

Article

ExtrIntDetect—A New Universal Method for the Identification of Intelligent Cooperative Multiagent Systems with Extreme Intelligence

László Barna Iantovics ^{1,*} , Roumen Kountchev ² and Gloria Cerasela Crişan ³

¹ Department of Informatics, Faculty of Sciences and Letters, University of Medicine, Pharmacy, Sciences and Technology of Targu Mures, 540124 Târgu Mureş, Romania

² Department of Radio Communications and Video Technologies, Faculty of Telecommunications, Technical University of Sofia, 1000 Sofia, Bulgaria

³ Department of Mathematics and Informatics, Faculty of Sciences, Vasile Alecsandri University of Bacău, 600115 Bacău, Romania

* Correspondence: ibarna@science.upm.ro

Received: 4 July 2019; Accepted: 8 August 2019; Published: 4 September 2019



Abstract: In this research, we define a specific type of performance of the intelligent agent-based systems (IABSs) in terms of a difficult problem-solving intelligence measure. Many studies present the successful application of intelligent cooperative multiagent systems (ICMASs) for efficient, flexible and robust solving of difficult real-life problems. Based on a comprehensive study of the scientific literature, we conclude that there is no unanimous view in the scientific literature on machine intelligence, or on what an intelligence metric must measure. Metrics presented in the scientific literature are based on diverse paradigms. In our approach, we assume that the measurement of intelligence is based on the ability to solve difficult problems. In our opinion, the measurement of intelligence in this context is important, as it allows the differentiation between ICMASs based on the degree of intelligence in problem-solving. The recent *OutIntSys* method presented in the scientific literature can identify systems with outlier high and outlier low intelligence from a set of studied ICMASs. In this paper, a novel universal method called *ExtrIntDetect*, defined on the basis of a specific series of computing processes and analyses, is proposed for the detection of the ICMASs with statistical outlier low and high problem-solving intelligence from a given set of studied ICMASs. *ExtrIntDetect* eliminates the disadvantage of the *OutIntSys* method with respect to its limited robustness. The recent symmetric *MetrIntSimil* metric presented in the literature is capable of measuring and comparing the intelligence of large numbers of ICMASs and based on their respective problem-solving intelligences in order to classify them into intelligence classes. Systems whose intelligence does not statistically differ are classified as belonging to the same class of intelligent systems. Systems classified in the same intelligence class are therefore able to solve difficult problems using similar levels of intelligence. One disadvantage of the symmetric *MetrIntSimil* lies in the fact that it is not able to detect outlier intelligence. Based on this fact, the *ExtrIntDetect* method could be used as an extension of the *MetrIntSimil* metric. To validate and evaluate the *ExtrIntDetect* method, an experimental evaluation study on six ICMASs is presented and discussed.

Keywords: computational-hard problem; symmetric travelling salesman problem; similarity in intelligence; machine intelligence; intelligent cooperative multiagent system; system with outlier intelligence; machine intelligence quotient; symmetric intelligence metric

1. Introduction

The evaluation of the performance of computing systems includes diverse research topics, including the performance evaluation of cognitive packet networks [1], cognitive packet networks where network worms operate [2], peering-agreements among systems for peer-to-peer traffic that has some kind of autonomy [3], the Internet [4], large-scale SCI multiprocessors [5], algorithms [6], and supply chain management based on Internet of Things technology [7]. Performance evaluation in terms of intelligence is rarely addressed for IABSs, where the property of intelligence requires more complex analysis than that carried out in a usual performance evaluation. IABSs are used for solving a variety of real-life problems: quality measurement of Golden bleached raisins [8], predicting the machining performance parameters of Inconel 690 [9], intelligent diagnosis of certain diseases [10], intelligent support of automatic metrological measurement results [11], intelligent fault diagnosis of wind turbines [12], and the detection and segmentation regions of interest in retinal images [13]. Many IABSs are ICMASs [14]. ICMASs can frequently solve problems with various types of computational complexity. ICMASs are particularly important with respect to the fact that even in CMASs composed of less complex agents, increased intelligence could emerge at a system level [15] as result of flexible, robust and efficient cooperative problem-solving. The cooperative swarms of ground and aerial vehicles with agent properties can be mentioned as examples of developed ICMASs. Kwon and Pack [16] proposed a robust mobile target localization method for cooperative unmanned aerial vehicles. Saska et al. [17] proposed a model for the coordination and navigation of formations of heterogeneous unmanned aerial and ground vehicles (MAV-UGV).

In this research, we consider the expression of a specific type of performance of ICMASs in terms of the intelligence required to solve difficult problems. We carried out long-term study and research into the relevant literature; based on this, we concluded that there is no unanimous definition for, or even a view on, the intelligence of ICMASs. However, even in this situation, machine intelligence can be measured, and its measurement could be effective for the differentiation of ICMASs based on their intelligence. This is somehow similar to human intelligence, for which there is also no unanimous definition. This fact is based on the enormously high complexity of the human brain and human cognition. Many effective intelligence tests for the measurement of human intelligence have been proposed. The most frequently used human IQ tests are carried out using: the Stanford-Binet scale [18–20], Raven's Progressive Matrices [21,22], the Wechsler Adult Intelligence Scale [23,24], and the Kaufman Assessment Battery for Children [25–27]. Neisser et al. performed a study [28] that showed that the measurement of intelligence based on IQ test scores without any other considerations ignores many important aspects that are specific to human cognition. It has been proved in some studies that the measurement of human intelligence has applicative effectiveness. According to [29], the human intelligence quotient (IQ) influences, among other things, job and school performance. However, this depends on other factors such as specialty knowledge and, to some degree, commonsense knowledge.

In our approach, it is assumed that the measurement of the machine intelligence of ICMASs is based on some kind of ability to solve difficult problems. The *Machine Intelligence Quotient (MIQ)*, the indicator of the central intelligence tendency of an ICMAS, is obtained by statistical computations after measuring how intelligently a set of difficult problems has been solved. A universal method is effectively being proposed, called the *Robust Extreme Intelligence Detection Method (ExtrIntDetect)*, for the detection of ICMASs with statistically extremely low and high MIQ in solving problems from a set of considered ICMASs. The novelty of the proposal lies in designing a specific series of computing processes and analyses. The usefulness of the method consists in indicating the brilliant systems that are able to most intelligently solve difficult problems and the poor systems that will solve problems with unexpectedly low intelligence. *ExtrIntDetect* represents an extension of a novel method for detecting outlier intelligence of ICMASs called *OutIntSys* [30]. The purpose of extending the *OutIntSys* method was to conserve its properties, while at the same time making it more robust.

The symmetric *MetrIntSimil* metric [31] is able to compare ICMASs based on the intelligence with which they solve difficult problems and classify them into intelligence classes. Systems that

are classified as belonging to the same class are able to solve the given problems with same level of intelligence. One limitation of *MetrIntSimil* is the fact that it is not able to identify extreme intelligence. The *ExtrIntDetect* method could be used as an extension of the *MetrIntSimil* metric in order to detect systems with extremely low and extremely high intelligence.

To evaluate and validate the *ExtrIntDetect* method, an illustrative experimental case study is presented. We studied six ICMASSs, all of which are considered intelligent in the scientific literature based on their efficient and flexible problem-solving, and specialize in solving the class of NP-hard problems.

The rest of this paper is organized as follows: In Section 2, some representative state-of-the-art metrics for the measurement of machine intelligence are presented. Section 3 presents the proposed *ExtrIntDetect* method. The experimental study for the validation of the *ExtrIntDetect* method is presented in Section 4. In Section 5, a discussion of the *ExtrIntDetect* method is presented. Section 6 presents the conclusions of this work.

2. State-of-the-Art Metrics Designed to Measure Machine Intelligence

There are very few developed intelligence metrics that are able to carry out different types of measurement of machine intelligence. In our paper published in 2018 [32], we presented a comprehensive review of recent research directions in measuring the IABSSs intelligence. In the following, we present some studies and research focused on metrics/methods for measuring machine intelligence.

According to the well-known work of Alan Turing, as formulated in 1950 [33], a computing system can be considered intelligent if a human judge could not classify it as being artificial or human based on how it responds to questions. Dowe and Hajek in 1998 [34] proposed a specific computational extension of the Turing Test. In 2000, Schreiner [35] analyzed the possibility of creating standard measurements for computing systems that would be able to classify them as intelligent. Park et al., in 2001 [36], studied the measurement of intelligence of human-machine systems, proposing a so-called intelligence task graph as an effective tool for modeling and analysis. In 2003, Sanghi and Dowe [37] presented an advanced computer program that had been successfully evaluated by means of some standard human IQ tests. A measure of performance in difficult and easy environments was defined in 2006 [38]. In 2007 [39], method for measuring intelligence based on the capacity to compare alternatives with different levels of complexity was proposed. In 2010, Hernández-Orallo [40] presented an interesting study related to the Turing Test. A universal anytime intelligence test was proposed by Hernández-Orallo and Dowe in 2010 [41]. The concept of universality was regarded as the advanced ability of a metric to measure artificial and biological intelligence at the same time. In [42], published in 2011, a metric for the measurement of intelligence in consideration of a hierarchy of sets of increasingly difficult environments was proposed. In 2014, Hernández-Orallo et al. [43] studied the development of universal metrics for the measurement of the capabilities of intelligent systems that was inspired by the psychometrics applied for the measurement of human intelligence. In 2015, Besold et al. [44] approached the important research question of whether machine intelligence could be measured using the same methods as human intelligence. In 2016, Chmait et al. [45] elaborated some dynamic intelligence tests for measuring combined intelligence in cooperative systems. In [30], published in 2017, a method called *OutIntSys* for the identification of ICMASSs with extreme intelligence was presented. Liu et al., in 2017 [46], presented a recent complex study related to the analysis of the machine intelligence quotient (MIQ). The recent symmetric *MetrIntSimil* metric, published in 2018 [31], is able to compare the intelligence of a large number of ICMASSs.

As a concluding remark related to state-of-the-art studies and research focusing on the measurement of intelligence, current intelligence metrics are founded on very different principles the measurement of machine intelligence. Based on this consideration, most of them cannot be effectively compared in any way. The main limitation of the designed intelligence metrics and methods presented in the literature consists in the lack of the ability to identify ICMASSs with statistically extreme intelligence from a set of studied ICMASSs. The maximum or poorest intelligence does not necessarily mean statistically extreme (high/low) intelligence. The expectation of such a method/metric

consists in the identification of systems that statistically have a much higher (or lower) problem-solving intelligence than the others. The *OutIntSys* method is designed with this purpose in mind, but it has limited robustness. The *MetrIntSimil* metric is appropriate for the measurement and comparison of intelligence, and the classification of intelligence into intelligence classes, but unfortunately it is not able to detect statistical extremes.

