A Proposal for the Organisational Measure in Intelligent Systems

Jesús-Ángel Román * † and María-Luisa Pérez-Delgado †

Escuela Politécnica Superior de Zamora, University of Salamanca, Av. Requejo 33, C.P. 49022 Zamora, Spain; mlperez@usal.es

* Correspondence: zjarg@usal.es; Tel.: +34-980545000 (ext. 3635)
† These authors contributed equally to this work.

Received: 2 February 2020; Accepted: 2 March 2020; Published: 6 March 2020

Abstract: The collaboration within organisations and among organisations is a fundamental concept in the attainment of the overall objectives pursued by an enterprise network in human companies. Swarm systems are intelligent systems that show collaboration within the system; moreover, some models, such as multiple ant colonies, show the collaboration of several systems to achieve a global goal. The collaboration in this type of system optimises the achievement of the overall objectives as in an enterprise network in human organisations. Being able to measure this collaboration allows establishing a relationship between the improvement in the results of the system and the degree of collaboration, both at the level of specialisation of each element of the system and the systems as a whole. The performance of a swarm system depends on the number of members in many cases, so that if we can establish a measure of specialisation and collaboration, we could tipify and classify these systems in terms of the efficiency and the realiability to perform different tasks.

Keywords: multi-ant system; enterprise network; swarm systems; specialisation; collaboration; intelligent systems

1. Introduction

In recent years several artificial intelligence methods have been proposed to tackle difficult problems. Among them, bio-inspired computation includes a set of methods that attempt to imitate behaviours or phenomena observed in nature in order to solve complex problems. Swarm-based algorithms are a subset of bio-inspired methods that have gained interest in recent years as powerful methods to solve complex problems [1]. These algorithms are based on the collective behavior of self-organised and decentralised systems [2,3]. They consider a population of very simple individuals (ants, particles or bees, among others) that cooperate to solve a problem. Each individual can perform a limited number of simple operations, based on the behaviours observed in nature. Despite the limited capacity of each individual considered independently, the swarm can solve complex problems.

If we not only consider the field of artificial intelligence but also consider all the solution methods proposed to address difficult problems, swarm algorithms belong to the group of solution methods called metaheuristics. A metaheuristic is a method designed to obtain an approximate solution for a wide range of difficult optimisation problems without having to deeply adapt to each problem. Metaheuristics can be classified into single-solution based metaheuristics and population based metaheuristics [4]. Swarm methods belong to the second group of metaheuristics, as they consider a set of solutions to define the final solution to the problem [5–7].

Swarm systems are considered organised and intelligent systems whose collaboration allows them to reach objectives with optimal results. For example, this is applied to find optimal solutions in a wide search space. Swarm intelligence allows the individual agents to model and self-organise...
through interactions with their neighbours. This type of intelligence arises from the social insects and from the existence of the organisation.

The concept of organisation has been extensively studied in sciences such as economics, sociology and psychology. Based on distributed artificial intelligence, we can apply the concept of organisation in these types of systems with the objective of developing intelligent systems in an efficient and organised way. The concept of specialisation is closely related to the concept of organisation, if specialisation is understood as a feature that gives an individual belonging to a group a particularity within that group. This specialisation is considered within organisations as a mechanism to achieve greater efficiency in the objectives they pursue.

Both in business organisations and in the swarm systems, the collaboration of the agents belonging to these systems is essential to achieve the established objectives. This collaboration can provide knowledge about the organisation’s environment to a group of members of that organisation, which allows them to benefit from this information and to obtain better results.

The specialisation in an organisation can be analysed from the perspective of business networks. In the literature, the concept of enterprise network can be found in [8–10] among others. This concept refers to a set of groups, institutions or organisations that interact with each other to obtain favourable results, both for the individual units and the network as a whole. Thus, a business network can be seen as an association of companies that collaborate with each other, so that there is a complementarity among them, with the aim of being able to solve situations that they could not solve individually, and doing it in the most efficient way possible. The concept of enterprise network in swarm systems appears in the multi-colonies of ants [11–13] and other multi-swarm systems. The collaboration among different swarms in the achievement of a common and global objective allows to improve the obtained results.

Based on the previous discussion, the possibility of measuring the specialisation and cooperation within an organisational system and among different organisational systems allows us to make a classification. This could allow comparing and relating these measures to the improvements of the systems when they cooperate together. With this purpose, this article considers several popular swarm-based systems and analyses the application of the measures to such systems. While swarm-based solutions usually consider a single swarm, there are also several methods that use a set of swarms. Such cases apply multi-swarm systems to one or more given problems, and they demonstrate the improvement of the results due to the collaboration among different swarms. Therefore, the article also applies the measures to multi-swarm systems, in order to analyse the relationship between cooperation and results of the system. As a result, we will show that the specialisation and collaboration measures defined for business organizations are also applicable to multi-swarm systems.

