



# Interactive machine learning: experimental evidence for the human in the algorithmic loop

## A case study on Ant Colony Optimization

Andreas Holzinger<sup>1</sup> · Markus Plass<sup>1</sup> · Michael Kickmeier-Rust<sup>2</sup> · Katharina Holzinger<sup>1</sup> · Gloria Cerasela Crişan<sup>3</sup> · Camelia-M. Pinteaa<sup>4</sup> · Vasile Palade<sup>5</sup>

Published online: 7 December 2018  
© The Author(s) 2019

### Abstract

Recent advances in automatic machine learning (aML) allow solving problems without any human intervention. However, sometimes a human-in-the-loop can be beneficial in solving computationally hard problems. In this paper we provide new experimental insights on how we can improve computational intelligence by complementing it with human intelligence in an interactive machine learning approach (iML). For this purpose, we used the Ant Colony Optimization (ACO) framework, because this fosters multi-agent approaches with human agents in the loop. We propose unification between the human intelligence and interaction skills and the computational power of an artificial system. The ACO framework is used on a case study solving the Traveling Salesman Problem, because of its many practical implications, e.g. in the medical domain. We used ACO due to the fact that it is one of the best algorithms used in many applied intelligence problems. For the evaluation we used gamification, i.e. we implemented a snake-like game called Traveling Snakesman with the MAX–MIN Ant System (MMAS) in the background. We extended the MMAS–Algorithm in a way, that the human can directly interact and influence the ants. This is done by “traveling” with the snake across the graph. Each time the human travels over an ant, the current pheromone value of the edge is multiplied by 5. This manipulation has an impact on the ant’s behavior (the probability that this edge is taken by the ant increases). The results show that the humans performing one tour through the graphs have a significant impact on the shortest path found by the MMAS. Consequently, our experiment demonstrates that in our case human intelligence can positively influence machine intelligence. To the best of our knowledge this is the first study of this kind.

**Keywords** Interactive machine learning · Human-in-the-loop · Combinatorial optimization · Ant Colony Optimization

## 1 Introduction

### 1.1 Automatic machine learning

One of the fundamental objectives of Artificial Intelligence (AI) in general and of Machine Learning (ML) in particular is to find methods and develop algorithms and tools that automatically learn from data, and based on them, provide results *without human interaction*. Such algorithms can be called automatic ML (aML) - where automatic means autonomous in the sense of classical AI [1]. A close concept

is *automated* ML (AutoML) [2], which focuses on end-to-end automation of ML and helps, for example, to solve the problem of automatically (without human interaction) producing test set predictions for a new data set.

Automatic approaches are present in the daily practice of human society, supporting and enhancing our quality of life. A good example is the breakthrough achieved with deep learning [3] on the task of phonetic classification for automatic speech recognition. Actually, speech recognition was the first commercially successful application of deep convolutional neural networks [4]. Today, autonomous software is able to conduct conversations with clients in call centers; Siri, Alexa and Cortana make suggestions to smartphone users. A further example is automatic game playing without human intervention [5]. Mastering the game of Go has a long tradition and is a good benchmark for progress in automatic approaches, because Go is hard for computers [6].

---

✉ Andreas Holzinger  
andreas.holzinger@medunigraz.at

Extended author information available on the last page of the article.

Even in the medical domain, automatic approaches recently demonstrated impressive results: automatic image classification algorithms are on par with human experts or even outperforms them [7]; automatic detection of pulmonary nodules in tomography scans detected the tumoral formations missed by the same human experts who provided the test data [8]; neural networks outperformed a traditional segmentation methods [9], consequently, automatic deep learning approaches became quickly a method of choice for medical image analysis [10].

Undoubtedly, automatic approaches are well motivated for theoretical, practical and commercial reasons. However, in many real-world applications a human-in-the-loop can be beneficial. This paper explores some catalytic and synergistic effects of the integration of human expertise not only into the data processing pipeline as in standard supervised learning, but directly into the algorithm [11].

## 1.2 Disadvantages of automatic approaches

Unfortunately, automatic approaches and particularly deep learning approaches have also several disadvantages. Automatic approaches are intensively resource-consuming, require much engineering effort, need large amounts of training data (“big data”), but most of all they are often considered as black-box approaches. Although this is not quite true, they are kind of opaque, meaning that they are complex even if we understand the underlying mathematical principles. For a recent discussion of black-box approaches in AI refer to [12]. International concerns are raised on ethical, legal and moral aspects of developments of AI in the last years; one example of such international effort is the Declaration of Montreal.<sup>1</sup>

Black-box approaches have - at least when applied to a life-critical domain, such as medicine - one essential disadvantage: they are lacking transparency, i.e. they often do not expose the decisional process. This is due to the fact that such models have no explicit declarative knowledge representation, hence they have difficulty in generating the required explanatory structures – which considerably limits the achievement of their full potential [13].

To our experience this does not foster trust and acceptance among humans. A good example are medical professionals, e.g. physicians, which are particularly reluctant to human-machine interactions and prefer personal deductions based on human-to-human discussions and on personal expertise. Most of all, legal and privacy aspects make black-box approaches difficult [14, 15].

Implementing these new legal regulations requires supplementary costs for software companies targeting the

European market, especially start-ups and small companies. Consequently, two big issues come up: firstly, to enable - on demand - to re-trace *how* a machine decision has been reached; secondly, to control the impact of data removal on the effectiveness of automatic approaches. The software producers will need to consider privacy-aware data communications methods and also secured models for open data sets [16], which require new approaches on the effectiveness of ML implementations previously neglected [17].

*Data representation* is a key factor in the effectiveness of ML implementations. Contemporary approaches do not (automatically) *extract the discriminative knowledge* from bulk data. Bengio et al. [18] claims that only really intelligent algorithms that understand the *context*, and which have the ability to retain significant features may achieve (in the future) the discriminative characteristics. Genuinely human questions including interest and relevance are inherently difficult for AI/ML, as long as by now the true intelligence is not *automatically* achieved. Of course, this is the grand goal of AI research, as outlined in the first paragraph; however, it is assumed that reaching these goals still need quite some time [19].

## 1.3 Motivation for a human-in-the-loop

Current ML algorithms work asynchronously in connection with a human expert who is expected to help in data preprocessing and data interpretation - either before or after the learning algorithm. The human expert is supposed to be aware of the problem’s context and to correctly evaluate specific datasets. This approach inherently connects ML to cognitive sciences, AI to human intelligence [20].

Interactive Machine Learning (iML) often refers to *any* form of user-facing machine learning approaches [21]. Several authors consider that a human intervention is compulsory, but in our opinion these type of interventions are just forms of classical supervised ML approaches [22], and an entirely distinct approach to ML is to insert the human into physical feedback loops [23].

By putting the human in-the-loop (a human kernel, as defined in [20]), iML looks for “*algorithms which interact with agents and can optimize their learning behaviour through this interaction – where the agents can be humans* [24]”. This perspective basically integrates the human into the *algorithmic* loop. The goal is to opportunistically and repeatedly use human knowledge and skills in order to improve the quality of automatic approaches. The iML-approaches can therefore be effective on problems with scarce and/or complex data sets, when aML methods become inefficient. Moreover, iML enables features as re-traceability and explainable-AI, important characteristics in the medical domain [25].

<sup>1</sup><https://www.montrealdeclaration-responsibleai.com>

In this paper, we are further developing previous iML research. In the empiric previous works [26, 27], we provided practical examples of iML approaches to the Traveling Salesman Problem (TSP).

This difficult optimization problem models many real-world situations in the medical domain, e.g. in protein folding processes described as a free energy minimization problem [28].

As TSP is proven to be NP-complete [29], its high-dimension instances are unlikely to be solved with exact methods. Consequently, many heuristic and approximate TSP solvers have been described for finding close-enough solutions [30] and inspired psychological research, which found (a) the complexity of solutions to visually presented TSPs depends on the number of points on the convex hull; and (b) the perception of optimal structure is an innate tendency of the visual system [31].

Widely used meta-heuristic algorithms include: Tabu Search, genetic algorithms, simulated annealing and Ant Colony Optimization, etc. (a brief overview is presented in [32]).