3. The Proposed *ExtrIntDetect* Method

In this section, a novel method called *Robust Extreme Intelligence Detection Method (ExtrIntDetect)* is proposed for the detection of the ICMASs with extreme intelligence from a set of studied ICMASs, denoted $ICM = \{ICM_1, ICM_2, \dots, ICM_z\}$. Each studied ICMAS comprises a specific number of agents. Each ICMAS can have any number of agents, and could possess any architecture. There are no restrictions related to either the studied systems having an equal number of agents, or to their having the same architectural homogeneity. We will denote with $|ICM_r|$ the cardinality of ICM_r (i.e., the number of its agents).

Human evaluator (He) represents the human specialist (or the groups of human specialists, in the case of extremely complex systems) that performs the measurement of the machine intelligence of the studied ICMASs. A calculated *intelligence indicator (IntIndic)* gives a quantitative measure of the studied system's intelligence, which corresponds to an evaluated problem-solving intelligence measure. In the case of a particular set of studied ICMASs, the *He* must decide on the best-fitting intelligence indicator measure based on the envisioned type of machine intelligence.

$Prob = \{Prob_1, Prob_2, \dots, Prob_z\}$ denotes the set of problems used for the evaluation of problem-solving intelligence in the studied set of ICMASs. Each system's problem-solving intelligence is evaluated based on a specific set of problems. We will denote with $Prob_k = \{Prob_{k,1}, Prob_{k,2}, \dots, Prob_{k,m}\}$ the set of problems used for the problem-solving intelligence evaluation of ICM_k ($ICM_k \in ICM$). $|Prob_k|$, ($|Prob_k| = m$) represents the cardinality (i.e., the number of problems) used in the experimental intelligence evaluation of ICM_k . There is no requirement for the sets of problems $Prob_1, Prob_2, \dots$ and $Prob_z$ to be the same. There is no restriction that the cardinalities $|Prob_1|, |Prob_2|, \dots$ and $|Prob_z|$ of $Prob_1, Prob_2, \dots$ and $Prob_z$ to be the same. *He* is responsible in the case of each ICM_k for the establishment of $|Prob_k|$ and $Prob_k$ (as described in the *MPSI* algorithm). It was determined that the $|Prob_k|$ value should be at least 5 ($|Prob_k| \geq 5$) for the calculation of the intelligence quotient of the ICMAS. This minimum permitted value was set in order to allow the application of statistical testing for the verification of the data normality assumption. It is recommended, when possible, that intelligence evaluations be carried out for larger sets, $|Prob_k| \geq 10$, or even $|Prob_k| \geq 30$.

In the following, $Intellig_k = \{Intellig_{k,1}, Intellig_{k,2}, \dots, Intellig_{k,m}\}$ denotes the intelligence indicators obtained as a result of the $Prob_k$ problem-solving intelligence evaluations performed for a specific ICMAS denoted ICM_k .

If necessary, problem-solving intelligence can be calculated as the weighted sum of c types of intelligence components (1). These measure different aspects of the intelligence of the studied ICMASs.

$$\begin{aligned} IntIndic &= imp_1 \times ms_1 + imp_2 \times ms_2 + \dots + imp_c \times ms_c. \\ imp_1 + imp_2 + \dots + imp_c &= 1, imp_i > 0 \end{aligned} \quad (1)$$

where: ms_1, ms_2, \dots, ms_c represent the considered intelligence components measure at a particular problem-solving intelligence evaluation; $imp_1, imp_2, \dots, imp_c$ represent the weights associated with the intelligence components. The weight of a component quantifies the importance of that component in the intelligence evaluation measure.

The Scenario of Coalitions of Agent-Based Flying Drones

With to the aim of explaining the previously introduced notions, in the following a scenario is presented. A set of ICMASs is considered that is composed of coalitions of flying agent-based

drones denoted $ICM_1, ICM_2, \dots, ICM_n$. Each ICMAS is composed of a specific number of agents specialized in the delivery of products to customers. Each ICMAS solves problems in problem-solving cycles. A problem-solving cycle consists of a certain number of deliveries to clients by the drones that compose the respective ICMAS. A problem-solving cycle is finished when all the products have been delivered. A drone could undertake the task of a certain number of product deliveries to one or multiple destinations. Each visited client could be the recipient of one or more products. Each flying drone visits each client only once, at most. After all the deliveries have been completed, the drones should return to their starting locations. To determine the optimal route requirements for delivery, it is necessary that each drone solves a Travelling Salesman Problem.

He decides to evaluate the intelligence of each ICMAS with respect to the intelligence demonstrated in the delivery of the products. The intelligence of a particular flying drone in a flight is calculated based on two intelligence components, with weights imp_1 and imp_2 , established by *He*, based on the intelligence of the delivery. For the delivery of products (one or more) to clients (one or more) at home, a mark between 0 and 7 is given by *He*. The first intelligent component measure has a weight of 0.65 ($ms_1 = 0.65$). The second intelligence component measure is established as the mean of the marks with values between 0 and 7 that have been given by clients who have received products in that delivery, with the weight of 0.35 ($ms_2 = 0.35$). For instance, a scenario involving the delivery of some products to two clients is considered. Each client gives a mark, and the mean of the values is computed. The weight of this mean is marked as 0.35. The reason for choosing these weighting values by *He* lies in the fact that human assessors have deeper knowledge regarding ICMASs than the clients. The clients are common people who conclude their evaluation mostly based on their satisfaction with the delivery. The reason for the weighting of the evaluation of clients being no less than half of the weighting of the evaluation of the human specialist is that clients' satisfaction with deliveries is an important factor for the company that owns the drones. The final obtained intelligence indicator value based on the values of the two intelligence components will be a number in the interval $[0, 7]$. 0 signifies a system with no measurable intelligence. 7 signifies brilliant system (with the utmost possible intelligence).

The intelligence of a CMAS in a problem-solving cycle can be calculated as the average of the marks received by the contributing member agents of the CMAS.

He is in charge of selecting the type of intelligence that is to be measured by the *He*. This is the first step, and it is executed before the application of the *Measuring the Problem-Solving Intelligence* (MPSI) algorithm (Algorithm 1) for each studied ICMAS. By applying the MPSI algorithm for each analyzed ICMAS, the problem-solving intelligence indicator is computed for each problem used for the intelligence evaluation.

Algorithm 1 MPSI measurement

Measuring the Problem-Solving Intelligence Algorithm

IN:

//Given the studied ICM_k that have a specific architecture and a specific number of agents.

$ICM_k = \{Agent_1, Agent_2, \dots, Agent_q\}$;

OUT: $|Prob_k|$; $Prob_k = \{Prob_{k,1}, \dots, Prob_{k,m}\}$;

$Intellig_k = \{Intellig_{k,1}, Intellig_{k,2}, \dots, Intellig_{k,m}\}$;

Begin

Step 1. *He* establishes $|Prob_k|$ and $Prob_k$.

$|Prob_k| = m$; $Prob_k = \{Prob_{k,1}, \dots, Prob_{k,m}\}$

Step 2. *Obtaining the intelligence indicators data.*

//If there are more intelligence components, than each evaluated problem-solving intelligence is calculated according to (1).

@Performing the problem-solving intelligence evaluations and calculus;

$Intellig_k = \{Intellig_{k,1}, Intellig_{k,2}, \dots, Intellig_{k,m}\}$;

EndMPSIAlgorithm

$|ICM| = z$ represents the cardinality of the set of studied ICMASs. z sample intelligence indicator datasets are obtained, as the number of studied ICMASs is z . $Intellig = \{Intellig_1, Intellig_2, \dots, Intellig_z\}$ denotes the problem-solving intelligence indicators of the studied ICMASs. $Intellig_1$ represents the problem-solving intelligence obtained by evaluating ICM_1 . $Intellig_2$ represents the problem-solving intelligence obtained by evaluating ICM_2 $Intellig_z$ represents the problem-solving intelligence obtained by evaluating ICM_z . The *MPSI* algorithm is applied in order to obtain the intelligence indicators for each ICMAS.

Robust Measurement of the Intelligence Quotient (MMI) algorithm calculates the machine intelligence quotients of $ICM_1, ICM_2, \dots, ICM_z$. It describes the calculation of machine intelligence quotients, $MIQS = \{MIQ_1, MIQ_2, \dots, MIQ_z\}$ based on the provided intelligence indicator data obtained for $ICM_1, ICM_2, \dots, ICM_z$ by applying the *MPSI* algorithm. MIQ_k denotes the machine intelligence quotient of the ICM_k , which indicates its central intelligence tendency. It should be noted that the most suitable indicator of the central intelligence tendency of an *ICMAS* is the median in cases where the intelligence indicator data does not pass the normality assumption, and the mean in where the data passes the normality assumption. The previous assertion is based on the fact that a very high or a very low value influences the median less than the mean. *TypeMIQ*, used in the algorithm, is a variable that indicates the calculus of *MIQS* as mean or median.

For the verification of the data normality assumption of the intelligence indicator, a variety of goodness-of-fit statistical normality tests are available. Among the most frequently used tests, the following must be mentioned [47]: Shapiro-Wilk test (*SW test*) and Lilliefors test (*Lill test*). In the *MMI* algorithm (Algorithm 2), in order to check the normality of the intelligence indicators, the *Lill test* [47–49] and the *SW test* [47,50] were chosen. In the scientific literature, the usage of the *SW test* is recommended even in cases with smaller sample sizes. Razali and Wah [47] performed a study that proved that the *SW test* had the highest power for a given significance when compared to other very frequently used tests, including the Anderson Darling test (*AD test*), the One-sample Kolmogorov-Smirnov test (*KS test*), the *Lill test*, and the *SW test*. Stephens [51] proved that the *SW test* was more powerful for testing normality than the *KS test* and the *AD test*. In the present study, limitations in cases with many identical values were observed to be a disadvantage of the *SW test*.

Algorithm 2 MMI measurement method

Robust Measurement of the Intelligence Quotient Algorithm

IN: $Intellig = \{Intellig_1, Intellig_2, \dots, Intellig_z\}$; //Intelligence indicators sample of the *ICM*.

OUT: *TypeMIQ*; *MIQS*;

Begin

Step 1. *Calculus of the MIQS.*

TypeMIQ: = "Mean";

For ($i = 1$ to z) **Do**

 @Verify if $Intellig_i$ passes the normality assumption.

If ($Intellig_i$ failed to passes the normality assumption) **Then**

TypeMIQ: = "Median"; **Exit For**;

EndIf

EndFor

//He can change *TypeMIQ* based on the problem and domain-specific knowledge.