Section 2 introduces several popular algorithms for swarm systems. Section 3 defines specialisation as an improvement in productive processes and it presents several metrics to measure the types of specialisation and collaboration proposed. Section 4 analyses the specialisation in swarm systems proposed in this article. In Section 5, several multi-swarm methods described in the literature are analysed and collaboration among swarms is measured. Section 6 presents the conclusions.

2. Swarm Algorithms

Swarm algorithms mimic the behaviour of some natural systems that have intelligent behaviour. These systems are formed of members who perform simple individual operations, but the interactions among them lead to the emergence of collective intelligence called swarm intelligence.

Several algorithms which imitate the behaviour of those intelligent systems have been proposed. This article focuses on some of the most used: Ant-based algorithms [14,15], Particle Swarm Optimisation [16,17], Artificial Swarm Fish Algorithm [18], Artificial Bee Colonies [19,20] and Firefly Algorithm [21,22]. Some recent articles that analyse publications related to swarm methods show that the selected methods are among those considered in more publications [1,7]. This section includes a brief description of each method that can be completed by reading the references cited for each case.
Let us consider an optimisation problem whose objective function, denoted \( f(x) \), is defined on an \( r \)-dimensional space, where \( x \) is a vector with \( r \) components representing an element of the solution space. To solve the problem, these algorithms consider a set or swarm of \( n \) individuals that define a set of \( n \) feasible solutions to the problem, \( \{x_1, ..., x_n\} \), with \( x_i = (x_{i1}, ..., x_{ir}) \). The algorithms apply an iterative process that ends after a predetermined number of iterations, or when the solution converges. The solution represented by each individual is updated along the iterations. Therefore, to represent the solution associated with the individual \( i \) at iteration \( t \), this solution is denoted \( x_i(t) \). The quality or fitness of a solution is computed by applying the function \( f \) to that solution. The best solution found by the swarm along the iterations is selected as the solution to the problem. Table 1 summarises the notation used to define the algorithms described in this section.

### Table 1. Notation used to describe the algorithms.

| Symbol      | Description                                                                 |
|-------------|-----------------------------------------------------------------------------|
| \( f(x) \) | Objective function of a problem, which is computed for a feasible solution to the problem \( x \) |
| \( r \)    | Dimension of the solution space                                             |
| \( x_i = (x_{i1}, ..., x_{ir}) \) | A solution to the problem (a vector with \( r \) components)               |
| \( n \)    | Number of individuals or elements used by the algorithm to find a solution (ants, particles, fish, food sources or fireflies) |
| \( t \)    | The current iteration of the algorithm                                       |
| \( x_i(t) \) | The solution represented by the individual \( i \) at iteration \( t \), with \( i = \{1, ..., n\} \) |
| \( x' \)   | A tentative solution evaluated by the algorithm                             |
| \( \text{rnd}_{[\text{min}, \text{max}]} \) | A real value randomly selected from the interval \( (\text{min}, \text{max}) \) (that is, excluding extreme values), \( \text{rnd}_{[\text{min}, \text{max}]} \) or \( [\text{min}, \text{max}] \) (that is, including both extreme values). |
| \( \text{rnd} \) | When the limits are not specified, this means that they must be set for each specific problem |
| \( ||A - B|| \) | Euclidean distance between vectors \( A \) and \( B \)                      |

### 2.1. Ant-Based Algorithms

Ant-based algorithms are inspired by the behaviour of natural ants, which communicate among themselves by the pheromone that they deposit on the ground when they walk. If there is only one ant, it moves randomly; but if there are several ants, the movement of each ant is conditioned by the movement of the others. Ants prefer moving by paths with more pheromone and this movement increases the amount of pheromone of these paths, making them more desirable for other ants. The pheromone evaporates over time, and this makes the paths selected by a small number of ants less desirable.