Several features that enable the success of the human-in-the-loop in the medical domain are presented in [11]. More general, this paradigm can be broadly rewarding in multiple situations of different domains, where human creativity, mental representations and skills are able to heuristically focus on promising solution space regions.

To our knowledge, the efficiency and the effectiveness of the iML approaches have not been studied in-depth so far. The underlying mechanism of *how* the human-computer system may improve ML approaches has to be described by cognitive science [33]. For example, physicians can give correct diagnoses, but they can not explain their deduction steps. In such cases, iML may include such “instinctive” knowledge and learning skills [34].

### 1.4 Learning from very few examples

This work was motivated by the observation that humans are sometimes surprisingly good in learning from very few examples. Josh Tenenbaum from MIT asked “*How do humans get so much from so little (data)*” and even a two year old child can discriminate a cat from a dog without learning millions of training examples. Scientific examples include the aML approaches based on Gaussian processes (e.g., kernel machines [35]), which are weak on function extrapolation problems, although these problems are quite simple for humans [36].

The paper is organized as follows: In Section 2 we provide some background and related work, in Section 3 we give some background on Ant Colony Optimization, in Section 4 we introduce the new concepts based on

human interactions with artificial ants, and we conclude with several observations and an outlook to future work.

## 2 Background and related work

### 2.1 Human vs. computer in problem solving

Extrapolation problems, as mentioned before, are generally considered challenging for ML approaches [37]. This is mostly due to the fact that rarely one has exact function values without any errors or noise, which is therefore posing difficulties when solving real-world problems [38].

Interestingly, such problems are extremely easy for humans, which have been nicely demonstrated in an experiment by [20]: a sequence of functions extracted from Gaussian Processes [39] with known kernels was presented to a group of 40 humans. They showed that humans have systematic expectations about smooth functions that deviate from the inductive biases inherent in the kernels that have been used in past models of function learning. A kernel function measures the similarity between two data objects.

Formally, a kernel function takes two data objects  $x_i$  and  $x_j \in \mathbb{R}^d$ , and produces a score  $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ .

Such a function can also be provided by a human to the machine learning algorithm, thus it is called: *human kernel*. Automatically, this is done by a Support Vector Machine (SVM), because under certain conditions a kernel can be represented by a dot product in a high-dimensional space [40]. One issue here is that a kernel measures the similarity between two data objects, however, it **cannot** explain *why* they are similar. Here a human-in-the-loop can be of help to find the underlying explanatory factors of why two objects are similar, because it requires context understanding in the target domain.

In the experiment described in [20], the human learners were asked to make extrapolations in sequence; they had the opportunity to use prior information on each case. At the end, they could repeat the first function; then they were questioned on deductions they made, in an attempt to understand the effect of inductive biases that exhibit difficulties for conventional Gaussian process (GP) kernels. The open question is still: how do humans do that. Even little children can learn surprisingly well from very few examples and grasp the meaning, because of their ability to generalize and apply the learned knowledge to new situations [41]. The astonishing good human ability for inductive reasoning and *concept generalization from very few examples* could derive from a prior combined with Bayesian inference [42].

Bayesian approaches provide us with a computational framework for solving inductive problems, however, much

remains open and we are still far away from being able to give answers on the question of how humans can gain out so much from so little data [43]. More background on these problems can be found for example in [44, 45].

As stated in the very beginning, a *true intelligent* ML algorithm must be able to automatically learn from data, extract knowledge and make decisions without human intervention. Therefore, ML was always inspired by *how* humans learn, how they extract knowledge and how they make decisions. Key insights from past research provided probabilistic modelling and neurally inspired algorithms (see e.g. [46]). The capacity of a model to automatically provide patterns and to extrapolate is influenced by a priori possible solutions and a priori likely solutions. Such a model should represent a large set of possible solutions with inductive biases, be able to extract complex structures even from scarce data.

Function learning is also important for solving tasks in everyday cognitive activities: nearly every task requires mental representations mapping inputs to outputs  $f : X \rightarrow Y$ . As the set of such mappings is infinite, inductive biases need to constrain plausible inferences. Theories on how humans learn such mappings when continuous variables are implied have focused on two alternatives: 1) humans are just estimating explicit functions, or 2) humans are performing associative learning supported by similarity principles. [47] developed a model that unifies both these assumptions.

## 2.2 Human abilities on optimization problems

The research at the boundary between Cognitive Science and Computational Sciences is challenging as, on one hand, improving automatic ML approaches will open the road of performance enhancements on a broad range of tasks which are difficult for humans to process (i.e. big data analytics or high-dimensional problems); on the other hand, these practical assignments may further the results in Computational Cognitive Science.

Undoubtedly, aML algorithms are useful in computationally demanding tasks: processing large data volumes with intricate connections, solving multi-dimensional problems with complicated restrictions, etc. However, aML algorithms proved to be less efficient when they approach problems where context information is missing, for example when solving  $\mathcal{NP}$ -hard problems. In general, aML is weak in *unstructured problem* solving: the computer lacks the human *creativity* and *curiosity*. The synergistic interaction between people and computers are highly investigated in games crowdsourcing. An example of successful AI product is Google's AlphaGo, which won the match against the world Go champion Lee Sedol [48]. Such results emphasize the potential human-computer cooperation has [49].

Understanding the human skills used in optimization is one of the current research concerns in Cognitive Science [50]. The relatively low complexity of the algorithms processing visual data could help solving difficult problems. This is a significant feature, as many practical tasks in medicine and health are exceptionally difficult to solve [51].

In [30] the solutions obtained by humans on the Traveling Salesman Problem (TSP) were compared to those provided by Nearest Neighbor, Largest Interior Angle and Convex Hull methods. The study showed that the humans constructed the solutions based on a **perceptual process**. This result was validated by two other experiments presented in [52]. The former test used the same dataset as in [30]. The latter test used TSP instances with uniformly distributed vertices. The TSP instances were individually displayed. The human used the mouse to build a solution. No solution revision was allowed. This investigations revealed that the humans mostly used the Nearest Neighbor method. When the Nearest Neighbor algorithm was hybridized with a method controlling the global shape of the solution, the fit of the simulation to human performance was quite close.

There are clear evidences that humans are exceptionally good in finding near optimal solutions to difficult problems; they can detect and exploit some structural properties of the instance in order to enhance solution parts. It is interesting that e.g. medical experts are not aware on how expensive it would be to computationally solve these problems [53, 54].

In [55], the solutions of 28 humans to 28 instances of the Euclidean TSP were assessed [56]. After each try on each TSP instance, the participants received the cost of their solution. Unlimited tries were allowed. The authors reported important enhancement of the solution's quality provided by each participant, and that the solutions differed in multiple ways from the random tours that follow the convex hull of the vertices and are not self-crossing. They also found that the humans more often kept the edges belonging to the optimal solution (the good edges) in their solution than the other edges (the bad edges), which is an indication for good structural exploitation. Too many trials impaired the participants' ability to correctly make use of good edges, showing a threshold which makes them to simply ran out of ideas.

As proof-of-concept, we have chosen the Traveling Salesman Problem (TSP), a problem with many applications in the health domain. The proposed human-computer system considered an enhanced Ant Colony Optimization (ACO) version which include one more special agent - the human [57]. This idea can be seen either as an inclusion of the human as a pre-processing module in the multi-agent artificial system, or as a transition to an "enhanced expert", where "enhanced" has the same meaning as in "enhanced reality".

### 2.3 Traveling Salesman Problem (TSP)

The *Traveling Salesman Problem* (TSP) is one of the most known and studied *Combinatorial Optimization Problems* (COPs). Problems connected to TSP were mentioned as early as the last eighteenth century [58]. During the past century, TSP has become a traditional example of difficult problems and also a common testing problem for new methodologies and algorithms in Optimization. It has now many variants, solving approaches and applications [59]. For example, it models in computational biology the construction of the evolutionary trees [60], in genetics - the DNA sequencing [61]; it proved to be useful in designing of healthcare tools [62], or in healthcare robotics [63].

