
**UCT Actions:** local route modifications due to TJ: (swapping the two next clients in a route, move client from a jammed route to unjammed one, reverse the order of all clients in a route, etc.)

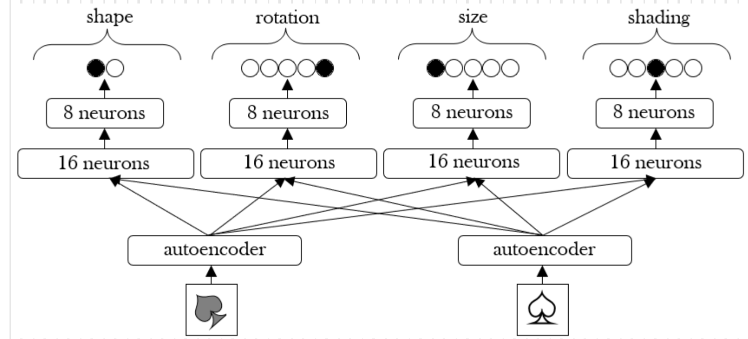


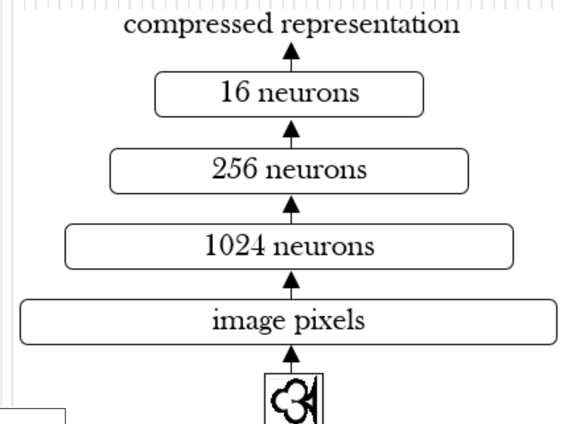

**J. Mańdziuk, M.Świechowski, (2017)**, *UCT in Capacitated Vehicle Routing Problem with Traffic Jams*, **Information Sciences**, vol. 406-407, 42-56, Elsevier

# Deep Autoencoders + MLPs ➔ Transfer Learning #1

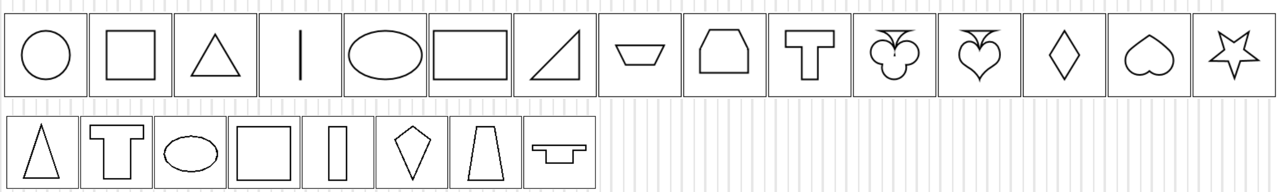




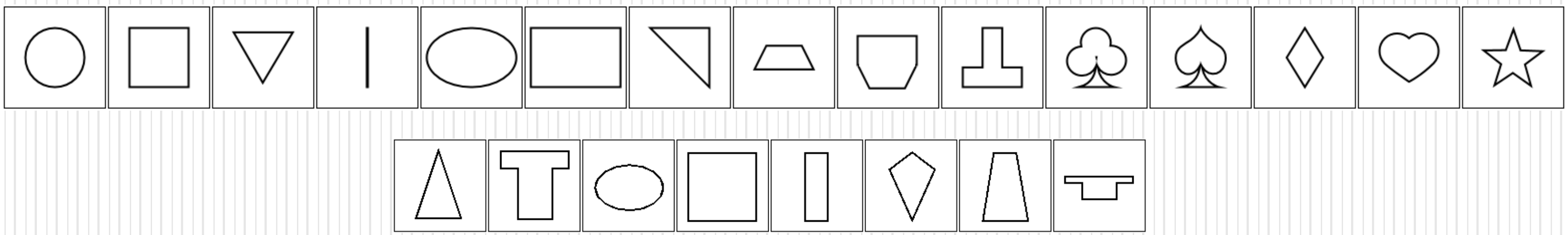

1. Different types of figures used in training and testing
2. Different types of IQ problems in training and testing

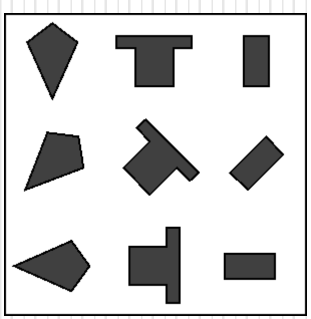


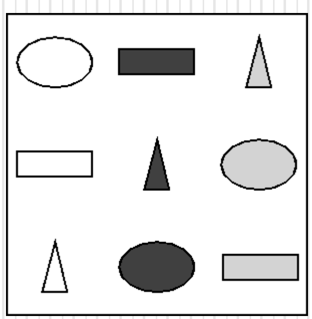
**J. Mańdziuk, A. Żychowski (*submitted*, 2019)**, DeepIQ: *A Human-Inspired AI System for Solving IQ Test Problems*

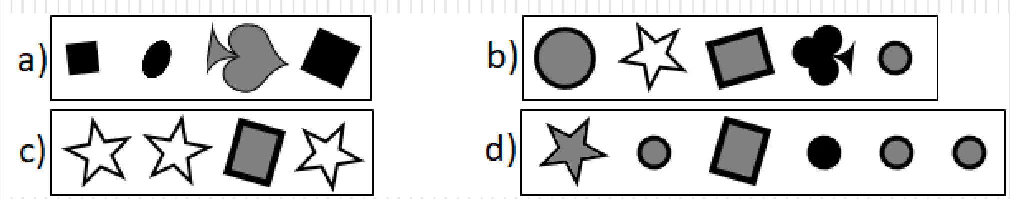
# Deep Autoencoders + MLPs ➔ Transfer Learning #2

1. Different types of figures used in training and testing
2. Different types of IQ problems in training and testing










No additional training or tuning of the system

# My Research Team

Dr Michał Okulewicz
(Assistant Professor)

Jan Karwowski, MSc
(Assistant Professor)

Dominik Lewy,
(PhD Candidate)

Stanisław Kaźmierczak,
(PhD Candidate)

Mateusz Zaborski,
(PhD Candidate)

Maciej Żelaszczyk,
(PhD Candidate)

Adam Żychowski,
(PhD Candidate)

# SUMMARY

- **GAMES / MULTISTEP DECISION MAKING PROBLEMS**
  - Human-like problem solving
  - Multigame Playing
- **BI-LEVEL OPTIMIZATION (Scalable solutions)**
  - MCTS/UCT simulation approaches
- **DYNAMIC OPTIMIZATION PROBLEMS (Scalable solutions)**
  - MCTS/UCT simulation approaches
  - Evolutionary Algorithms
  - Memetic Algoritms

**Thank You!**

**Questions?**