The first algorithm proposed to mimic the behaviour of natural ants was called Ant System algorithm and it was applied to solve the Travelling Salesman Problem (TSP) [23,24]. Given a set of nodes interconnected by weighted connections, the objective of the TSP is to find a closed tour of minimum cost which includes each node once and only once [25,26]. The problem is represented by a graph with two values associated with each connection \( (i, j) \): the cost, which represents the distance between the nodes \( i \) and \( j \), and the pheromone \( \tau_{ij}(t) \in (0, 1] \). The Ant System method considers a set of \( n \) ants and applies the operations shown in Algorithm 1. The first operation defines the initial pheromone (random values are usually considered) and then an iterative process is applied. At each iteration, each ant defines a solution to the problem (a tour) and the cost of this solution is computed. After this, the pheromone is updated, to make the connections of the tours defined at the present iteration more desirable for the next iteration. The pheromone matrix is the means used by each ant to share information with the other ants.
Algorithm 1: Artificial ants algorithm

Initialise population;
repeat
  Each ant defines a solution to the problem by progressively adding new connections;
  The cost of the solution defined by each ant is computed;
  Update the pheromone;
until Requirements are met;

To define its tour, the ant \( k \) starts at a randomly selected node and progressively selects the other nodes. When it is at node \( i \), Equation (1) determines the probability of choosing \( j \) as the next node of its path. \( \eta_{ij}(t) \) is the visibility of the connection \((i, j)\); for a TSP, it is the inverse of the cost of \((i, j)\). \( N_k^i \) is the set of nodes connected to \( i \) not visited by the ant (Figure 1). The parameters \( \alpha \) and \( \beta \) determine the relative influence of the pheromone and the visibility, respectively. Equation (1) indicates that ants prefer connections with small length and high amount of pheromone.

\[
p_{ij}^k(t) = \frac{\tau_{ij}(t)\alpha \eta_{ij}(t)\beta}{\sum_{l \in N_k^i} \tau_{il}(t)\alpha \eta_{il}(t)\beta}
\]

Figure 1. The ant \( k \) is at node \( i \) and must select the next node of its path from \( N_k^i = \{r, s, t, u\} \).

The pheromone is updated by applying Equation (2) to each connection \((i, j)\) of the graph, where \( \rho \in (0, 1) \) is the evaporation rate of the pheromone and \( \Delta \tau_{ij}^k \) is the amount of pheromone deposited by ant \( k \) on the connection \((i, j)\). \( \Delta \tau_{ij}^k \) is inversely proportional to the cost of the solution defined by \( k \) if \((i, j)\) belongs to said solution, and it is 0 in other case.

\[
\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^{n} \Delta \tau_{ij}^k
\]

There are several variants of the previous algorithm, which use different state transition and updating rules. Some examples are the Ant Colony System [27], the Max-Min Ant System [28], the Rank-based Ant-System [29], or the Elitist Ant System [30]. These algorithms have been applied to several graph-based problems [31–35], and to real-world problems, such as waste management problems [36–40].

2.2. Particle Swarm Optimisation

The Particle Swarm Optimisation (PSO) algorithm was proposed by Kennedy and Eberhart [41]. It is inspired by the movement of a flock of birds or insects and it can be applied to solve optimisation problems [42,43]. With this purpose, the algorithm considers a set of \( n \) particles. Each particle \( i \) has a position \( x_i(t) \) and a velocity \( v_i(t) \) that are updated as \( t \) increases. Moreover, the particle ‘remembers’ the best solution it found until the current iteration, which is denoted \( b_i(t) \). Each particle moves on the search space guided by its previous experience and the experience of other particles, which is
represented by the velocity vector of the particle. The movement of all the particles tries to find a good solution to the problem. The best solution found by the swarm until iteration $t$ is denoted $g(t)$ and its value at the end of the algorithm defines the solution to the problem.

The new position of particle $i$, $x_i(t+1)$, is computed by Equation (3) as a function of the previous position, $x_i(t)$, and the velocity of the particle (Figure 2).

$$x_i(t+1) = x_i(t) + v_i(t+1)$$ (3)

To compute the new velocity of this particle (by Equation (4)), three elements are considered: The previous velocity, $v_i(t)$, the best position of this particle, $b_i(t)$, and the best solution found by the swarm, $g(t)$. Equation (4) includes three parameters that weight these three elements: $\omega$ (the inertia coefficient), $\phi_1$ and $\phi_2$. The velocity of each particle is limited to an interval $[v_{min}, v_{max}]$.

$$v_i(t+1) = \omega v_i(t) + \phi_1 r_{n1}(b_i(t) - x_i(t)) + \phi_2 r_{n2}(g(t) - x_i(t))$$ (4)

Figure 2. Movement of particle $i$ in a two-dimensional search space. The particle, which is currently at position $x_i(t)$, will move to a new position, $x_i(t+1)$, taking into account its new velocity, $v_i(t+1)$. That velocity is computed based on the current velocity of the particle, $v_i(t)$, the best position found so far by the particle, $b_i(t)$, and the global best position found so far by the swarm, $g(t)$.