The formal TSP definition, based on Graph Theory, is the following [59]:

**Definition 1.** On a complete weighted graph  $G = (V, E, w)$ , where  $V$  is the set with  $n$  vertices, let  $(w_{ij})_{1 \leq i, j \leq n}$  being the weight matrix  $W$  associated with  $E$ . The goal of TSP is to find a minimum weight Hamiltonian cycle (i.e. a minimum weight cycle that passes through any vertex once and only once).

From the complexity theory point of view, TSP is a NP-hard problem, meaning that a polynomial time algorithm for solving all its cases has not been found by now and it is unlikely that it exists [64].

As an integer linear program, TSP considers the vertices forming the set  $\{1, 2, \dots, n\}$  and searches for the values for the integer variables  $(x_{ij})_{1 \leq i, j \leq n}$  which are meant to describe a path:

$$x_{ij} = \begin{cases} 1 & \text{if the edge } (ij) \text{ is used} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

TSP objective (2) is subject to the constraints (3)–(5) [65].

$$\min \left( \sum_{i,j=1}^n x_{ij} w_{ij} \right). \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall 1 \leq j \leq n, \quad (3)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall 1 \leq i \leq n, \quad (4)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V, \quad 2 \leq |S| \leq n - 1. \quad (5)$$

The constraints (3) and (4) certify that each vertex has only two incident edges. The constraint (5) excludes the subtours (i.e. the cycles with less than  $n$  distinct vertices).

Several restrictive TSP variants have important transportation applications. The symmetric TSP considers that the weight matrix is symmetric. In the opposite case, TSP is

asymmetric. Metric TSP has the weights forming a metric that obeys the triangle inequality (in this case, the weights can be seen as distances). *Euclidean TSP* has the weights computed using the *Euclidean* norm.

TSP can be generalized in order to model more complex real-life situations. Time dependent TSPs use variable values stored in the weight matrix  $W$  [66]. The *Vehicle Routing Problem* (VRP) seeks for the least weighted set of routes for a set of vehicles that have to visit a set of customers [65].

The most used TSP benchmarks are <sup>2</sup> and <sup>3</sup>. They help in assessing the implementations of the new methodologies for solving TSP and its variants. New data features as geographic coordinates are used for integration with GIS technologies or allowing network analysis [67].

### 3 Ant Colony Optimization (ACO)

The *Ant Colony Optimization* (ACO) is a bio-inspired meta-heuristic approach for solving Optimization problems, modeling the ability of real ants to quickly find short paths from food to nest. ACO has been devised in the last years of the past century and today is intensively studied and applied [68]. The behavior of real ants can be used in artificial ant colonies for searching of close-enough solutions mainly to discrete optimization problems [69].

As one of the most successful swarm-based eusocial animals on our planet, ants are able to form complex social systems. Without central coordination and external guidance, the ant colony can find the shortest connection between two points based on indirect communication. A moving ant deposits on the ground a chemical substance, called pheromone. The following ants detect the pheromone and is more likely to follow it. Specific ant species exhibit more elaborated behavior [70].

The multi-agent system following the natural ants' behavior consists of artificial ants that concurrently and repeatedly construct Hamiltonian circuits in a complete graph. Initially, all the edges receive the same pheromone quantity. The next added edge to a partial solution is chosen based on its length and on its pheromone quantity. Shorter edges and also highly-traversed edges are more likely to be chosen.

After all the circuits are constructed, the pheromone on the traversed edges is reinforced. These steps are iterated until the stopping condition is met. The algorithm returns the best solution ever found (*best\_solution*). In this case, *DaemonActions* could include centralized modules (as guidance through promising zones of the solution space) or supplementary activities for some or all ants (for example,

<sup>2</sup>[iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95](http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95)

<sup>3</sup><http://www.math.uwaterloo.ca/tsp/data/index.html>

the solution construction is hybridized with local search procedures).

*ACO* is a metaheuristic based on the artificial ants paradigm. The *ACO* pseudocode contains generic methods, which allow approaching a broad set of combinatorial optimization problems [71]. Each problem defines a complete graph, where the artificial ants work based on specific rules for solution construction, pheromone depositing and other supplementary methods. For example, a supplementary activity is an exchange heuristics (2-Opt, 2.5-Opt or 3-Opt) done by each ant after its solution is constructed. In 2-Opt, two edges are replaced by other two that maintains the circuit but produces a smaller length [72]. The 2.5-Opt also includes a better relocation of three consecutive vertices [73]. The 3-Opt excludes three edges and reconnect the circuit in a better way [74]. The pheromone update rule can be changed in order to improve the local search, as in [75].

```

procedure ACO_Metaheuristic
  set parameters, initialize pheromone
  trails
  ScheduleActivities
    ConstructAntsSolutions
    UpdatePheromones
    DaemonActions % optional
  end-ScheduleActivities
  return best_solution
end-procedure

```

Several *ACO* methods monitor the overall colony activity and reacts when undesired situations manifest. For example, in [76], stagnation is avoided by decreasing the pheromones on best paths. Other methods of pheromone alteration is described in [77]. It is specifically designed for dynamic TSP with altered distances. The authors consider that the algorithm is aware when alteration moments appear and immediately reacts.

All these modules belong to the software solving package, so they are seen as *DaemonActions* activities. To our knowledge, this approach is the first attempt to introduce totally external guidance to a working, automated process. Our investigation wants to opportunistically use the good ability of humans to solve TSP instances, proven by broad researches on human trained or untrained TSP solving [50]. These studies showed that: the humans have the ability to better solve an instance on repeated trials, they perform better on visually described instances with less vertices, they rarely provide crossing tours, and instinctively use the convex hull property (the vertices on the convex hull are traversed in their order). A visual interface for a game designed for assessing the human abilities in solving small Euclidean TSP instances is in [55].

### 3.1 MAX-MIN Ant System (MMAS)

Probably one of the most studied *ACO* algorithms is the MAX-MIN Ant System [78]. There are four changes regarding the initial AS algorithm. First, it exploits the best tour of an ant, like the *ACO*. Second, it limits the excessive growth of the pheromones on good tours (which in some cases is suboptimal), by adding upper and lower pheromone limits  $\tau_{\min}$  and  $\tau_{\max}$ . Third, it initializes the pheromone amount of each edge to the upper pheromone limit  $\tau_{\max}$ , which increases the exploration of new tours at the start of the search. Finally, each time, if there is a stagnation in some way or no improvement of the best tour for a particular amount of time, it reinitializes the pheromone trails.

## 4 New concepts based on human manipulation of pheromones

Following the *aML* paradigm, *ACO* solvers are closed processes. The ants form a closed multi-agent system which as a whole constructs a set of solutions and delivers the best one. Following the *iML*-approach, here the human becomes an agent too, able to open the *ACO* process and to manipulate it by dynamically changing its parameters. As a result, *ACO* is enhanced with a new, special agent: the human.

As described in [26] and [27], the control in the procedure *ConstructAntsSolutions* is given to one more agent: the human. This special agent (the human) has the option to change the pheromones spread by the ants. For this, the human selects 2 nodes in the graph and doubles the pheromone-level of a specific edge.

The setup of the Experiment consists of to parts, the *aML*-Part, based on *MMAS*-Algorithm as described above, and the *iML*-Part.