@He makes the final decision on the *TypeMIQ* calculus as the *mean* or the *median*;

For ($i = 1$ to z) **Do**

If (*TypeMIQ* = "Mean") **then** MIQ_i : = *MEAN*($Intellig_i$);

Else MIQ_i : = *MEDIAN*($Intellig_i$);

EndIf

EndFor

EndMMImeasuring

Visual appreciation is frequently a useful alternative to performing a numerical analysis. One such frequently applied approach is the Quantile-Quantile Plot (QQ plot). The QQ plot [52] is a scatterplot that is appropriate for the visual appreciation of normality. Technically, a QQ plot is created by plotting two sets of quantiles against one another. Visually, it should be interpreted as follows: if both sets of quantiles belong to the same distribution, then the plotted points form a line that is roughly straight. The joint use of the QQ plot with the SW test could be suggested to be a good practice in order to answer the requirement for precise verification of the normality.

There are z sample intelligence indicator datasets, because the number of studied ICMASs is z . All sample intelligence data, $Intellig_1, Intellig_2, \dots, Intellig_z$, are initially verified as having passed the normality assumption. According to the algorithm, if one or more ICMASs do not pass the normality assumption, then the automatic decision will be to calculate the *MIQ* of all ICMASs based on the intelligence indicator medians ($TypeMIQ = "Median"$). In the contrary case, the automatic decision will be to compute the *MIQ* of all the ICMASs based on the mean ($TypeMIQ = "Mean"$). He has the right to make final modifications when choosing the mean or the median if she/he is in possession of domain-specific knowledge.

The *ExtrIntDetect* algorithm (Algorithm 3) describes the proposed identification of ICMASs with extreme intelligence from among the set *ICM* of the studied ICMASs. Figure 1 presents the main processing and analysis steps comprising the *ExtrIntDetect* method. The *Detection of ICMASs with Extreme Intelligence* algorithm presents the performed processing in depth.

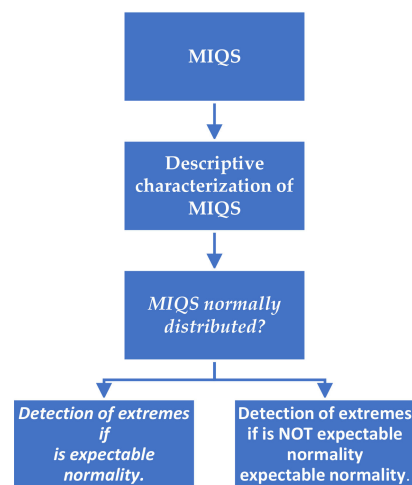


Figure 1. The main processing and analysis steps performed by *ExtrIntDetect*.

The notation “@” used in the algorithm indicates a method that is launched by the algorithm, but that is not described explicitly and in-depth in the algorithm. Step 1 of the *ExtrIntDetect* method makes a statistical characterization of the $MIQS = \{MIQ_1, MIQ_2, \dots, MIQ_z\}$ data. The variable *mean* represents the calculated mean of *MIQS*. The variable *median* represents the calculated median of *MIQS*. The median is the middle of a sorted list of numbers. If the list has an even number of items, then the median is calculated as the average of the two numbers in the middle. The value *Standard deviation (SD)* expresses a quantity by the degree to which the members of a dataset differ from the mean of that dataset. *SD* effectively quantifies the amount of variation/dispersion in a dataset [53]. The *standard error (SE)* of a specific parameter is effectively the standard deviation of its sampling distribution. When the parameter or the statistic is the mean, then it is referred to as the *standard error of the mean (SEM)*. $SEM = SD/\sqrt{N}$, where *sqrt* denotes the square root. The variable *min* denotes the lowest value. The variable *max* denotes the highest value. The variable $range = max - min$ is frequently an important indicator.

Algorithm 3 ExtrIntDetect method of measurement and detection**Detection of ICMASSs with Extreme Intelligence Algorithm**

IN: $MIQS = \{MIQ_1, MIQ_2, \dots, MIQ_z\}$; //Calculated machine intelligence quotients.

OUT:

// $IdOut^L, IdOut^H$, if normality expected.
 $IdOut^L$; //MIQ of ICMASSs with low extreme intelligence.
 $IdOut^H$; //MIQ of ICMASSs with high extreme intelligence.
// $HExtr, HExtrOut$, if normality NOT expected.
 $HExtr$; //MIQ of ICMASSs with high extreme intelligence.
 $HExtrOut$; //MIQ of ICMASSs with high outlier extreme intelligence.
 $LExtr$; //MIQ of ICMASSs with low extreme intelligence.
 $HExtrOut$; //MIQ of ICMASSs with low outlier extreme intelligence.

Begin

Step 1. Descriptive characterization of the $MIQS$ data.

@Calculates for $MIQS$ the: *mean, median, SD, SEM, skew, and kurt*.

$CV = 100 \times (SD/mean)$;

Step 2. Analyze if $MIQS$ data passes the normality assumption.

@Apply *Lill* and *SW* tests in order to verify $MIQS$ normality.

@Constructs the *QQ plot* for visual analysis of $MIQS$.

@Based on the obtained result of data normality analysis set the value of *norm* variable;

If (*norm* = "No") **Then**

@He has the right to change *norm* value to "Yes".

EndIf

Step 3. Detection of extremes if normality expected.

If (*norm* = "Yes") **Then**

$IdOut^L := \emptyset; IdOut^H := \emptyset;$

$OutFound := "Yes"; IdOut := MIQS;$

$avg^* := mean(IdOut);$

While (($OutFound = "Yes"$) **and** ($|IdOut| \geq 3$)) **Do**

$OutFound := "No";$ @It is applied the *Grubbs* test;

If (an outlier denoted *OUT* was identified) **Then**

$IdOut := IdOut \setminus \{OUT\}; OutFound := "Yes";$

If ($OUT < avg^*$) **Then** $IdOut^L := IdOut^L \cup \{OUT\};$

Else $IdOut^H := IdOut^H \cup \{OUT\};$

EndIf

EndIf

$avg^* := mean(IdOut);$

EndWhile

EndIf

If ($|IdOut^L| \geq 2$) **Then** @Order ascending $IdOut^L$. **EndIf**

If ($|IdOut^H| \geq 2$) **Then** @Order ascending $IdOut^H$. **EndIf**

Step 4. Detection of extremes if normality is NOT expected.

If (*Norm* = "No") **Then**

$HExtr := \emptyset; HExtrOut := \emptyset; LExtr := \emptyset; LExtrOut := \emptyset;$

@Arrange $MIQS = \{MIQ_1, MIQ_2, \dots, MIQ_z\}$ in ascending order.

@Calculate Q_1 and Q_3 ; //The first and third quartile.

$IQR = Q_3 - Q_1$; //Calculates the interquartile range.

@He establishes *RangL* and *RangU* values.

$HLowLim = Q_3 + RangL \times IQR; HUppLim = Q_3 + RangU \times IQR;$

$LLowLim = Q_1 - RangU \times IQR; LUppLim = Q_1 - RangL \times IQR;$

For ($i = 1$ to z) **Do**

If ($MIQ_i \in [HLowLim, HUppLim]$) **Then** $HExtr := HExtr \cup \{MIQ_i\};$

```

Elseif ( $MIQ_i \geq HUppLim$ ) Then  $HExtrOut := HExtrOut \cup \{MIQ_i\}$ ;
EndIf
If ( $MIQ_i \in (LLowLim, LUppLim)$ ) Then  $LExtr := LExtr \cup \{MIQ_i\}$ ;
    Elseif ( $MIQ_i \leq LLowLim$ ) Then  $LExtrOut := LExtrOut \cup \{MIQ_i\}$ ;
EndIf
EndFor
If ( $|HExtr| \geq 2$ ) Then @Order ascending  $HExtr$ . EndIf
If ( $|HExtrOut| \geq 2$ ) Then @Order ascending  $HExtrOut$ . EndIf
If ( $|LExtr| \geq 2$ ) Then @Order ascending  $LExtr$ . EndIf
If ( $|LExtrOut| \geq 2$ ) Then @Order ascending  $LExtrOut$ . EndIf
End
EndExtrIntDetectAlgorithm

```

Skewness (skew) [54] is a measure of the symmetry of MIQS data. A distribution is called symmetric if it looks the same on both the right and left of the center point. If $skew \geq 1$ or $skew \leq -1$, the distribution is called highly skewed. If $skew \in [0.5, 1)$ or $skew \in (-1, -0.5]$, the distribution is called moderately skewed. If $skew \in (-0.5, 0.5)$, the distribution is called approximately symmetric.

Kurtosis (kurt) [54] is a measure of whether the MIQS data are heavy-tailed or light-tailed relative to the base normal distribution. Datasets with high kurtosis have heavy tails that could include extreme values. Datasets with low kurtosis tend to have light tails, and do not include extremes. A distribution with zero excess kurtosis is called mesokurtotic. A distribution with positive excess kurtosis is called leptokurtic. A distribution with negative excess kurtosis is called platykurtotic.

The *coefficient of variation (CV)*, $CV = (SD/mean) \times 100$ characterizes the homogeneity–heterogeneity of MIQS. It was viewed that the values of CV_a , CV_b , and CV_c should be set based on the specifics of the studied ICMASs. The dataset exhibits homogeneous variability of intelligence level (*hom.*) when $CV < CV_a$; relatively homogeneous variability of intelligence level (*rel-hom.*) when $CV \in [CV_a, CV_b)$; relatively heterogeneous variability of intelligence level (*rel-het.*) when $CV \in [CV_b, CV_c)$; and heterogeneous variability of intelligence level (*het.*) when $CV \geq CV_c$. In most cases, the recommended values for the parameters described above are $CV_a = 10$, $CV_b = 20$, and $CV_c = 30$.

The detection of statistical outliers/extremes is an important subject in statistical data analysis. The detection of outliers/extremes and anomalies has diverse real-life applications, including: detection for plant-wide processes with outlier multisampling rates [55], industrial data stream anomaly detection [56], and sliding nest window control chart in the task of data stream anomaly detection [57].

In experimental studies, outlier values can sometimes influence the numerical results, and even the conclusions derived from them, to a high degree. Many tests for the detection of statistical outlier values have been presented in the scientific literature. These include: Peirce's criterion [58], the ROUT test [59], the Tietjen-Moore test [60], Chauvenet's criterion [61], Dixon's Q test [62] and the Grubbs test [63–66], just to mention a few.

Step 3 of the algorithm is applied for the recognition of statistical extremes if it is expected that the data will pass the normality assumption. The presented calculus allows the detection of $z-3$ ICMASs at most with extreme (low or high) intelligence, or an intelligence that is not extreme but is further removed from the rest, from a set of z studied ICMASs.