The basic steps of PSO algorithm are shown in Algorithm 2. To start the operations, the particles are placed at random positions on the search space, and each particle is assigned a random velocity in $[v_{min}, v_{max}]$. Moreover, the best position of each particle equals its initial position. The global best particle is the one whose position gives the best value for the objective function of the problem. Next, an iterative process is applied to find a good solution to the problem. Each iteration updates the position and the velocity of the particles, by Equations (3) and (4). Then, the fitness of each particle is computed. This value is used to determine if the new position of the particle is better than the one considered the best so far, in which case this new position replaces the previous one. If some particle has better fitness than that considered as the global best, it becomes the new global best.

**Algorithm 2: Particle Swarm Optimisation algorithm**

1. Initialise the swarm of particles;
2. repeat
   1. Update the velocity of each particle;
   2. Update the position of each particle;
   3. Calculate the fitness of each particle;
   4. Update the memory of each particle;
   5. Update the global best solution;
3. until Requirements are met;
This algorithm has been applied to problems such as the task assignment problem [44], the flowshop scheduling problem [45], the capacitated vehicle routing problem [46], and several real-time optimisation problems [47].

2.3. Artificial Swarm Fish Algorithm

The Artificial Swarm Fish algorithm is based on the swarm behaviour of natural fish [18,48,49]. It imitates the collective movement of natural fish when they look for food or they are in danger.

The position of a fish \( i \) is a solution of the problem, \( x_i(t) \), and the fitness of the position represents the food concentration. Each fish can see a limited area around it, defined by \( \text{visual} \) (Figure 3), so as fish \( i \) can see fish \( j \) if the second one is in its neighbourhood: \( ||x_j(t) - x_i(t)|| < \text{visual} \).

When a fish moves, it prefers positions with more food than its current position and not too crowded. The position of a fish is crowded if the ratio between the number of fish in its \( \text{visual} \) and \( n \) reaches the value \( \rho \), where \( \rho \in (0, 1) \) is a parameter called crowd factor.

When a fish decides to move from its current position \( x_i(t) \) towards another position \( x' \), its new position is computed according to Equation (5), where the parameter \( \text{step} \) limits the distance it can swim.

\[
x_i(t+1) = x_i(t) + \frac{x' - x_i(t)}{||x' - x_i(t)||} \cdot \text{rnd}(0,1) \cdot \text{step}
\]

The algorithm imitates several behaviours of natural fish: Prey, swarm and follow. The prey behaviour allows the fish to move to a better position around its current position. A fish \( i \) randomly selects a position \( x' \) in its \( \text{visual} \) and then moves towards it according to Equation (5) if \( x' \) has more food than its current position. If the fish cannot move towards a better position after a certain number of attempts, it moves randomly within its \( \text{visual} \).

The swarm behaviour allows the fish to move towards the central point of its \( \text{visual} \). Let us use \( x' \) to represent the centre of the \( \text{visual} \) of fish \( i \). If the surrounding is not very crowded and the centre has more food than the current position of \( i \), the fish \( i \) swims towards \( x' \) according to Equation (5); in other case, \( i \) applies the prey behaviour.

The follow behaviour allows a fish to follow to other fish which has found food. Let \( j \) represent the fish in the position with the greatest food concentration in the visual area of fish \( i \). If \( x_j(t) \) has more food than \( x_i(t) \) and the surrounding is not very crowded, the fish \( i \) swims towards the fish \( j \) according to Equation (5) (where \( x' \) is replaced by \( x_j(t) \)); in other case, this fish applies the prey behaviour.

The main operations of the Artificial Swarm Fish algorithm are shown in Algorithm 3. After assigning random starting positions to the fish, an iterative process attempts to move them to better positions by applying the behaviours listed above. The behaviour of the fish \( i \) depends on its position and the state of the environment. When there are no fish in the visual of fish \( i \), it moves randomly to attempt to find a better position. When the visual area of \( i \) is crowded, the fish does not follow any
other fish; on the contrary, it moves to a random position in the visual area. When the visual area of $i$ is not crowded, this fish can move to the centre of its neighbourhood or to the best position in the visual.

**Algorithm 3: Swarm Fish algorithm**

Generate an initial population of $n$ fish;

repeat
  foreach fish $i$ do
    apply follow behaviour;
    if follow behaviour fails then
      apply swarm behaviour;
    end
    if follow behaviour fails then
      apply prey behaviour;
    end
  end
until Requirements are met;

Several improvements of the basic algorithm are described in [50–52].

2.4. Artificial Bee Colony

The Artificial Bee Colony (ABC) algorithm, developed by Karaboga [53,54], imitates the foraging behaviour of honey bees, and has been applied to solve several optimisation problems [55].