The *aML*-Part is pre-configured with the following *MMAS*-Parameters:

- $\alpha = 1$
- $\beta = 2$
- $\rho = 0.02$
- $p_{\text{best}} = 0.05$

## 5 The experiment

As input for the *iML* part, the data is generated out of a “snake”-like game, called Traveling Snakesman see Fig. 1 and.<sup>4</sup> An apple represent an node of the graph. The path traveled to the next apple is the edge. Each new game

<sup>4</sup><https://hci-kdd.org/project/iml>

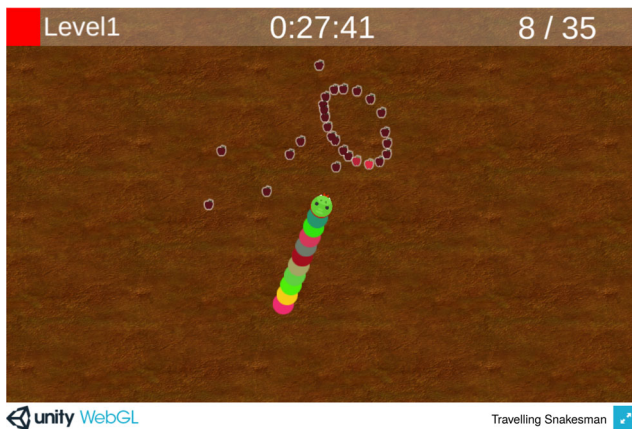


Fig. 1 Screenshot of the the traveling Snakesman game

is creating a new instance of the MMAS-Algorithm. The algorithm runs automatically 5 iterations, after this he is waiting for the user interaction. Each time the user eats an apple, the pheromone-level of the traveled edge is multiplied by 5. This can be seen as:

```
start the GAME
init MMAS
draw appels
run 5 iterations
while (apple left)
  wait for snake to eat apple
  edge=[lastApple][currentApple]
  pheromone-level of edge*5
  run 5 iterations
end_while
return path
```

On the one hand, the central goal of the game from the user side is to eat all apples as fast as possible. However, in the game the users do not recognize that they are manipulating the pheromone-levels, because they just want to find the *fastest* path between all apples.

On the other hand, the goal the game from the algorithm side is to find the *shortest* path.

This approach allows us to generate a competitive time basted challenge for the users and decouples the fact that the algorithm itself is faster than that one with the iML-Approach because the users have not to wait for the interaction (the reader should play the game which is openly available.<sup>5</sup>

In one version the user receives a recommendation for the next apple to eat, this is generated by the MMAS and represents the edge with the highest pheromone-level.

In the experiments three pre-generated graphs are used in order to get comparable results (see Figs. 2, 3 and 4).

<sup>5</sup><https://iml.hci-kdd.org/TravellingSnakesmanWS>

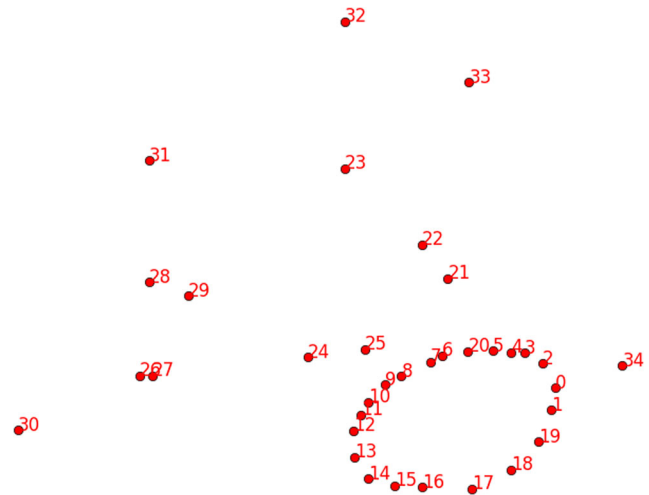


Fig. 2 Graph Level 1

## 6 Results and discussion

For our evaluation we split the game in two parts.

1. **iML-Part** The game as described above was played by 26 students (95 games played)
2. **aML-Part** For each game played we started a completely independent MMAS algorithm without human interaction

In a first step we multiplied the pheromone-values by 2 but, this had no significant impact on the ants, so we tried several other values and we came to the conclusion, that multiplying the value by 5 will increase the performance significantly, a larger value decreases the performance of the algorithm again, because a mistake of the human has major impact on the ants. As shown in Figs. 5, 6 and 7 our iML approach results in a performance increase (distance reduction) of the MMAS.

During the game the human travels a whole round through the graph, we have recognized (Figs. 8, 9 and 10) that the path taken by the human is longer than the path taken with the iML approach. We can also show that the shortest distance traveled with the aML approach is significant longer than the shortest distance traveled with the iML approach ( $\alpha = 95\%$ ).

To investigate the difference between the aML and the iML group, we conducted a repeated measure analysis of variance (ANOVA) for the independent variables level and group. The variable level, ranging from 1 to 3 with an increasing difficulty, is the repeated factor since all participants played the levels consecutively. The ANOVA yielded a significant main effect of the factor level [ $F(2, 189) = 79546.172, p < .001$ ]. This effect seems trivial, though, since the levels require traveling different distances to collect all apples, it is interesting. Even more