Step 3 describes a method for the detection of ICMASs with extreme intelligence from among the studied set *ICM*. MIQ values that are significantly/statistically different from the others are considered to be extremes. $IdOut^L$ is the set of identified ICMASs with low extreme intelligence values. $IdOut^H$ is the set of identified ICMASs with high extreme intelligence values. Depending on the type of studied ICMASs, it could be decided to extend the detection to ICMASs whose intelligence may not be extreme, but which are further removed from the rest. The Grubbs outliers detection test [63,64] is used for the identification of extremes. It is suggested that this test be applied with a significance level $\alpha_{Ext} = 0.05$. This value means that a 5% chance of mistakenly identifying an outlier in the sample is accepted. Other

values of significance level could be set, but they are less frequently appropriate. A two-sided test was applied in order to be able to identify both low and high extremes.

This method is able to detect ICMASs with extreme intelligence one-by-one, removing them when the number of remaining ICMASs is greater than or equal to 3. At the first verification, it is able to detect the system (only one system) with the most extreme (low or high) intelligence, if such a system exists. If such an ICMAS is detected, the verification can then be applied once more, to detect the existence of the system with the second most extreme intelligence. For each new application of the method, the ICMASs previously identified as having extreme intelligence are no longer considered. This process is repeated iteratively until no other extremes are identified, or until the condition (≥ 3) for the number of studied intelligent systems cannot be verified. This method is even capable of detecting systems with intelligence that is not extreme, but which can be considered to be statistically further removed (different) from the rest.

The Tietjen-Moore test [60] proposed in the scientific literature is a generalization of Grubbs' test. The Tietjen-Moore test can be applied when the number of outliers is known in advance. In the case of a single outlier, the Tietjen-Moore test will give the same results as the Grubbs' test.

Step 4 of the algorithm describes the process for detecting extremes in cases where normality has not been proven. *HE_{extr}* denotes the set of identified high extreme intelligence values. *HE_{extrOut}* denotes the set of identified high outlier extreme intelligence values. *LE_{extr}* denotes the set of identified low extreme intelligence values. *LE_{extrOut}* denotes the set of identified low outlier extreme intelligence values.

According to Step 4 of the algorithm, the *MIQS* dataset is initially arranged in ascending order. Q_1 denotes the first quartile, which represents the median of the lower half of the dataset. About 25% of the elements in the dataset will be positioned below Q_1 . Q_3 denotes the third quartile, the median of the upper half of the dataset. About 25% of the numbers will be positioned above Q_3 . *Range_L* and *Range_U* are parameters whose values must be established by *He* based on the specific characteristics of the studied set of ICMASs.

Higher machine intelligence quotient values mean higher intelligence. If a system's intelligence falls within [*H_{LowLim}*, *H_{UppLim}*), that ICMAS can be considered to have high extreme intelligence. If a system's intelligence is higher than *H_{UppLim}*, the ICMAS can be considered to have high outlier extreme intelligence. *HE_{extr}* and *HE_{extrOut}* could even be the empty set. In the final step, if *HE_{extr}* or *HE_{extrOut}* include at least 2 values, then they will be ordered in ascending order.

If the intelligence of an ICMAS falls within [*L_{LowLim}*, *L_{UppLim}*], that ICMAS can be considered to have low extreme intelligence. If a system's intelligence is lower than or equal to *L_{LowLim}*, that ICMAS can be considered to have low outlier extreme intelligence. *LE_{extr}* and *LE_{extrOut}* could even be the empty set. In the final step, if *LE_{extr}* or *LE_{extrOut}* include at least 2 values, then they will be ordered in ascending order.

4. Experimental Evaluation of the ExtrIntDetect Method

In this section, the results of the experimental evaluation of the *ExtrIntDetect* method are presented. A set of ICMASs denoted $ICM = \{ICM_1, ICM_2, ICM_3, ICM_4, ICM_5, ICM_6\}$, specialized in solving an NP-hard problem, the *Symmetric Travelling Salesman Problem (sTSP)* [67,68], is considered. The *sTSP* can be defined by considering a set of cities (nodes of a weighted undirected graph), with the distances between each pair of cities being the same in both directions. The aim is to find the shortest route that visits each city (node) exactly once and then returns to the origin city. A traditional formulation of the *sTSP* is as an integer linear programming problem [69]. Diverse real-life problems, including vehicle routing, warehouse logistics, circuit board planning, and virtual networking, are modelled as symmetric graphs.

Each studied ICMAS consisted of simple computing agents (artificial ants) that operated by mimicking the search for food of biological ants. Many CMASs mimic the problem-solving that is carried out by biological ants: Merkle and Middendorf [70] proposed a solution to the permutation scheduling problem; Runkler [71] proposed the optimization of clustering models; Crisan, Pintea and

Palade [72] proposed emergency management using geographic information systems; and Pholdee and Bureerat [73] proposed the optimization of the constrained mechanical design.

The general operation of the ICMASs presented in [74–76] can be explained as follows. Initially, each agent is placed on a randomly chosen node of the graph. An $agent_k$ currently at node i chooses to move to node j by applying the probabilistic transition rule (2). The parameter α denotes the power of the pheromone, and β denotes the relative weight of the heuristic visibility of the pheromone trail. The value d_{kh} represents the distance between nodes (k and h); $\eta_{kh} = 1/d_{kh}$ is the heuristic visibility of the edge (k, h); and $\tau_{kh}(t)$ represents the pheromone amount deposited on the edge (k, h) at the considered iteration t .

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta}{\sum_{l \in J_k \setminus i} [\tau_{il}(t)]^\alpha \times [\eta_{il}(t)]^\beta}, & \text{if } j \in J_k \setminus i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

After each agent completes its tour, the pheromone amount on each edge will be adjusted according to (3), (4) and (5). Q is an arbitrary constant. $0 < \rho < 1$, is the trail evaporation measure when an agent chooses a node to move to. m is the number of operating agents. L_k is the length of the tour performed by $agent_k$.

$$\tau_{ij}(t+1) = (1 - \rho) \times \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{k=m} \Delta\tau_{ij}^k(t) \quad (4)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{if } \{i, j\} \in \text{tour_performed_by_agent_}k \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

He establishes the type of intelligence sought to be composed of two components, $q = 2$, which are denoted as ms_1 and ms_2 . ms_1 is the global-best solution found during the problem-solving cycle. ms_2 is the number of iterations before the global-best solution is found. The importance of the two components is considered by He as follows, $imp_1 = 0.9$ and $imp_2 = 0.1$ ($imp_1 + imp_2 = 1$). In all cases, the intelligence indicator is calculated on the basis of (1). It should be noted that, based on the specificity of the intelligence indicator, lower values indicate higher intelligence.

ICM_1 is operated as an Ant System (AS) [74–76]; ICM_2 is operated as an Elitist Ant System (eAS) [77]; ICM_3 is operated as a Ranked Ant System (rAS) [78]; ICM_4 is operated as a Best-Worst Ant System (bwAS) [79]; ICM_5 is operated as a Min-Max Ant System (mmAS) [80]; ICM_6 is operated as an Ant Colony System (aCS) [75,76,81].

The experimentally determined parameter values were: maps with randomly placed cities on the map, $nr = 90$; number of agents that formed each studied CMAS, $|Probl_i| = 12, \forall I = 1.6$; $m = 10$; maximal number of iterations, $ItNum = 995$; $\alpha = 1.3$; $\beta = 1.35$ and $\rho = 0.25$. Table 1 presents the intelligence component results obtained in the experimental problem-solving evaluations. Table 2 presents the problem-solving intelligence evaluation results $Intellig = \{Intellig_1, Intellig_2, \dots, Intellig_6\}$ calculated according to (1) based on the two measured components, with each component counted based on the established weighting. Table 3 presents the results of the statistical characterization of intelligence realized based on $Intellig = \{Intellig_1, Intellig_2, \dots, Intellig_6\}$, which allows the formulation of useful conclusions, like the homogeneity–heterogeneity of intelligence indicator data. For the verification of homogeneity–heterogeneity, the following values for the parameters were used: $CV_a = 10$, $CV_b = 20$ and $CV_c = 30$.

Table 1. Obtained experimental evaluation results of ms_1 and ms_2 .

<i>Evaluation Number</i>	<i>ICM₁</i>	<i>ICM₂</i>	<i>ICM₃</i>	<i>ICM₄</i>	<i>ICM₅</i>	<i>ICM₆</i>
	ms_1/ms_2	ms_1/ms_2	ms_1/ms_2	ms_1/ms_2	ms_1/ms_2	ms_1/ms_2
<i>Nr₁</i>	1063/104	1050/165	1268/18	1583/22	1488/29	1093/146
<i>Nr₂</i>	1073/559	1045/115	1157/32	1709/32	1387/38	1123/226
<i>Nr₃</i>	1067/306	1094/70	1255/58	1792/22	1481/19	1184/221
<i>Nr₄</i>	1046/92	1048/131	1170/47	1645/21	1380/33	1170/68
<i>Nr₅</i>	3698/21	3906/43	7730/53	9518/18	8597/10	3628/67
<i>Nr₆</i>	8304/29	7747/64	9484/31	1381/12	1198/38	7935/53
<i>Nr₇</i>	7527/16	7384/110	8791/37	1590/27	1090/44	7860/91
<i>Nr₈</i>	8291/16	7823/93	9911/16	1423/15	1059/100	8359/87
<i>Nr₉</i>	8054/168	8930/991	1023/37	1470/17	1156/39	9123/31
<i>Nr₁₀</i>	9294/181	9598/41	1098/24	1424/28	1205/64	1092/290
<i>Nr₁₁</i>	1015/68	9427/93	1124/48	1664/43	1263/28	1023/174
<i>Nr₁₂</i>	1019/64	9434/182	1182/27	1377/22	1371/19	1024/68

Table 2. Calculated intelligence evaluation results.

<i>Evaluation Number</i>	<i>Intellig₁</i>	<i>Intellig₂</i>	<i>Intellig₃</i>	<i>Intellig₄</i>	<i>Intellig₅</i>	<i>Intellig₆</i>
<i>Nr₁</i>	967.1	961.5	1143.0	1426.9	1342.1	998.3
<i>Nr₂</i>	1021.6	952.0	1044.5	1541.3	1252.1	1033.3
<i>Nr₃</i>	990.9	991.6	1135.3	1615.0	1334.8	1087.7
<i>Nr₄</i>	950.6	956.3	1057.7	1482.6	1245.3	1059.8
<i>Nr₅</i>	3330.3	3519.7	6962.3	8568.0	7738.3	3271.9
<i>Nr₆</i>	7476.5	6978.7	8538.7	1244.1	1082.0	7146.8
<i>Nr₇</i>	6775.9	6656.6	7915.6	1433.7	985.4	7083.1
<i>Nr₈</i>	7463.5	7050.0	8921.5	1282.2	963.1	7531.8
<i>Nr₉</i>	7265.4	8136.1	924.4	1324.7	1044.3	8213.8
<i>Nr₁₀</i>	8382.7	8642.3	990.6	1284.4	1090.9	1011.8
<i>Nr₁₁</i>	920.3	8493.6	1016.4	1501.9	1139.5	938.1
<i>Nr₁₂</i>	923.5	8508.8	1066.5	1241.5	1235.8	928.4

Table 3. Characterization of the intelligence of the studied ICMASs.