The bees of a hive communicate among themselves to get food. There are three types of bees which contribute to perform this task: Employed bees, onlooker bees, and scout bees. An employed bee goes to a food source and takes the nectar of such source to the hive. Moreover, this bee performs a dance on the dance area to inform to the other bees about the source it is exploiting. The onlooker bees are on the dance area and they choose a food source depending on the information obtained from the dance of the employed bees. The scout bees perform a random search of new food sources. When the nectar of a source is exhausted, the bees which exploited that source become scouts.

The ABC algorithm considers a set of $n$ food sources. The position of each food source $i$ is a solution to the problem, $x_i(t)$, and the fitness of such solution represents the nectar of the source. The operations of the bees allow them to find the best solution to the problem (the best food source).

The main operations of this algorithm are shown in Algorithm 4. First, $n$ food sources are randomly selected from the search space and their fitness is computed. Next, an iterative process is performed, which executes the operations of the three types of bees.

Each employed bee $i$ selects a tentative food source by Equation (6), where $k$ is a randomly selected neighbour food source (different from $i$). If $x'$ has more nectar than $x_i(t)$, the bee memorises the new position and forgets the old position. This process is repeated for each solution in the population.

$$x' = x_i(t) + \text{rnd}_{[-1,1]}(x_i(t) - x_k(t))$$

Each onlooker bee chooses a food source $i$ of the current set with a probability proportional to the nectar of the source (the probability is computed by dividing the nectar of the source $i$ by the sum of the nectar of all sources). If the selected source has more nectar than the previous position memorised by this bee, it forgets the previous position and memorises the new one.

The last phase includes operations of the scout bees, which allow to replace the food sources abandoned by the bees. The solutions which have not been improved after a predefined number of trials are replaced by randomly generated new solutions.
Algorithm 4: Artificial bees algorithm

Initialise the set of food sources;
repeat
  Operations of employed bees;
  Operations of onlooker bees;
  Operations of scout bees;
  Memorise the best food source found so far;
until Requirements are met;

Some applications of artificial bees are described in [56–60].

2.5. Firefly Algorithm

The Firefly algorithm is inspired by the flashing patterns and behaviour of natural fireflies. It was proposed by Yang as a method to solve optimisation problems [61,62], and it has been applied to problems such as image processing or clustering [63–66].

Each firefly $i$ defines a solution to the problem that evolves with $t$, $x_i(t)$, and the fitness of this solution represents the brightness of the firefly. The main operations of this algorithm are shown in Algorithm 5.

Algorithm 5: Firefly algorithm

Generate an initial population of fireflies;
repeat
  Move each firefly towards brighter fireflies according to Equation (7);
  Move the brightest firefly according to Equation (8);
  Update the brightness of the fireflies;
  Rank the fireflies according to their brightness and find the best;
until Requirements are met;

The first operation generates the initial population of $n$ fireflies. They are randomly distributed on the search space and then the brightness of each firefly is computed. Next, an iterative process is applied to move each firefly towards the partners brighter than it. The firefly $i$ moves towards a neighbour $j$ brighter than it (Figure 4), according to Equation (7) where $\gamma$ is the light absorption coefficient of the medium, $\beta_0$ is the attractiveness at distance 0, $\exp$ represents the exponential function. The second addend of Equation (7) takes values in $[\beta_0, 0)$ and it decreases as the distance between fireflies $i$ and $j$ increases.

$$x_i(t+1) = x_i(t) + \beta_0 \exp(-\gamma ||x_i(t) - x_j(t)||^2) (x_j(t) - x_i(t)) + \text{rnd}$$

(7)

The brightest firefly, denoted $\text{best}$, moves randomly according to (8), because it is not attracted to any other.

$$x_{\text{best}}(t+1) = x_{\text{best}}(t) + \text{rnd}$$

(8)

The brightness of each firefly is updated after it moves. When this new value is computed, the fireflies are sorted according to their brightness and the brightest firefly represents the best solution of the iteration. When the algorithm ends, the position of the brightest firefly found so far is the solution to the problem.
Figure 4. The firefly $i$ moves towards the neighbours brighter than it (fireflies 2, 3 and 7 in this example).

3. Specialisation

Specialisation as a particular feature of an individual within a group is present in all kinds of areas in our world. Biological systems, like ant colonies and swarms of bees, include the specialisation as a benefit to improve their aims [67]. In the productive systems, the specialisation is considered a way to achieve more efficiency in the common purposes of the group and into the group [8,68,69]. There are several types of specialisation, taking into account the origin, the size of the group or the depth of hierarchy into the group.