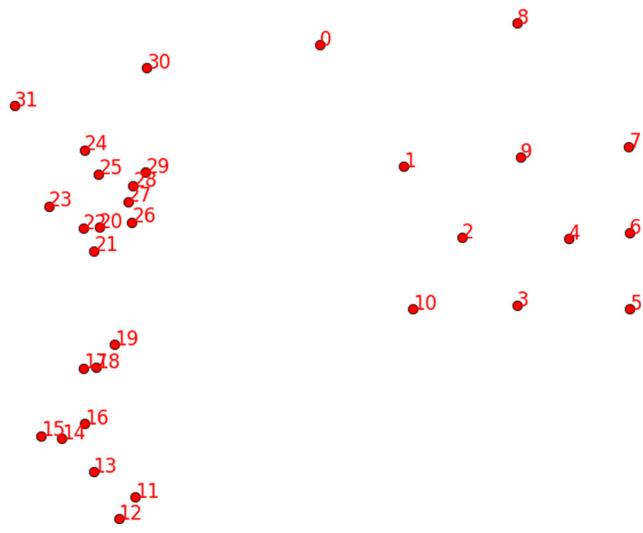


Fig. 3 Graph Level 2

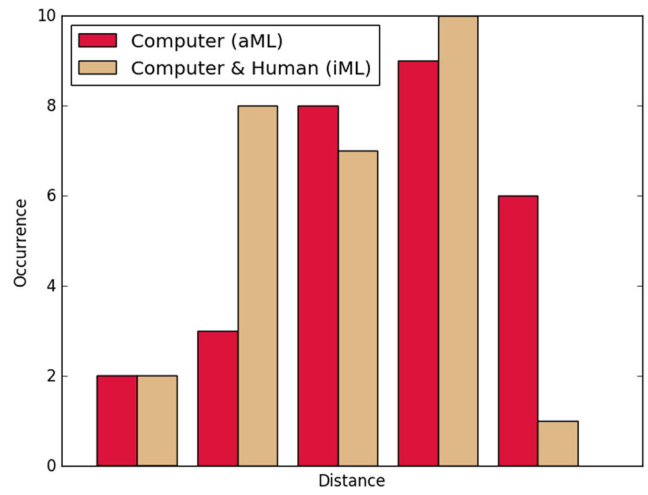


Fig. 6 Distance distribution of Level 2; 28 games played

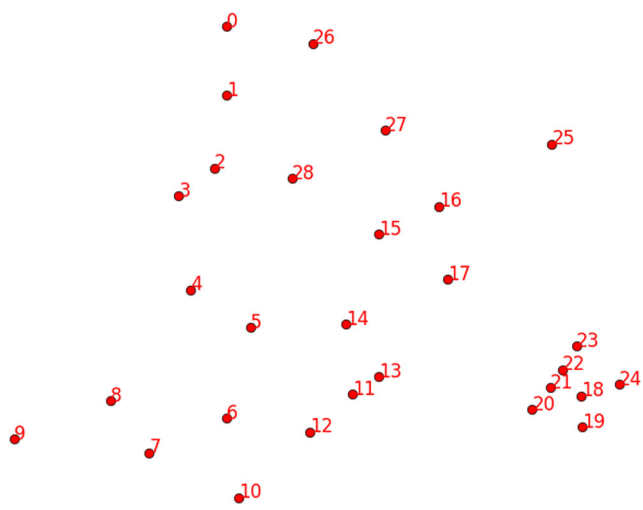


Fig. 4 Graph Level 3

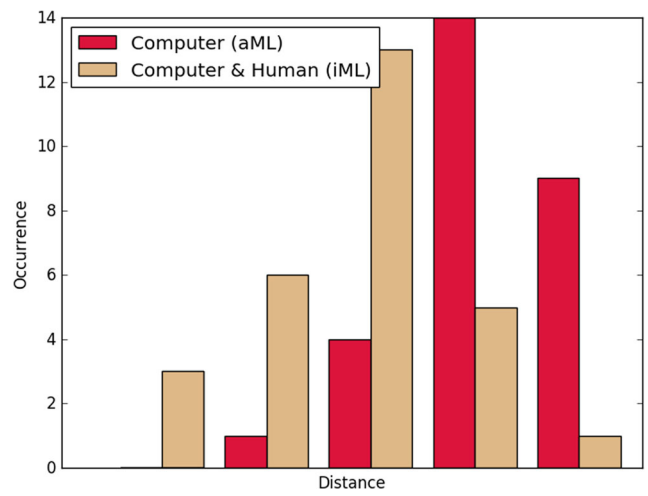


Fig. 7 Distance distribution of Level 3; 28 games played

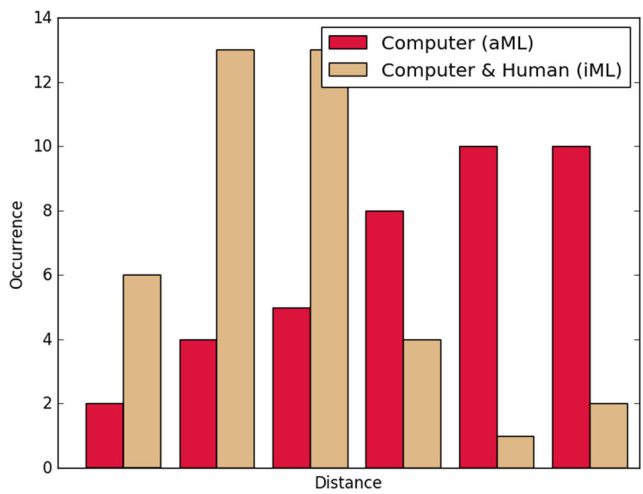


Fig. 5 Distance distribution of Level 1; 39 games played

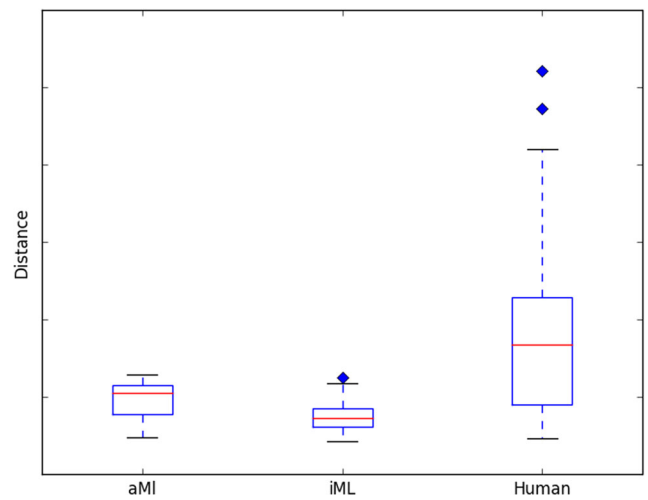


Fig. 8 Distance traveled of Level 1; 39 games played



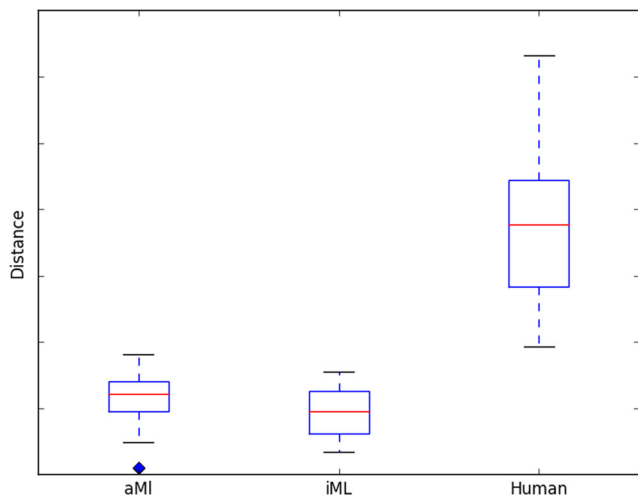


Fig. 9 Distance traveled of Level 2; 28 games played

interesting is the factor groups, where we found a significant main effect as well [ $F(1, 189) = 33.951, p < .001$ ]. At level 1, the mean of group aML was 4489802.48 (SD = 109628.351), the mean of group iML was 4376090.665 (SD = 94430.853). At level 2, the mean of group aML was 36281284.86 (SD = 855204.253), the mean of group iML was 35839124.63 (SD = 722500.697). At level 3, the mean of group aML was 44247653.59 (SD = 713300.268), the mean of group iML was 43233333.61 (SD = 865187.636). Across all levels, group iML resulted in shorter distances traveled. On the basis of the absolute distances, a comparison of the levels is not possible since the minimum distance varies. Thus, we focused on the differences between both groups. Instead of computing the difference with each trial, a more reasonable way is to compute the difference between group iML and the average of all computer only (aML) trials. To be able to compare all levels, we transformed the distances into a range between 0 and 1, which can be

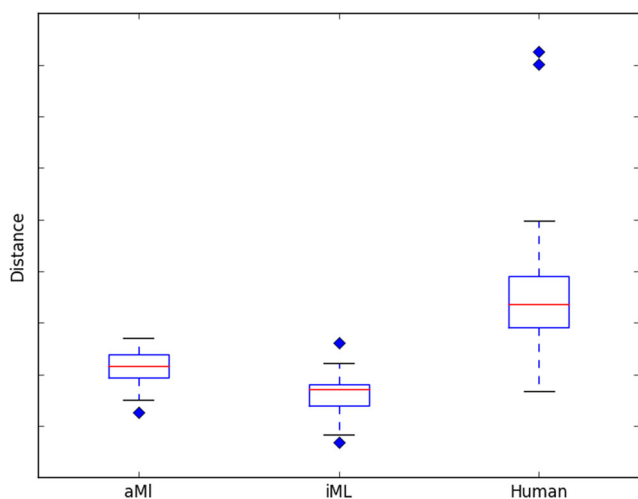


Fig. 10 Distance traveled of Level 3; 28 games played

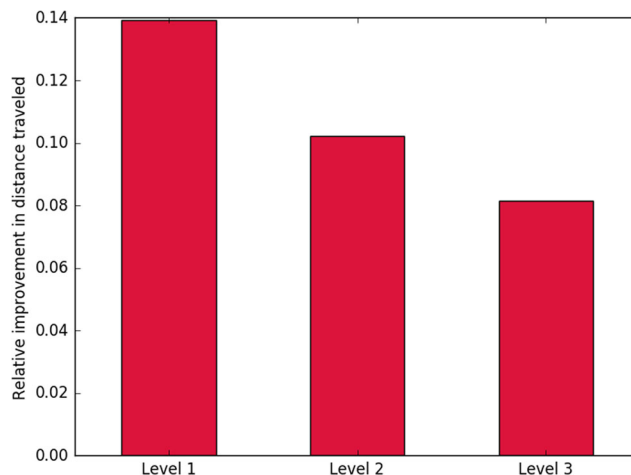


Fig. 11 Relative improvement in distance traveled (y axis) in group iML as opposed to the average distance of group aML for the 3 levels (x axis)

considered as the improvement in group iML as opposed to the average of group aML. The results are shown in Fig. 11 The relative improvement for level 1 was .1394, for level 2 .1021 and for level 3 .0814. One-sample t-tests computed for each level yielded that this improvement is significant for each level. *Level1* :  $t(38) = 7.519, p < .001$ ; *level2* :  $t(26) = 4.310, p < .001$ ; *level3* :  $t(27) = 3.346424, p = .002$ , which is significant.

In this study we found clear indications that the human interventions in the path finding processes results in generally better results, that is, distances traveled to collect all apples. This is reflected also in the absolute minimal distance traveled across all trials. For two of the three levels, the minimum was obtained in group iML (Table 1). In future steps, we will look into the mechanisms and specific characteristics of human interventions that improve the algorithm. We hypothesize that the superiority of humans may lie in the intuitive overview that is obtained by humans at the first sight. On the basis of the present results, which confirm that human interventions can improve the algorithm, we will systematically design game levels in favor of the human and the algorithm and repeat the experiment in future work on a large scale.