<i>Calculus</i>	<i>Intellig₁</i>	<i>Intellig₂</i>	<i>Intellig₃</i>	<i>Intellig₄</i>	<i>Intellig₅</i>	<i>Intellig₆</i>
<i>Mean</i>	3872.36	5153.93	3393.04	1995.53	1704.47	3358.73
<i>SD</i>	3265.3	3377.76	3494.04	2073.47	1904.44	3132.09
<i>Median</i>	2175.95	6817.65	1100.9	1430.3	1187.65	1073.75
<i>Kurt</i>	−2.1	−1.9	−1.4	11.9	11.9	−1.6
<i>Skew</i>	0.3	−0.4	0.9	3.4	3.4	0.7
<i>Min</i>	920.3	952	924.4	1241.5	963.1	928.4
<i>Max</i>	8382.7	8642.3	8921.5	8568	7738.3	8213.8
<i>range</i>	7462.4	7690.3	7997.1	7326.5	6775.2	7285.4
<i>CV</i>	84.3	65.5	102	103.9	111.7	93.3
<i>hom.-het.</i>	het.	het.	het.	het.	het.	het.

Figures 2–7 represent the QQ plots of $Intellig_1, Intellig_2, \dots, Intellig_6$. The visual analysis of Figures 2–7 enables the formulation of the conclusion that none of $Intellig_1, Intellig_2, \dots, Intellig_6$ fail to pass the normality assumption.

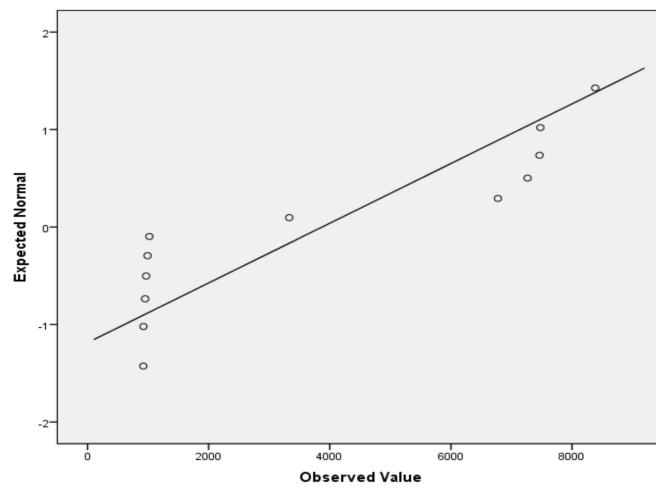


Figure 2. QQ plot of *Intellig*₁.

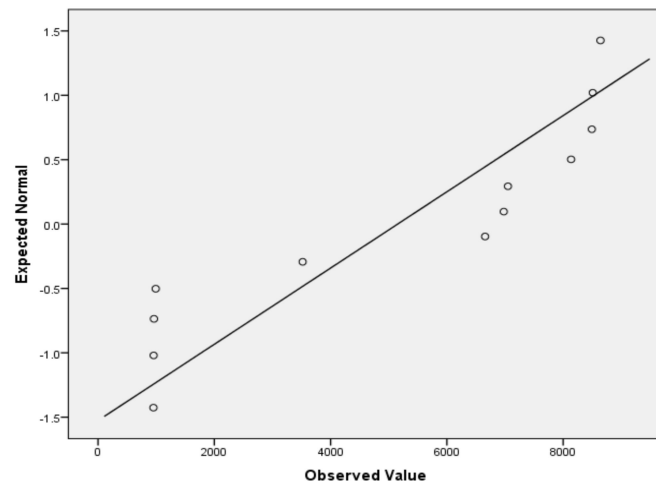


Figure 3. QQ plot of *Intellig*₂.

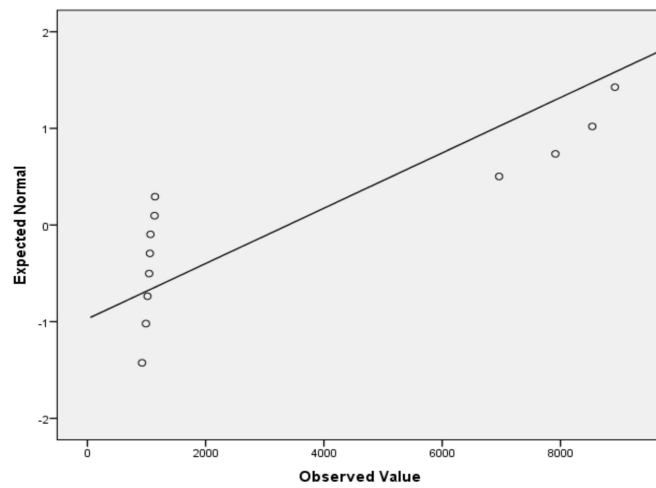


Figure 4. QQ plot of *Intellig*₃.

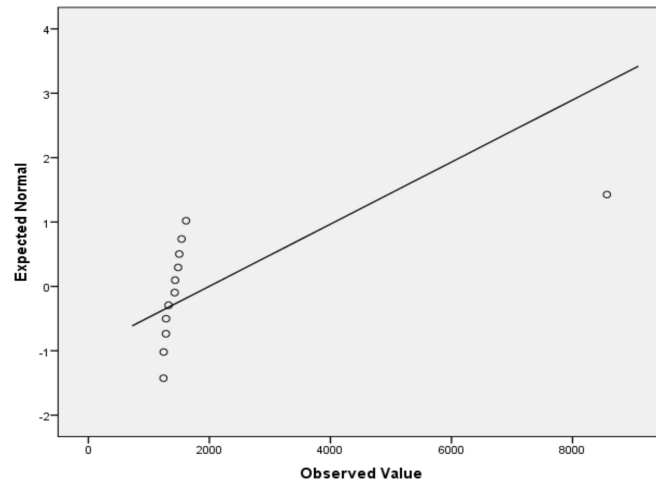


Figure 5. QQ plot of *Intellig*₄.

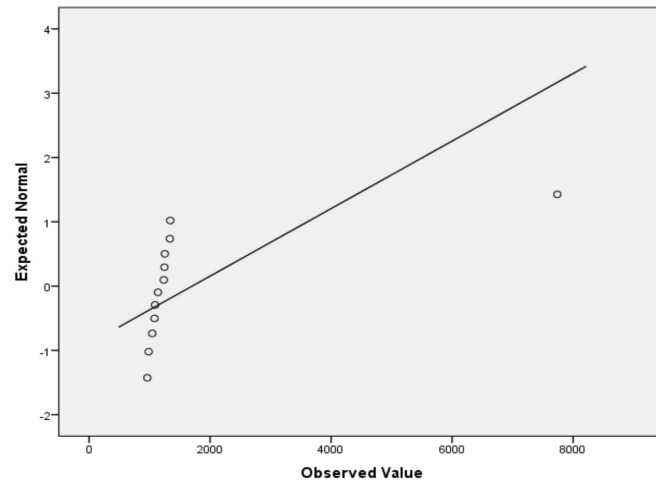


Figure 6. QQ plot of *Intellig*₅.

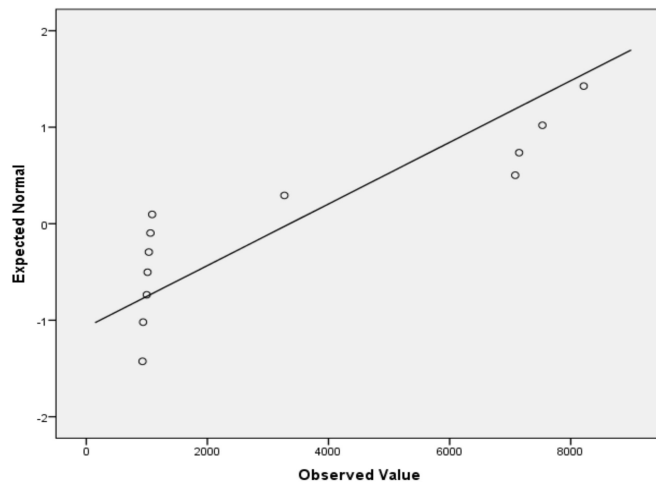


Figure 7. QQ plot of *Intellig*₆.

Table 4 presents the results of the Lill and SW normality goodness-of-fit tests applied to *Intellig*₁, *Intellig*₂, ..., *Intellig*₆. It was considered that a significance level of $\alpha_{Norm} = 0.05$ was the most appropriate for the application of both tests. An intelligence indicator dataset fails to pass the normality assumption when $P_{norm} < \alpha_{Norm}$, where P_{norm} represents the calculated p-value of an applied

normality test. Based on the obtained results, the same conclusion can be formulated with respect to normality as was formulated for the QQ plots.

Table 4. Lill and SW normality tests results.

Calculus	Intellig ₁	Intellig ₂	Intellig ₃	Intellig ₄	Intellig ₅	Intellig ₆
Lill Test						
Lill statistic	0.309	0.255	0.407	0.489	0.492	0.349
Pnorm (calculated p-value)	0.002	0.03	~0	~0	~0	~0
Normality passed (Pnorm > αNorm)	No	No	No	No	No	No
SW Test						
SW statistic	0.753	0.79	0.666	0.381	0.389	0.717
Pnorm (calculated p-value)	0.003	0.007	~0	~0	~0	0.001
Normality passed (Pnorm > αNorm)	No	No	No	No	No	No

Based on the fact that none of the intelligence indicator datasets pass the normality assumption, it can be concluded that the most appropriate choice would be to calculate the machine intelligence quotient on the basis of the median. $MIQS = \{MIQ_1 = 2175.95, MIQ_2 = 6817.65, MIQ_3 = 1100.9, MIQ_4 = 1430.3, MIQ_5 = 1187.65, MIQ_6 = 1073.75\}$. Figure 8 presents the MIQS data. Based on the specificity of the considered intelligence components and the intelligence indicator, smaller MIQ values indicate higher machine intelligence. For instance, $MIQ_2 > MIQ_6$ suggests that ICM_6 has a higher intelligence than ICM_2 .