In biological systems, specialisation emerges in two ways: In terms of the structure of the system [70] and in their behaviour [67], but the aim of this specialisation is the efficiency in achieving the overall objectives of the group through the division of labor into specialised groups and specialised members.

Human organisations try to achieve their objectives by improving their productive processes and minimising their resources, taking into account that the principal aim is to increase the benefit of the organisation. Therefore, the specialisation in biological systems has been taken in many cases as a reference for human organisations. One way to increase the benefit among human organisations emerges in the cooperation among them. These synergies occur among different specialised organisations that collaborate for optimal performance on common productive processes, as in enterprise networks [8].

3.1. Determining the Specialisation of Tasks

First of all, it is necessary to define a measure of specialisation to classify the organisations, the members into an organisation and the tasks that they perform. In [71] the specialisation of tasks is defined as the number of different tasks performed by a worker and the frequency with which they are repeated, taking into account the decision-making ability of that job for self-management. Tasks specialisation, in turn, is divided into vertical and horizontal tasks specialisation.

Horizontal specialisation can be defined as the number of tasks to be performed by a particular job and its main objective is the increase in productivity. This specialisation is considered high when the tasks are repetitive and simple, and low when it is necessary to perform non-repetitive tasks. This specialisation regulates the extent of the job. On the other hand, vertical specialisation refers to the differentiation between the planning of the tasks and its control and execution, that is to say, if there is a hierarchy that plans and controls these tasks. The vertical specialisation is high when the worker only performs tasks that have been planned by someone else, and it is low when the tasks to be executed have not been planned by someone else. Both types of specialisation are necessary and they are closely related, since a job in which there is a high degree of horizontal specialisation also requires a high degree of vertical specialisation, because the person which performs a task may lose sight of the global objectives. Therefore, another person with a global vision must plan, organise and control such work [71]. The degree of specialisation of tasks in a specific job, $E_T$, is defined in [72] by Equation (9),

$$E_T = \frac{\sum_{t=1}^{T} \text{Frequency}_{t}}{\text{Number of Tasks}_{t}}$$
where $N_T$ is the number of tasks performed by each person in that job. Taking into account that in this case we are measuring the tasks that are performed in a single job position, we have $P = 1$. However, it is possible that in a work team (considered as a unit) several jobs are developed, in which case $P$ would take values other than 1.

$$E_T = \frac{P}{N_T}$$ (9)

If $E_{TH}$ is the degree of specialisation of horizontal tasks, the relationship between $E_{TH}$ and $E_T$ is given by Equation (10).

$$E_{TH}(E_T) = \begin{cases} 
\text{if } E_T \geq 0.2 & E_{TH} \text{ is very high} \\
0.2 > E_T \geq 0.1 & E_{TH} \text{ is high} \\
0.1 > E_T \geq 0.05 & E_{TH} \text{ is low} \\
E_T < 0.05 & E_{TH} \text{ is very low}
\end{cases}$$ (10)

The degree of vertical specialisation in a given job, $E_{TV}$, is computed by Equation (11), where $N_{TP}$ is the number of tasks that a person performs, which have been planned and controlled by another person. $N_{TP}$ must be in $(0, N_T]$; if $N_{TP} = 0$, there is no specialisation.

$$E_{TV} = \frac{P}{N_{TP}}$$ (11)

The relationship between $E_{TV}$ and $E_T$ is given by Equation (12). Taking into account that the vertical task specialisation measures the proportion of tasks that are planned externally on a job, this implies that this type of specialisation increases as the number of tasks planned externally approaches the total number of tasks. Therefore, the maximum is obtained when all the tasks performed in a job are externally planned.

$$E_{TV}(E_T) = \begin{cases} 
\text{if } E_{TV} \simeq E_T & E_{TV} \text{ is very high} \\
E_{TV} > E_T & E_{TV} \text{ is high} \\
E_{TV} \gg E_T & E_{TV} \text{ is low}
\end{cases}$$ (12)

The thresholds established are part of an observation process in the industrial environment, as well as the results presented in [73–75]. These investigations measure the dispersion in a job and the variation of the different roles affected in work environments. In addition, the research presented in [73] proposes a measure for the division of tasks into a job, which in some way is complementary to specialisation in that job. In this case, it is clear that if all the tasks performed by a job have been planned and it has no capacity to make decisions, the job is vertically specialised, that is, there is no autonomy. However, a specific threshold for this type of specialisation has not been defined; only a simple comparison has been defined which indicates that vertical specialisation increases as the number of externally planned tasks approaches the total number of tasks.