Table 1 Absolute minimum distances obtained across groups and levels

	aML	iML	Diff
Level 1	4242192.5568	4215015.4717	27177.0851
Level 2	34178226.0850	34680651.6358	-502425.5508
Level 3	42529746.1429	41378863.0008	1150883.1421

**Table 2** Example Human-Interaction-Matrix (HIM) and Human-Impact-Factor (HIF)

a)					b)						
	A	B	C	D	E		A	B	C	D	E
A	N	0	0	0	0	A	N	0	0	0	0
B	0	N	0	0	0	B	0	N	0.5	0	0
C	0	0	N	0	0	C	0	0.5	N	0	0.1
D	0	0	0	N	0	D	0	0	0	N	0
E	0	0	0	0	N	E	0	0	0.1	0	N

a) Initially, no human intervention. The loops are forbidden (N on the main diagonal). b) The human sets 50% chances to use the edge (B C) and 10% chances to follow (C E). In the other 40% of the cases, the ACO solver proceeds on its own, but these two edges are forbidden

### 7 Future outlook

A possible further extension is the Human-Interaction-Matrix (HIM) and the Human-Impact-Factor (HIF) represent the tools chosen here for integrating the human into an artificial Ant Colony solver. Their definition and their usage by the artificial agents were designed as to have no impact on the colony physical evolution (the pheromone deposited on edges). These two new structures only influences the colony decisional feature.

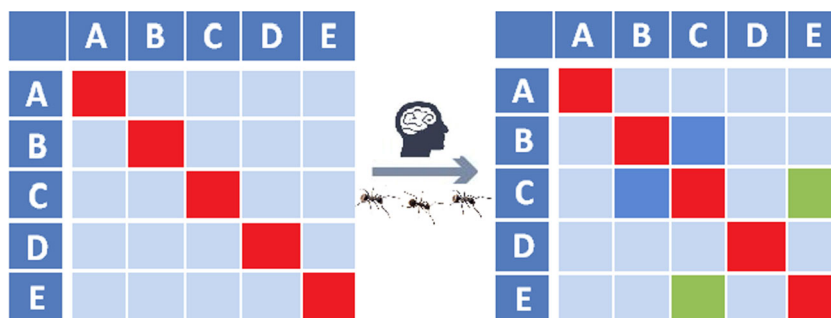
**Human-Interaction-Matrix (HIM)** HIM is a symmetric, positive matrix of dimension  $n$ , where  $n$  is the TSP instance dimension. The human sets its non-zero values as probabilities for realizing his/her decisions (Table 2). The sums on each column and each row must be at most 1. If such a sum is 0, then the ant staying in the corresponding vertex chooses the next vertex completely human-free. If the sum is 1, then the ant beginning its tour is just a human-reactive agent, with no decision power; the same situation is when all the available vertices have non-zero HIM values. The other cases model a human-ant collaboration as solution construction. This matrix may be dynamically modified by the

human’s decisions. If the human decides that one edge is no longer of interest, the corresponding two HIM values (due to the symmetry) are set to zero and the ants will proceed without human intervention. The modification of HIM and the solutions’ construction are two asynchronous processes.

**Human-Impact-Factor (HIF)** The HIF is the variable interpretation of HIM by each ant. When an ant begins its tour in vertex B, there are 50% chances to go to C. In the other 50% of the cases, when the ant does not go to C, it moves based on the algorithm’s transition rules, but C is excluded from the set of destinations. When an ant begins its tour in vertex C, there are 50% chances to go to B and 10% chances to go to E (Fig. 12). If neither B nor E are chosen, then the implemented algorithm decides where to go, excluding B and E from the available vertices. During the solution construction, HIM is the sum of the values from the corresponding column (or row) for the available moves. The situation when a tour is already started is discussed in the following section.

Of course, other operational methods could be imagined in order to include the human as a procedural module into a solver. The solving process and the human activity here are designed to run sequentially. More elaborated implementations could allow parallel executions, with sophisticated communication frameworks and unified (human, computer, collaborative) control.

The current implementation and the previous version from [26] could be further developed or used as local search methods. Extensions could consider edge blocking, when ants are forced to obey the edge exclusion from the available choices, or the asymmetric options for the human settings. If a larger instance clearly has several vertex clusters, then it can be split into several smaller instances, and the method presented here could be used to solve each cluster. The reason for the clusterization strategy is that the humans are very good in solving reasonably small TSP instances. Humans also could provide advices at this final step, as they



**Fig. 12** Example of the human interaction in guiding ants, based on Table 2. Initially the matrix HIM is empty (red squares mean loops are excluded); HIM=0.5 (blue squares) guide the ants to use the available edge (B C) in 50% of the cases. HIM=0.1 (green squares) guide the

ants to use the available edge (C E) in 10% of the cases. If the edges (B C) and (C E) are not used, then the ants move based on their transition rule only to the available vertices having zero values in HIM (light blue squares)

could observe topological properties, unavailable to artificial agents; other similar ideas are presented in [79–81].

## 8 Conclusion

In this work we have shown that the iML approach [11] can be used to enhance the results provided by the current TSP solvers. The human-computer co-operation is used for developing a new model, which was successfully instantiated [21]. This investigation complements other rewarding approaches. The repeated human interventions can orient the search through promising solutions. Gamification uses the theories of game playing in various contexts, for solving difficult problems. In such cases, gamification can be used as the human and the computer form a coalition, having the same goal. Several future research directions can be opened by our work. One interesting investigation is to translate the iML ideas in order to solve other similar real-world situations, for example on protein folding. Another challenge is to scale up this implementation on complex software packages. The most interesting part is surely to investigate re-traceability, interpretability and understandability towards explainable-AI. Here we envision insights in causality research (a.k.a. explanatory research to identify cause-and-effect relationships [82]), as well as practical applications in the medical domain, e.g. for education and decision support.

**Acknowledgements** We thank our international colleagues from the HCI-KDD and IET-CIS-IEEE networks for valuable discussions. We thank our students who helped to test our algorithms and provided valuable feedback, and Andrej Mueller from the Holzinger group for game design.

**Funding Information** Open access funding provided by Medical University of Graz.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## References

1. Turing AM (1950) Computing machinery and intelligence. *Mind* 59(236):433–460
2. Feurer M, Klein A, Eggenberger K, Springenberg J, Blum M, Hutter F (2015) Efficient and robust automated machine learning. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) *Advances in neural information processing systems* (NIPS 2015). Neural Information Processing Systems Foundation, Inc., pp 2962–2970
3. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
4. Hinton G, Li D, Yu D, Dahl GE, Mohamed A-r, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, Kingsbury B (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process Mag* 29(6):82–97
5. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, Chen Y, Lillitcrap T, Hui F, Sifre L, van den Driessche G, Graepel T, Hassabis D (2017) Mastering the game of go without human knowledge. *Nature* 550(7676):354–359
6. Richards N, Moriarty DE, Miiikkulainen R (1998) Evolving neural networks to play go. *Appl Intell* 8(1):85–96
7. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542(7639):115–118
8. Setio AAA, Traverso A, De Bel T, Berens MSN, van den Bogaard C, Cerello P, Chen H, Dou Q, Fantacci ME, Geurts B (2017) Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge. *Med Image Anal* 42:1–13
9. Ghafoorian M, Karssemeijer N, Heskes T, Uden IWM, Sanchez CI, Litjens G, de Leeuw F-E, van Ginneken B, Marchiori E, Platel B (2017) Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities. *Sci Rep* 7(1):5110
10. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JAWM, van Ginneken B, Sánchez CI (2017) A survey on deep learning in medical image analysis. *Med Image Anal* 42:60–88
11. Holzinger A (2016) Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Inf* 3(2):119–131
12. Holzinger A, Biemann C, Pattichis CS, Kell DB (2017) What do we need to build explainable ai systems for the medical domain? [arXiv:1712.09923](https://arxiv.org/abs/1712.09923)
13. Bologna G, Hayashi Y (2017) Characterization of symbolic rules embedded in deep dimlp networks: a challenge to transparency of deep learning. *J Artif Intell Soft Comput Res* 7(4):265–286
14. Malle B, Kieseberg P, Weippl E, Holzinger A (2016) The right to be forgotten: towards machine learning on perturbed knowledge bases. In: *Springer lecture notes in computer science LNCS 9817*. Springer, Heidelberg, pp 251–256
15. Newman AL (2015) What the “right to be forgotten” means for privacy in a digital age. *Science* 347(6221):507–508
16. Malle B, Kieseberg P, Holzinger A (2017) Do not disturb? Classifier behavior on perturbed datasets. In: *Machine learning and knowledge extraction, IFIP CD-MAKE*, lecture notes in computer science LNCS, vol 10410. Springer, Cham, pp 155–173
17. Malle B, Giuliani N, Kieseberg P, Holzinger A (2018) The need for speed of ai applications: performance comparison of native vs browser-based algorithm implementations. [arXiv:1802.03707](https://arxiv.org/abs/1802.03707)
18. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell* 35(8):1798–1828
19. McCarthy J (2007) From here to human-level ai. *Artif Intell* 171(18):1174–1182
20. Wilson AG, Dann C, Lucas C, Xing EP (2015) The human kernel. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) *Advances in neural information processing systems*, NIPS 2015, vol 28. NIPS Foundation, pp 2836–2844
21. Amershi S, Cakmak M, Knox WB, Kulesza T (2014) Power to the people: the role of humans in interactive machine learning. *AI Mag* 35(4):105–120