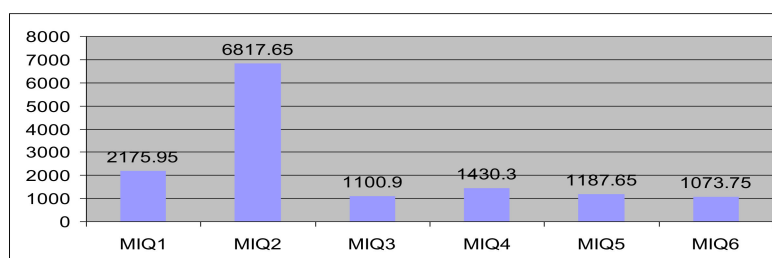


Figure 8. Graphical MIQS data (higher MIQ means lower intelligence).

Table 5 contains a descriptive statistical characterization of the MIQS. Table 6 presents the results of the Lill and SW tests applied to the MIQS. Both tests were applied with a significance level of $\alpha Norm = 0.05$. Figure 9 presents the QQ plot constructed based on the MIQS data. All the normality results indicate the failure of the MIQS data to pass the normality assumption.

Table 5. Descriptive characterization of the MIQS data.

Type of Calculus	Obtained Result
mean/SD/median	2297.7/2252.1/1308.98
Skew/interpretation	2.28/highly skewed
kurt/interpretation	5.28/leptokurtic
CV/hom-het	98/heterogeneous

Table 6. Analysis of normality of the MIQS data.

Performed Calculus	Lill Test	SW Test
Statistic	0.355	0.636
Pnorm (calculated p-value)	0.017	0.001
Normality passed (Pnorm > αNorm)	No	No

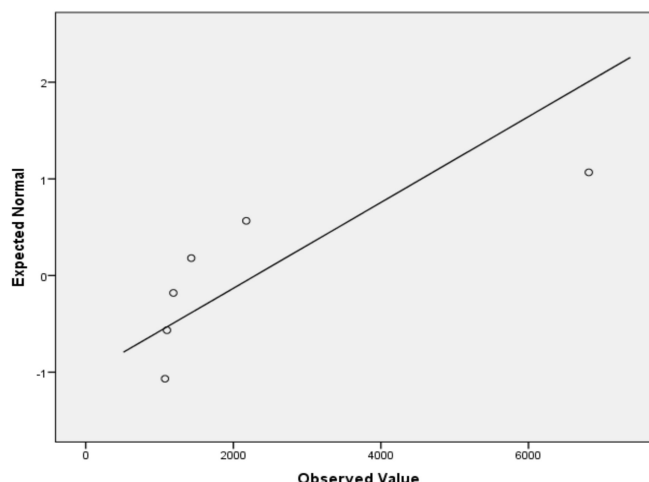


Figure 9. QQ plot of the MIQS data.

Based on the fact that the $MIQS = \{MIQ_1, MIQ_2, MIQ_3, MIQ_4, MIQ_5, MIQ_6\} = \{2175.95, 6817.65, 1100.9, 1430.3, 1187.65, 1073.75\}$ data does not pass the normality assumption, according to the *ExtrIntDetect* algorithm, Step 4 is indicated for the identification of extreme intelligence. Acting as *He*, we decided on $RangeL = 1.5$ and $RangeU = 3$. The obtained numerical results were: $median = 1308.98$, $Q_1 = 1094.11$, $Q_3 = 3336.37$, $IQR = 5743.9$, $HLowLim = 11,952.22$, $HUppLim = 20,568.07$, $LLowLim = -7521.74$, $LUppLim = -16,137.6$.

The criteria of $MIQ_i \in (LLowLim, LUppLim]$ and $MIQ_i \leq LLowLim$ were not verified. This indicates that there were no ICMASs with high extreme intelligence or high outlier extreme intelligence. Given that lower intelligence values indicate higher intelligence, this is the case for high intelligence.

The condition $MIQ_i \geq HUppLim = 20568.07$ was verified for $MIQ_1[2175.95]$ and $MIQ_2[6817.65]$. In this way, ICM_2 and ICM_1 were identified as having low outlier extreme intelligence, where ICM_2 was less intelligent than ICM_1 (based on the fact that $2175.95 < 6817.65$).

The condition of $MIQ_i \in [HLowLim, HUppLim) = [11952.22, 20568.07)$ was verified by the value corresponding to $MIQ_4[1430.3]$, identifying ICM_4 in this way as an ICMAS with low extreme intelligence.

The execution of Step 3 of the *ExtrIntDetect* algorithm is recommended when, after the elimination of extremes, it is expected that the normality assumption will be passed. It was applied for the $MIQS = \{MIQ_1, MIQ_2, MIQ_3, MIQ_4, MIQ_5, MIQ_6\}$ data. For the identification of the extremes, Grubbs' test was applied with a significance level of $\alpha_{Ext} = 0.05$. A two-sided test was applied, as both high and low extreme intelligence values were being sought at the same time. Table 7 presents the identified extremes.

Table 7. The results of the *ExtrIntDetect* method.

ICMAS	MIQ	Extreme */Furthest #	No.Id @	Type &
ICM_1	2175.95	No/Yes	2	Low
ICM_2	6817.65	Yes/No	1	Low
ICM_3	1100.9	/		
ICM_4	1430.3	No/Yes	3	Low
ICM_5	1187.65	No/Yes	4	Low
ICM_6	1073.75	/		

* indicates an ICMAS identified as having extreme intelligence; # indicates an ICMAS intelligence that is not significantly extreme, but is further removed from the intelligence of the rest of ICMASs; @ indicates the application number of the extreme detection test; & indicates the type of intelligence "Low" or "High".

$MIQS^* = \{MIQ_1, MIQ_2, MIQ_3, MIQ_4, MIQ_5, MIQ_6\}$, $avg^* = mean(MIQS^*) = 2297.7$. With the first application of the identification of extremes based on $MIQS^*$ data, ICM_2 was identified as having extreme intelligence, $MIQ_2 = 6817.65$. ICM_2 has low extreme intelligence based on the fact that

$MIQ_2 > avg^*$. $MIQS^* = MIQS^* - \{MIQ_2\}$. With the second application of the identification of extremes based on $MIQS^*$ data, no extremes were identified.

With the aim of verifying the correctness of the application of Step 3 of the algorithm (after the elimination of the extremes, a set of normally distributed data was obtained), the data normality of $MIQS^*$ was verified using the numerical Lill and SW tests. Both tests were applied at $\alpha Norm = 0.05$. The Lill test results were: *Lill statistics* = 0.273, *Pnorm* = 0.2 (p-value of the Lill test), where $Pnorm > \alpha Norm$. The SW test results were: *SW statistics* = 0.273, *Pnorm* = 0.057 (p-value of the SW test), where $Pnorm > \alpha Norm$. All of the obtained results confirmed that the normality assumption had been passed at the established $\alpha Norm$ significance level. This proves the correctness of the application of Step 3 of the algorithm, and that by eliminating the extremes, a set of normally distributed data will be obtained.

On the second application of the recognition, only an intelligence indicator value, $MIQ_1 = 2175.95$, which is far removed from the values, was detected. $avg^* = mean(MIQS^*) = 1393.71$; $MIQ_1 > avg^*$, proving that ICM_1 is lower than the rest of the intelligence values. $MIQS^* = MIQS^* - \{MIQ_1\}$, $avg^* = mean(MIQS^*) = 1198$. To identify further different intelligence values, the detection of extremes was applied again on the $MIQS^*$ data, identifying $MIQ_4 = 1430.3$ as another value that is not extreme but is far removed from the rest; $MIQ_4 > avg^*$ indicates that it was lower than the rest of the intelligence values. $avg^* = mean(MIQS^*) = 1120.77$, $MIQS^* = MIQS^* - \{MIQ_4\}$. With a new application, $MIQ_5 = 1187.65$ was identified as being far removed from the rest ($MIQ_5 > avg^*$). Based on the fact that $|MIQS^*| = 2$, which is not higher than or equal to 3, the detection of other extremes or values far removed from the rest could not be continued.

5. Discussion

5.1. Discussion of the Experimental Results

For the validation of the proposed *ExtrIntDetect* method, an illustrative experimental case study was performed. In the experimental setup, we used six ICMASs that were specialized in solving a very difficult NP-hard problem. As initial steps, the *ExtrIntDetect* algorithm performs some initial processing and analyses. According to the algorithm, based on the fact that the $MIQS$ data did not pass the normality assumption, Step 4 of the algorithm was applied.

The results of Step 4 for the set of six studied intelligent systems showed that no ICMAS had high extreme or high outlier extreme intelligence. Two ICMASs were identified, namely ICM_2 , and ICM_5 , as having low outlier extreme intelligence, and one ICMAS, namely ICM_4 , as having low extreme intelligence.

In the presented experimental study, the application of Step 3 was possible even if the $MIQS$ data initially did not pass the normality assumption. The correctness of the application is a result of the elimination of ICMASs, which leads to normally distributed machine intelligence quotient data.

Step 3 identified on ICMAS, namely ICM_2 , with low extreme intelligence, and three ICMASs, namely ICM_5 , ICM_4 and ICM_1 , with low intelligence that was far removed from the rest.

It should be noted that in Step 3, the values of the parameters *RangeL* and *RangeU* influenced which ICMASs were identified as extremes or outlier extremes. *RangeL* and *RangeU* values must be specified by the *He*. The most appropriate value in most cases for *RangeL* is 1.5, and for *RangeU*, the most appropriate value is a number between 2 and 3. One recommended method for setting these values is to vary the values of *RangeL* and *RangeU*, and to analyze the identified ICMASs as extremes and outlier extremes.

5.2. Discussion of the ExtrIntDetect Method

Several intelligence metrics have been presented in the literature, most of which are based on different principles of measuring machine intelligence. In our study, the most appropriate measurement of intelligence is based on difficult problem-solving intelligence, which represents a field of interest for human evaluators specializing in intelligent systems. In the case of each studied ICMAS, evaluations of

problem-solving intelligence were performed on several testing problem sets, and the obtained results were used to calculate the *MIQ*, which is the indicator of the central problem-solving intelligence tendency. This kind of procedure provides universality for the approaches that are based on it. It does not depend on the studied ICMAS architecture. It also does not require the elaboration of definitions of intelligence.

In [30], a method for the identification of ICMASs with extreme intelligence called *OutIntSys* was presented. One limitation of that method was the fact that its application is based on the assumption that by eliminating the extreme intelligence values, normally distributed intelligence data will be obtained. *ExtrIntDetect* preserves the properties of the *OutIntSys* method, while extending it at the same time in order to be more robust. It is applicable even in situations where data normality is not expected. *ExtrIntDetect* is able to detect extreme (low and high) intelligence and outlier extreme (low and high) intelligence.