The definition of these thresholds allows to classify the degree of specialisation of tasks in a given job. Moreover, it is possible to transfer these thresholds to swarm systems and to determine the number of tasks performed by each member of the group; from this information, it is also possible to determine the degree of specialisation of tasks for each member. The metrics described in this section will be applied to swarm systems in Section 4.

3.2. Collaboration among Organisations

The collaboration among various organisations allows them to achieve more complex aims and to improve their resources. This requires the existence of a synergy among them from which the organisations can benefit. The concept of enterprise networks can be found in articles by authors cited in [8], where the organisations interact with each other to get the best results both as individual units or as a whole.
An enterprise network is considered as an association of companies working together to get more benefits while reducing their effort, so that there is a complementation among them with the aim of solving situations that they could not solve individually; they perform these operations in the most efficient way possible.

There are various objectives which govern these enterprise networks [10]:

- To increase competitiveness and profitability of the companies in the network.
- To induce specialisation of companies which belong to the network.
- To strengthen market presence of the companies that make up the network.
- To facilitate the access to business services that are inaccessible to them individually.

Taking into account the structure of the cooperation of the network and its specialisation, there are two types of enterprise networks: Horizontal networks and vertical networks.

Horizontal networks are represented as a collaboration of groups of companies offering the same services or products. These enterprises benefit from several activities, such as trading with providers; nevertheless, they compete in the same market. On the other hand, vertical networks are alliances among different companies that offer different products or services which are complementary and they collaborate to achieve competitive advantages that would be impossible individually.

In these terms, it is necessary to measure the degree of cooperation among these companies, and the degree of specialisation within an organisation and within an enterprise network. These measures were presented in [72]. Let \( N_E \) be the number of employees and \( N_{TE} \) the number of tasks performed in a company. A task represents whichever operation related to administration, marketing, etc., in addition to those related to the various production processes. Then, the degree of corporate specialisation, \( E_E \), is given by Equation (13).

\[
E_E = \frac{N_{TE}}{N_E} \tag{13}
\]

The corporate specialisation is very high when \( E_E \) is close to 0; it is high when \( E_E \in (0, 0.75] \); it is low when \( E_E \in (0.75, 1] \) and it is very low when \( E_E > 1 \). The ranges defined for \( E_E \) have been obtained from direct observation of the industrial environment and from the investigation described in [73–75].

We take into account that \( E_E \) improves with the dedication of a greater number of workers to global tasks in the enterprise. In this way, if there are a greater number of workers who are dedicated to perform a specific job, \( E_E \) increases; however, if a worker must perform many tasks, dedication to each of them affects this specialisation. In fact, taking into account that to determine this measure the tasks to be considered correspond to tasks such as marketing, administration or counting, it is possible that the number of tasks is less than the number of workers.

In addition, it is interesting to quantify the degree of cooperation, both vertical and horizontal, relative to a group of companies that constitute a business network. To do this, we define the following indicators: \( G_{CV} \) is the degree of vertical cooperation and \( G_{CH} \) the degree of horizontal cooperation within the enterprise network, according to the terms defined above. Taking into account [10], enterprise networks must not exceed a certain number of members with a threshold of 15 or 20. Then, the degree of cooperation, \( G_C \), is computed by Equation (14).

\[
G_C = \frac{N_{SP}}{N_{EM}} \tag{14}
\]

\( N_{EM} > 2 \)  number of organisations that belong to the enterprise network
\( N_{SP} > 0 \)  number of products or services which the organisations offer
The value of $G_C$, which is greater than 0, allows to obtain the degrees of vertical and horizontal cooperation by Equation (15) and Equation (16), respectively.

\[
G_{CV}(G_C) = \begin{cases} 
\text{if } G_C > 1 & G_{CV} \text{ is high} \\
\text{if } G_C = 1 & G_{CV} \text{ is medium} \\
\text{if } G_C < 1 & G_{CV} \text{ is low}
\end{cases} 
\] (15)

\[
G_{CH}(G_C) = \begin{cases} 
\text{if } G_C < 1 & G_{CH} \text{ is high} \\
\text{if } G_C = 1 & G_{CH} \text{ is medium} \\
\text{if } G_C > 1 & G_{CH} \text{ is low}
\end{cases} 
\] (16)

Many authors agree that the division of the objectives pursued by a group specialised in subtasks benefits the achievement of those objectives [76–78]. Nevertheless, the collaboration of specialised groups improves the consecution of the aims pursued. Thus, it is intended to perform an adaptation of the concept of enterprise network and its measure into the swarm systems.

4. Specialisation in Intelligent Systems

In artificial swarm systems, specialisation is an issue analysed by several authors in articles like [79,80], among others. This specialisation aims to provide more efficiency for the artificial system, so that there is an improvement in the achievement of its overall objectives.