22. Shyu C-R, Brodley CE, Kak AC, Kosaka A, Aisen AM, Broderick LS (1999) Assert: a physician-in-the-loop content-based retrieval system for hrcr image databases. *Comput Vis Image Underst* 75(1–2):111–132
23. Schirmer G, Erdogmus D, Chowdhury K, Padir T (2013) The future of human-in-the-loop cyber-physical systems. *Computer* 46(1):36–45
24. Holzinger A (2016) Interactive machine learning (iml). *Informatik Spektrum* 39(1):64–68
25. Holzinger A, Malle B, Kieseberg P, Roth PM, Müller H, Reihs R, Zatloukal K (2017) Towards the augmented pathologist: Challenges of explainable-ai in digital pathology. arXiv:1712.06657
26. Holzinger A, Plass M, Holzinger K, Crisan GC, Pinteá C-M, learning VP (2016) Towards interactive machine (iml): applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach. In: Springer lecture notes in computer science LNCS 9817. Springer, Heidelberg, pp 81–95
27. Holzinger A, Plass M, Holzinger K, Crisan GC, Pinteá C-M, Palade V (2017) A glass-box interactive machine learning approach for solving np-hard problems with the human-in-the-loop. arXiv:1708.01104
28. Crescenzi P, Goldman D, Papadimitriou C, Piccolboni A, Yannakakis M (1998) On the complexity of protein folding. *J Comput Biol* 5(3):423–465
29. Papadimitriou CH (1977) The euclidean travelling salesman problem is np-complete. *Theor Comput Sci* 4(3):237–244
30. Macgregor JN, Ormerod T (1996) Human performance on the traveling salesman problem. *Percept Psychophys* 58(4):527–539
31. Vickers D, Butavicius M, Lee M, Medvedev A (2001) Human performance on visually presented traveling salesman problems. *Psychol Res* 65(1):34–45
32. Holzinger K, Palade V, Rabadan R, Holzinger A (2014) Darwin or lamarck? Future challenges in evolutionary algorithms for knowledge discovery and data mining. In: Interactive knowledge discovery and data mining in biomedical informatics: state-of-the-art and future challenges. Lecture notes in computer science LNCS 8401. Springer, Heidelberg, pp 35–56
33. Gigerenzer G, Gaissmaier W (2011) Heuristic decision making. *Annu Rev Psychol* 62:451–482
34. O’Sullivan S, Holzinger A, Wichmann D, Saldiva PHN, Sajid MI, Zatloukal K (2018) Virtual autopsy: machine learning and ai provide new opportunities for investigating minimal tumor burden and therapy resistance by cancer patients. *Autopsy Case Rep*, 8(1)
35. Hofmann T, Schoelkopf B, Smola AJ (2008) Kernel methods in machine learning. *Ann Statist* 36(3):1171–1220
36. Griffiths TL, Lucas C, Williams J, Kalish ML (2008) Modeling human function learning with gaussian processes. In: Koller D, Schuurmans D, Bengio Y, Bottou L (eds) *Advances in neural information processing systems (NIPS 2008)*, vol 21. NIPS, pp 553–560
37. Wilson AG, Gilboa E, Nehorai A, Cunningham JP (2014) Fast kernel learning for multidimensional pattern extrapolation. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) *Advances in neural information processing systems (NIPS 2014)*. NIPS foundation, pp 3626–3634
38. Auer P, Long PM, Maass W, Woeginger GJ (1995) On the complexity of function learning. *Mach Learn* 18(2–3):187–230
39. Wilson AG, Adams RP (2013) Gaussian process kernels for pattern discovery and extrapolation. In: *International conference on machine learning (ICML 2013)*, vol 28. JMLR, pp 1067–1075
40. Steinwart I, Scovel C (2012) Mercer’s theorem on general domains: on the interaction between measures, kernels, and rkhs. *Constr Approx* 35(3):363–417
41. Xu F, Tenenbaum JB (2007) Word learning as Bayesian inference. *Psychol Rev* 114(2):245–272
42. Tenenbaum JB, Kemp C, Griffiths TL, Goodman ND (2011) How to grow a mind: statistics, structure, and abstraction. *Science* 331(6022):1279–1285
43. Thaker P, Tenenbaum JB, Gershman SJ (2017) Online learning of symbolic concepts. *J Math Psychol* 77:10–20
44. Chater N, Tenenbaum JB, Yuille A (2006) Probabilistic models of cognition: conceptual foundations. *Trends Cogn Sci* 10(7):287–291
45. Steyvers M, Tenenbaum JB, Wagenmakers E-J, Blum B (2003) Inferring causal networks from observations and interventions. *Cogn Sci* 27(3):453–489
46. Wolpert DM, Ghahramani Z, Jordan MI (1995) An internal model for sensorimotor integration. *Science* 269(5232):1880–1882
47. Lucas CG, Griffiths TL, Williams JJ, Kalish ML (2015) A rational model of function learning. *Psychon Bull Rev* 22(5):1193–1215
48. Wang F-Y, Zhang JJ, Zheng X, Wang X, Yuan Y, Dai X, Zhang J, Yang L (2016) Where does alpha go: from church-turing thesis to alpha go thesis and beyond. *IEEE/CAA J Autom Sinica* 3(2):113–120
49. Holzinger A (2013) Human—computer interaction & knowledge discovery (HCI-KDD): what is the benefit of bringing those two fields to work together? In: Cuzzocrea A, Kittl C, Simos DE, Weippl E, Xu L (eds) *Multidisciplinary research and practice for information systems*, springer lecture notes in computer science LNCS 8127. Springer, Heidelberg, pp 319–328
50. MacGregor JN, Chu Y (2011) Human performance on the traveling salesman and related problems: a review. *J Probl Solv* 3(2):119–150
51. Monasson R, Zecchina R, Kirkpatrick S, Selman B, Troyansky L (1999) Determining computational complexity from characteristic ‘phase transitions’. *Nature* 400(6740):133–137
52. Best BJ, Simon HA (2000) Simulating human performance on the traveling salesman problem. In: *Proceedings of the third international conference on cognitive modeling*. Universal Press, pp 42–49
53. Knill DC, Pouget A (2004) The bayesian brain: the role of uncertainty in neural coding and computation. *Trends Neurosci* 27(12):712–719
54. Tenenbaum JB, Griffiths TL, Kemp C (2006) Theory-based Bayesian models of inductive learning and reasoning. *Trends Cogn Sci* 10(7):309–318
55. Acuna DE, Parada V (2010) People efficiently explore the solution space of the computationally intractable traveling salesman problem to find near-optimal tours. *PloS One* 5(7):1–10
56. Rooij IV, Stege U, Schactman A (2003) Convex hull and tour crossings in the euclidean traveling salesperson problem: implications for human performance studies. *Mem Cogn* 31(2):215–220
57. Crisan GC, Nechita E, Palade V (2016) Ant-based system analysis on the traveling salesman problem under real-world settings. In: *Combinations of intelligent methods and applications*. Springer, pp 39–59
58. Laporte G (1992) The traveling salesman problem: an overview of exact and approximate algorithms. *Eur J Oper Res* 59(2):231–247
59. Applegate DL, Bixby RE, Chvatal V, Cook WJ (2006) *The traveling salesman problem: a computational study*. Princeton University Press
60. Korostensky C, Gonnet GH (2000) Using traveling salesman problem algorithms for evolutionary tree construction. *Bioinformatics* 16(7):619–627