The recent symmetric *MetrIntSimil* metric [31] is able to compare the intelligence of a large number of ICMASs. *MetrIntSimil* is based on the same principle of considering the difficult problem-solving machine intelligence as the *ExtrIntDetect* method. *MetrIntSimil* is able to perform accurate classification of ICMASs that have the same intelligence level, but is not able to detect systems with extreme intelligence. *ExtrIntDetect* could be used as an extension of the *MetrIntSimil* metric. First, the *MetrIntSimil* metric should be applied for a set of ICMASs, measuring their machine intelligence, comparing their intelligence, and classifying them into intelligence classes. After that, the application of the *ExtrIntDetect* method will be able to detect ICMASs with extreme intelligence.

In [82], an interesting study was presented regarding the effect of motion on the spread of disease in a system composed of random walkers. The presented random walkers could perform long-distance jumps. The main conclusion of the study was that a reduced percentage of jumps in agent motion was sufficient to annihilate local correlations. It also produced a large drop in the epidemic threshold. The studied effect is to a certain degree similar to the crossover existent in the well-known static small-world networks. Furthermore, this can be linked to the structural properties of the dynamic network that defines the interactions between agents. A forthcoming study that we will perform will consist of defining intelligence in networks, the determination of a mathematically grounded method for measuring such intelligence, and finally the identification of extreme (low and high) intelligence.

6. Conclusions

In the research presented in this paper, we defined a specific type of performance of ICMASs in terms of difficult problem-solving intelligence. In this context, the problem of performance measurement was formulated as a problem of the measurement of intelligence. The motivation for the study consists in the fact that an ICMAS with higher intelligence can solve more difficult problems than a system with lower intelligence.

The *ExtrIntDetect* method proposed in this paper has practical application in identifying ICMASs with extreme (low and high) or outlier extreme (low and high) intelligence for solving difficult problems from among a set of studied ICMASs. For the evaluation and validation of the *ExtrIntDetect* method, an experimental study was performed, in which the intelligence of six ICMASs was analyzed.

ExtrIntDetect extends another recent method called *OutIntSys*, which was proposed to identify intelligent systems that had extreme intelligence, retaining its properties, while being more robust. The *MetrIntSimil* metric proposed in the literature is able, in cases of a larger number of studied intelligent systems, to measure their problem-solving intelligence, compare the systems based on their intelligence, and classify the systems into classes of intelligence. Its main limitation is the fact that it is not able to detect extreme intelligence. *ExtrIntDetect* can be used in addition after the application of the *MetrIntSimil* metric in order to detect extreme intelligence.

Although the method was proposed for ICMAS, it could be applied for all intelligent systems, for instance, to agents that need to solve problems in isolation without cooperation with other agents. The

method is universal, it is like the black-box method, in that it does not depend on the architecture of the studied ICMASs.

Examples of possible applications include: the identification of swarms of mobile robotic agents with extreme intelligence from a large set of swarms; identification of swarms of agent-based flying drones with extreme intelligence from a large set of swarms.

Author Contributions: L.B.I., R.K. and G.C.C. designed the proposed intelligence metric; L.B.I. and G.C.C. conceived, designed and performed the experiments; L.B.I. and G.C.C. analyzed the data and contributed analysis tools; L.B.I., R.K. and G.C.C. wrote the paper.

Funding: This work is a part of the CHIST-ERA programme supported by the Future and Emerging Technologies (FET) programme of the European Union through the ERA-NET funding scheme under the grant agreements, title *Social Network of Machines (SOON)*. This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CCCDI-UEFISCDI, project number 101/2019, COFUND-CHIST-ERA-SOON, within PNCDI III.

Acknowledgments: This work is a part of the CHIST-ERA programme supported by the Future and Emerging Technologies (FET) programme of the European Union through the ERA-NET funding scheme under the grant agreements, title *Social Network of Machines (SOON)*. This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CCCDI-UEFISCDI, project number 101/2019, COFUND-CHIST-ERA-SOON, within PNCDI III.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Gelenbe, E.; Lent, R.; Xu, Z. Design and performance of cognitive packet networks. *Perform. Eval.* **2001**, *46*, 155–176. [[CrossRef](#)]
- Sakellari, G. Performance evaluation of the Cognitive Packet Network in the presence of network worms. *Perform. Eval.* **2011**, *68*, 927–937. [[CrossRef](#)]
- Spoto, S.; Gribaudo, M.; Manini, D. Performance evaluation of peering-agreements among autonomous systems subject to peer-to-peer traffic. *Perform. Eval.* **2014**, *77*, 1–20. [[CrossRef](#)]
- Crovella, M.; Lindemann, C.; Reiser, M. Internet performance modeling: The state of the art at the turn of the century. *Perform. Eval.* **2000**, *42*, 91–108. [[CrossRef](#)]
- Al-Rousan, M.; Archibald, J.K.; Bearnson, L. Evaluating the impact of locality on the performance of large-scale SCI multiprocessors. *Perform. Eval.* **2001**, *46*, 275–302. [[CrossRef](#)]
- Liu, L.; Lim, S. A voxel-based multiscale morphological airborne lidar filtering algorithm for digital elevation models for forest regions. *Measurement* **2018**, *123*, 135–144. [[CrossRef](#)]
- Wibowo, S.; Grandhi, S. Fuzzy Multicriteria Analysis for Performance Evaluation of Internet-of-Things-Based Supply Chains. *Symmetry* **2018**, *10*, 603. [[CrossRef](#)]
- Karimi, N.; Kondrood, R.R.; Alizadeh, T. An intelligent system for quality measurement of Golden Bleached raisins using two comparative machine learning algorithms. *Measurement* **2017**, *107*, 68–76. [[CrossRef](#)]
- Sen, B.; Mandal, U.K.; Mondal, S.P. Advancement of an intelligent system based on ANFIS for predicting machining performance parameters of Inconel 690—A perspective of metaheuristic approach. *Measurement* **2017**, *109*, 9–17. [[CrossRef](#)]
- Ahmadi, F.F.; Layegh, N.F. Integration of close range photogrammetry and expert system capabilities in order to design and implement optical image based measurement systems for intelligent diagnosing disease. *Measurement* **2014**, *51*, 9–17. [[CrossRef](#)]
- Sobolev, V.; Aumala, O. Metrological automatic support of measurement results in intelligent measurement systems. *Measurement* **1996**, *17*, 151–159. [[CrossRef](#)]
- Liu, W. Intelligent fault diagnosis of wind turbines using multi-dimensional kernel domain spectrum technique. *Measurement* **2019**, *133*, 303–309. [[CrossRef](#)]
- Popescu, D.; Ichim, L. Intelligent Image Processing System for Detection and Segmentation of Regions of Interest in Retinal Images. *Symmetry* **2018**, *10*, 73. [[CrossRef](#)]
- Wang, D.; Ren, H.; Shao, F. Distributed Newton Methods for Strictly Convex Consensus Optimization Problems in Multi-Agent Networks. *Symmetry* **2017**, *9*, 163. [[CrossRef](#)]
- Iantovics, L.B.; Zamfirescu, C.B. ERMS: An evolutionary reorganizing multiagent system. *Innov. Comput. Inf. Control* **2013**, *9*, 1171–1188.

16. Kwon, H.; Pack, D.J. A Robust Mobile Target Localization Method for Cooperative Unmanned Aerial Vehicles Using Sensor Fusion Quality. *J. Intell. Robot. Syst.* **2012**, *65*, 479–493. [[CrossRef](#)]
17. Saska, M.; Vonasek, V.; Krajník, T.; Preucil, L. Coordination and Navigation of Heterogeneous MAV-UGV Formations Localized by a 'hawk-eye'-like Approach Under a Model Predictive Control Scheme. *Int. J. Robot. Res.* **2014**, *33*, 1393–1412. [[CrossRef](#)]
18. Chase, D. *Underlying Factor Structures of the Stanford-Binet Intelligence Scales*, 5th ed.; Drexel University: Philadelphia, PA, USA, 2005.
19. Kaufman, A.S. *IQ Testing 101*; Springer: New York, NY, USA, 2009.
20. Nicolas, S.; Andrieu, B.; Croizet, J.C.; Sanitioso, R.B.; Burman, J.T. Sick? Or slow? On the origins of intelligence as a psychological object. *Intelligence* **2013**, *41*, 699–711. [[CrossRef](#)]
21. Bilker, W.B.; Hansen, J.A.; Brensinger, C.M.; Richard, J.; Gur, R.E.; Gur, R.C. Development of abbreviated nine-item forms of the Raven's standard progressive matrices test. *Assessment* **2012**, *19*, 354–369. [[CrossRef](#)]
22. Raven, J.C. Mental Tests used in Genetic studies: The performance of Related Individuals on Tests Mainly Educative and Mainly Reproductive. MSc Thesis, University of London, London, UK, 1936.
23. Kaufman, A.S.; Lichtenberger, E. *Assessing Adolescent and Adult Intelligence*, 3rd ed.; Wiley: Hoboken, NJ, USA, 2006; p. 3.
24. Wechsler, D. *The Measurement of Adult Intelligence*. Baltimore (MD); Williams & Wilkins: Philadelphia, PA, USA, 1939; p. 229.
25. Kaufman, A.S.; Kaufman, N.L. *Kaufman test of Educational Achievement Comprehensive Form*, 2nd ed.; Circle Pines, N., Ed.; American Guidance Service: Circle Pines, MN, USA, 2004.
26. Kaufman, A.S.; Kaufman, N.L. *Kaufman Assessment Battery for Children*; American Guidance Service: Circle Pines, MN, USA, 1983.
27. Kaufman, A.S.; Kaufman, N.L. *Kaufman Assessment Battery for Children*, 2nd ed.; American Guidance Service: Circle Pines, MN, USA, 2004.
28. Neisser, U.; Boodoo, G.; Bouchard, T.J.J.; Boykin, A.W.; Brody, N.; Ceci, S.J.; Halpern, D.F.; Loehlin, J.C.; Perloff, R.; Sternberg, R.J.; et al. Intelligence: Knowns and unknowns. *Am. Psychol.* **1996**, *51*, 77–101. [[CrossRef](#)]
29. Schmidt, F.L.; Hunter, J. General mental ability in the world of work: Occupational attainment and job performance. *J. Pers. Soc. Psychol.* **2004**, *86*, 162–173. [[CrossRef](#)] [[PubMed](#)]
30. Arik, S.; Iantovics, L.B.; Szilagy, S.M. OutIntSys—A Novel Method for the Detection of the Most Intelligent Cooperative Multiagent Systems. In Proceedings of the 24th International Conference on Neural Information Processing (ICONIP 2017), Guangzhou, China, 14–18 November 2017; pp. 31–40, LNCS 10637.
31. Iantovics, L.B.; Dehmer, M.; Emmert-Streib, F. MetrIntSimil—An Accurate and Robust Metric for Comparison of Similarity in Intelligence of Any Number of Cooperative Multiagent Systems. *Symmetry* **2018**, *10*, 48. [[CrossRef](#)]
32. Iantovics, L.B.; Gligor, A.; Niazi, A.M.; Biro, A.I.; Szilagy, S.M.; Tokody, D. Review of Recent Trends in Measuring the Computing Systems Intelligence. *BRAIN Broad Res. Artif. Intell. Neurosci.* **2018**, *9*, 77–94.
33. Turing, A.M. Computing machinery and intelligence. Oxford University Press on behalf of the Mind Association, Mind. *New Ser.* **1950**, *59*, 433–460. [[CrossRef](#)]
34. Dowe, D.L.; Hajek, A.R. A non-behavioural, computational extension to the Turing Test. In *Proceedings of the International Conference on Computational Intelligence & Multimedia Applications (ICCIMA 1998)*, 7–10 February 1998, Gippsland, Australia; World Scientific Publishing: Singapore, 1998; pp. 101–106.
35. Schreiner, K. Measuring IS: Toward a US standard. *IEEE Intell. Syst. Appl.* **2000**, *15*, 19–21. [[CrossRef](#)]
36. Park, H.J.; Kim, B.K.; Lim, K.Y. Measuring the machine intelligence quotient (MIQ) of human-machine cooperative systems. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2001**, *31*, 89–96. [[CrossRef](#)]
37. Sanghi, P.; Dowe, D.L. A Computer Program Capable of Passing I.Q. Tests. In Proceedings of the Joint International Conference on Cognitive Science, 4th ICCS International Conference on Cognitive Science and 7th ASCS Australasian Society for Cognitive Science (ICCS/ASCS 2003), Sydney, NSW, Australia, 13–17 July 2003; pp. 570–575.
38. Legg, S.; Hutter, M. A formal measure of machine intelligence. In Proceedings of the 15th Annual Machine Learning Conference, Ghent, Belgium, 11–12 May 2006; pp. 73–80.

39. Anthony, A.; Jannett, T.C. *Measuring Machine Intelligence of an Agent-Based Distributed Sensor Network System; Advances and Innovations in Systems*; Elleithy, K., Ed.; Springer: Dordrecht, The Netherlands, 2007; pp. 531–535.
40. Hernández-Orallo, J. Beyond the Turing Test. *J. Logic. Lang. Inf.* **2000**, *9*, 447–466. [[CrossRef](#)]
41. Hernández-Orallo, J.; Dowe, D.L. Measuring universal intelligence: Towards an anytime intelligence test. *Artif. Intell.* **2010**, *174*, 1508–1539. [[CrossRef](#)]
42. Hibbard, B. Measuring Agent Intelligence Via Hierarchies of Environments. In *Artificial General Intelligence, Lecture Notes in Computer Science 6830*; Schmidhuber, J., Thórisson, K.R., Looks, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 303–308.
43. Hernández-Orallo, J.; Dowe, D.L.; Hernández-Lloreda, M.V. Universal psychometrics: Measuring cognitive abilities in the machine kingdom. *Cogn. Syst. Res.* **2014**, *27*, 50–74. [[CrossRef](#)]
44. Besold, T.; Hernandez-Orallo, J.; Schmid, U. Can machine intelligence be measured in the same way as human intelligence? *Künstliche Intelligenz* **2015**, *29*, 291–297. [[CrossRef](#)]
45. Chmait, N.; Li, Y.F.; Dowe, D.L.; Green, D.G. A Dynamic Intelligence Test Framework for Evaluating AI Agents. In *Proceedings of the Workshop Evaluating General-Purpose AI, The Hague, The Netherlands, 30 August 2016*; pp. 1–8, EGPAI 2016.
46. Liu, F.; Shi, Y.; Liu, Y. Intelligence quotient and intelligence grade of artificial intelligence. *Ann. Data Sci.* **2017**, *4*, 179–191. [[CrossRef](#)]
47. Razali, N.; Wah, Y.B. Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors, Anderson-Darling tests. *J. Stat. Model. Anal.* **2011**, *2*, 2–33.
48. Lilliefors, H. On the Kolmogorov-Smirnov test for the exponential distribution with mean unknown. *J. Am. Stat. Assoc.* **1969**, *64*, 387–389. [[CrossRef](#)]
49. Lilliefors, H. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *J. Am. Stat. Assoc.* **1967**, *62*, 399–402. [[CrossRef](#)]
50. Shapiro, S.S.; Wilk, M.B. An analysis of variance test for normality (complete samples). *Biometrika* **1965**, *52*, 591–611. [[CrossRef](#)]
51. Stephens, M.A. EDF Statistics for Goodness of Fit and Some Comparisons. *J. Am. Stat. Assoc.* **1974**, *69*, 730–737. [[CrossRef](#)]
52. Wilk, M.B.; Gnanadesikan, R. Probability plotting methods for the analysis of data, *Biometrika. Biometrika Trust* **1968**, *55*, 1–17.
53. Bland, J.M.; Altman, D.G. Statistics notes: Measurement error. *BMJ* **1996**, *312*, 1654. [[CrossRef](#)]
54. Joanes, D.N.; Gill, C.A. Comparing measures of sample skewness and kurtosis. *J. R. Stat. Soc. (Ser. D) Stat.* **1998**, *47*, 183–189. [[CrossRef](#)]
55. Tian, Y.; Yin, Z.; Huang, M. Missing Data Probability Estimation-Based Bayesian Outlier Detection for Plant-Wide Processes with Multisampling Rates. *Symmetry* **2018**, *10*, 475. [[CrossRef](#)]
56. Li, G.; Wang, J.; Liang, J.; Yue, C. The Application of a Double CUSUM Algorithm in Industrial Data Stream Anomaly Detection. *Symmetry* **2018**, *10*, 264. [[CrossRef](#)]
57. Li, G.; Wang, J.; Liang, J.; Yue, C. Application of Sliding Nest Window Control Chart in Data Stream Anomaly Detection. *Symmetry* **2018**, *10*, 113. [[CrossRef](#)]
58. Ross, S.M. Peirce's Criterion for the Elimination of Suspect Experimental Data. *J. Engr. Technol.* **2003**, *2*, 1–12.
59. Motulsky, H.J.; Brown, R.E. Detecting outliers when fitting data with nonlinear regression: A new method based on robust nonlinear regression and the false discovery rate. *BMC Bioinform.* **2006**, *7*, 123. [[CrossRef](#)] [[PubMed](#)]
60. Tietjen, G.; Moore, R. Some Grubbs-Type Statistics for the Detection of Several Outliers. *Technometrics* **1972**, *14*, 583–597. [[CrossRef](#)]
61. Zerbet, A.; Nikulin, M. A new statistics for detecting outliers in exponential case. *Commun. Stat. Theory Methods* **2003**, *32*, 573–584. [[CrossRef](#)]
62. Dean, R.B.; Dixon, W.J. Simplified Statistics for Small Numbers of Observations. *Anal. Chem.* **1951**, *23*, 636–638. [[CrossRef](#)]
63. Grubbs, F.E. Sample criteria for testing outlying observations. *Ann. Math. Stat.* **1950**, *21*, 27–58. [[CrossRef](#)]
64. Barnett, V.; Lewis, T. *Outliers in Statistical Data*, 3rd ed.; Wiley: Hoboken, NJ, USA, 1994.
65. Grubbs, F.E. Procedures for Detecting Outlying Observations in Samples. *Technometrics* **1969**, *11*, 1–21. [[CrossRef](#)]

66. Stefansky, W. Rejecting Outliers in Factorial Designs. *Technometrics* **1972**, *14*, 469–479. [[CrossRef](#)]
67. Dantzig, G.B.; Fulkerson, D.R.; Johnson, S.M. Solution of a large-scale traveling-salesman problem. *Oper. Res.* **1954**, *2*, 393–410. [[CrossRef](#)]
68. Kovács, L.; Iantovics, L.B.; Iakovidis, D.K. IntraClusTSP—An Incremental Intra-Cluster Refinement Heuristic Algorithm for Symmetric Travelling Salesman Problem. *Symmetry* **2018**, *10*, 663.
69. Laporte, G. The Traveling Salesman Problem: An overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **1992**, *59*, 231–247. [[CrossRef](#)]
70. Merkle, D.; Middendorf, M. On solving permutation scheduling problems with ant colony optimization. *Int. J. Syst. Sci.* **2005**, *36*, 255–266. [[CrossRef](#)]
71. Runkler, T.A. Ant colony optimization of clustering models. *Int. J. Int. Syst.* **2005**, *20*, 1233–1251. [[CrossRef](#)]
72. Crisan, G.C.; Pintea, C.M.; Palade, V. Emergency Management Using Geographic Information Systems. Application to the first Romanian Traveling Salesman Problem Instance. *Knowl. Inf. Syst.* **2017**, *50*, 265–285. [[CrossRef](#)]
73. Pholdee, N.; Bureerat, S. Hybrid real-code ant colony optimisation for constrained mechanical design. *Int. J. Syst. Sci.* **2016**, *47*, 474–491. [[CrossRef](#)]
74. Dorigo, M.; Maniezzo, V.; Coloni, A. *Positive Feedback as a Search Strategy*; Dipartimento di Elettronica, Politecnico di Milano: Milano, Italy, 1991.
75. Coloni, A.; Dorigo, M.; Maniezzo, V. Distributed optimization by ant colonies. In *Actes de la Premiere Conference Europeenne Sur la vie Artificielle*; Elsevier Publishing: Paris, France, 1991; pp. 134–142.
76. Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Milano, Italy, 1992.
77. Jaradat, G.M.; Ayob, M. *An Elitist-Ant System for Solving the Post-Enrolment Course Timetabling Problem*; Zhang, Y., Cuzzocrea, A., Ma, J., Chung, K., Arslan, T., Song, X., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 167–176, FGIT 2010. CCIS 118.
78. Bullnheimer, B.; Hartl, R.F.; Strauss, C. A new rank based version of the ant system. A computational study. *Cent. Eur. J. Oper. Res.* **1999**, *7*, 25–38.
79. Cordón, O.; de Viana, I.F.; Herrera, F. *Analysis of the Best-Worst Ant System and its Variants on the QAP*; Dorigo, M., Di Caro, G., Sampels, M., Eds.; Springer: Heidelberg, Germany, 2002; pp. 228–234, ANTS 2002. LNCS 2463.
80. Stutzle, T.; Hoos, H.H. Max-min ant system. *Future Gener. Comput. Syst.* **2000**, *16*, 889–914. [[CrossRef](#)]
81. Dorigo, M.; Stützle, T. *Ant Colony Optimization*; MIT Press: Cambridge, MA, USA, 2004.
82. Buscarino, A.; Fortuna, L.; Frasca, M.; Latora, V. Disease spreading in populations of moving agents, IOP Publishing. *Europhys. Lett. (EPL)* **2008**, *82*, 38002. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).