61. Karp RM (1993) Mapping the genome: some combinatorial problems arising in molecular biology. In: Proceedings of the twenty-fifth annual ACM symposium on theory of computing (STOC 1993). ACM, pp 278–285
62. Nelson CA, Miller DJ, Oleynikov D (2008) Modeling surgical tool selection patterns as a “traveling salesman problem” for optimizing a modular surgical tool system. *Stud Health Technol Inform* 132:322–326
63. Kirn S (2002) Ubiquitous healthcare: the onkonet mobile agents architecture. In: *Net. ObjectDays: international conference on object-oriented and internet-based technologies, concepts, and applications for a networked world*. Springer, pp 265–277
64. Michael RG, David SJ (1979) *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco
65. Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Manag Sci* 6(1):80–91
66. Gouveia L, Voss S (1995) A classification of formulations for the (time-dependent) traveling salesman problem. *Eur J Oper Res* 83(1):69–82
67. Crişan GC, Pinteá C-M, Palade V (2017) Emergency management using geographic information systems: application to the first romanian traveling salesman problem instance. *Knowl Inf Syst* 50(1):265–285
68. Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B: Cybern* 26(1):29–41
69. Blum C (2005) Ant colony optimization: introduction and recent trends. *Phys Life Rev* 2(4):353–373
70. Franks NR, Dechaume-Moncharmont F-X, Hanmore E, Reynolds JK (2009) Speed versus accuracy in decision-making ants: expediting politics and policy implementation. *Philos Trans R Soc B: Biol Sci* 364(1518):845–852
71. Dorigo M, Stützle T (2004) *Ant colony optimization*. MIT Press, Boston
72. Croes GA (1958) A method for solving traveling-salesman problems. *Oper Res* 6(6):791–812
73. Bentley JJ (1992) Fast algorithms for geometric traveling salesman problems. *ORSA J Comput* 4(4):387–411
74. Lin S (1965) Computer solutions of the traveling salesman problem. *Bell Syst Techn J* 44(10):2245–2269
75. Pinteá C-M, Dumitrescu D (2005) Improving ant systems using a local updating rule. In: *Seventh international symposium on symbolic and numerical algorithms for scientific computing*. IEEE, pp 295–298
76. Ventresca M, Ombuki BM (2004) Ant colony optimization for job shop scheduling problem. Technical Report CS-04-04, St. Catharines, Ontario
77. Angus D, Hendtlass T (2005) Dynamic ant colony optimisation. *Appl Intell* 23(1):33–38
78. Stützle T, Hoos HH (2000) Max–min ant system. *Fut Gen Comput Syst* 16(8):889–914
79. Hund M, Boehm D, Sturm W, Sedlmair M, Schreck T, Ullrich T, Keim DA, Majnarić L, Holzinger A (2016) Visual analytics for concept exploration in subspaces of patient groups: making sense of complex datasets with the doctor-in-the-loop. *Brain Inf* 3(4):233–247
80. Stoean C, Stoean R (2014) *Support vector machines and evolutionary algorithms for classification*. Springer, Cham
81. Matei O, Pop PC, Válean H (2013) Optical character recognition in real environments using neural networks and k-nearest neighbor. *Appl Intell* 39(4):739–748
82. Peters J, Janzing D, Schölkopf B (2017) *Elements of causal inference: foundations and learning algorithms*. Cambridge



**Andreas Holzinger** is heading the Holzinger Group, HCI-KDD research unit, at the Medical University Graz; he holds a PhD in Cognitive Science and a second PhD (habilitation) in applied computer science from Graz University of Technology; he has pioneered in interactive machine learning with the human-in-the-loop and works on augmenting human intelligence with artificial intelligence; currently he is Visiting Professor for Machine Learning in Health Informatics at Vienna University of Technology, more info see: <http://www.aholzinger.at>.



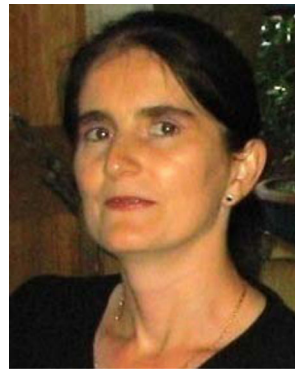
**Markus Plass** BSc, finishes currently his Master in biomedical engineering at the Holzinger Group with a topic on interactive machine learning with the human-in-the-loop and focusing on the travelling salesman problem.



**Michael Kickmeier-Rust** is Professor of Psychology at the University of Teacher Education St. Gallen, Switzerland. He holds a PhD in Cognitive Psychology from Graz University and has a strong technical background in computer science. His research focuses on pedagogical diagnostics, smart embedded competence assessment as well as serious games and embedded, game-based assessment, training and learning analytics with heterogeneous multi-modal and multi-source data.



**Katharina Holzinger** BA, BA, currently finalizing her MSc in Earth Sciences is now working at the Austrian National Defense Academy in Vienna and as junior researcher at Secure Business Austria, SBA-Research Vienna, focusing on causality research.



**Camelia-M. Pinte** is Associate Professor for informatics at the Technical University of Cluj-Napoca; she works in combinatorial optimization, artificial intelligence, complex Systems, metaheuristics, natural computing, multi-agent systems and applied mathematics; she is member of several international research groups including the Holzinger Group, HCI-KDD.



**Gloria Cerasela Crișan** is Associate Professor for informatics at the Department of Mathematics and Informatics, Faculty of Sciences of the “Vasile Alecsandri” University of Bacău, Romania. Her research domain is in solving computationally hard combinatorial optimization problems, such as real-world transportation problems with multiple restrictions.



**Vasile Palade** is a Reader in Pervasive Computing at Coventry University, UK, and previously held positions at the University of Oxford, UK, among others; his research interests include machine learning with applications to image processing, network and data analysis, and health applications. He is an IEEE Senior Member and a member of the IEEE Computational Intelligence Society and head of the IFIP workgroup computational intelligence.

## Affiliations

Andreas Holzinger<sup>1</sup>  · Markus Plass<sup>1</sup> · Michael Kickmeier-Rust<sup>2</sup> · Katharina Holzinger<sup>1</sup> · Gloria Cerasela Crișan<sup>3</sup> · Camelia-M. Pinte<sup>4</sup> · Vasile Palade<sup>5</sup>

Michael Kickmeier-Rust  
michael.kickmeier@phsg.ch

Gloria Cerasela Crișan  
ceraselacrisan@ub.ro

Camelia-M. Pinte  
dr.camelia.pinte@ieee.org

Vasile Palade  
vasile.palade@coventry.ac.uk

<sup>1</sup> Institute for Medical Informatics, Statistics & Documentation, Medical University Graz, Graz, Austria

<sup>2</sup> Institute for Educational Assessment, University of Teacher Education St. Gallen, St. Gallen, Switzerland

<sup>3</sup> Faculty of Sciences, Vasile Alecsandri University of Bacău, Bacău, Romania

<sup>4</sup> Technical University of Cluj-Napoca, Cluj-Napoca, Romania

<sup>5</sup> School of Computing, Electronics & Maths, Faculty of Engineering, Environment & Computing, Coventry University, Coventry, UK