The analysis of the specialisation in human organisations and in swarm systems, their similarities, and their implementation, has allowed to increase the efficiency of the group, and it has been able to implement improvements in the structure and operation on both types of organisations. In our case, we intend to measure the application of the specialisation of human organisations into the swarm systems, together with the organisation as enterprise networks that improve the efficiency of the units in the system.

According to [81] specialisation means more than just being diverse. In general, diversity means difference among individuals, although it does not matter whether the difference is good or bad with respect to the system performance. However, a system becomes specialised when, given specific constraints at the agent level, its diversity is focused for better system performance. In short, specialisation is the part of diversity that is demanded for better performance. In these terms, a specialisation metric should measure the part of diversity that enhances the performance. In reference [82] O’Donnell and Jeanne defined specialisation of an individual as the entropy of the proportions of its activity, so that low entropy means that an individual focuses on less activities.

However, the specialisation in human organisations allows to obtain a set of competitive advantages, improving their processes depending on the collaborative structure used. This organisational specialisation is given by the number of products or services the company offers to its customers. In addition, it is important to know and differentiate which type of cooperation is practised in this type of synergy. For this reason, a set of measures of specialisation have been defined in Section 3.1, both at the level of tasks within an organisation and at the level of cooperation among different companies. These measures can be implemented on the different intelligent systems so that they can be characterised in terms of specialisation and cooperation. In order to measure the specialisation within several intelligent systems, we have defined a set of tasks performed by each of the roles of these systems. From this definition of tasks we can determine the degree of specialisation in these systems.

Table 2 shows the tasks performed by each entity belonging to a particular system, from which we can determine the degree of specialisation of tasks of each component. Based on Tables 2 and 3 shows specialisation values computed for each swarm system.

The degree of vertical specialisation in a given job, $E_{TV}$, is not included in Table 3 because in swarm systems there is not a control on an upper level which defines either the global operation of the swarm or the operation of each individual. On the other hand, it can be observed that $E_{TH}$ is very high
in all cases except for fish systems, which correspond to the case for which more basic tasks have been identified for an individual entity.

Table 2. Basic tasks of a single entity.

| Swarm | Description                                                                 | Number |
|-------|-----------------------------------------------------------------------------|--------|
| Ant   | Choose the next connection of the path                                       | 3      |
|       | Determine the fitness of the path                                           |        |
|       | Update the pheromone                                                        |        |
| Particle | Update position                                                                   |        |
|       | Update velocity                                                              |        |
|       | Determine the fitness of the new position                                    | 3      |
| Fish | prey-1: Take a random position                                               |        |
|       | prey-2: Compute the food concentration of this position                      |        |
|       | prey-3: Move towards this position if it is better than the present position |        |
|       | prey-4: Move towards this position if a predefined number of unsuccessful attempts of movement towards a better position have been reached |        |
|       | swarm-1: Determine the central point of the neighbourhood                    |        |
|       | swarm-2: Compute the food concentration of this point                        |        |
|       | swarm-3: Move towards this position if it is better than the present position |        |
|       | follow-1: Determine the neighbour whose position has more food concentration  |        |
|       | follow-2: Move towards this position if it is better than the present position |        |
|       | (prey-2 and swarm-2 are the same operation; prey-3, swarm-3 and follow-2 are the same operation) | 6      |
| Bees | Employed bee-1: Take a random position as a candidate food source            |        |
|       | Employed bee-2: Compute the nectar of this position                          |        |
|       | Employed bee-3: Replace the previous food source by the candidate food source if the second one has more nectar |        |
|       | Onlooker bee-1: Choose another food source (with a certain probability)      |        |
|       | Onlooker bee-2: Memorise this food source and forget the previous one if the second one has more nectar |        |
|       | Scout bee-1: Take a random position as a new food source                     |        |
|       | Scout bee-2: Compute the nectar of this position                            | 3-2-2  |
| Firefly | Move (towards another firefly or randomly)                                  |        |
|       | Determine the fitness of the new position                                    | 2      |

Table 3. Measure of specialisation in several systems \((E_T: \text{Degree of specialisation of tasks}; E_{TH}: \text{Degree of specialisation of horizontal tasks}; E_E: \text{Degree of corporate specialisation}; n: \text{Number of individuals or particles})\).

| System                  | Rol               | \(E_T\)  | \(E_{TH}\)          | \(E_E\)                                      |
|-------------------------|-------------------|----------|----------------------|----------------------------------------------|
| Ant systems             | Ant               | 0.33     | Very high            | High or very high if \(n \geq 4\)            |
| PSO systems             | Particle          | 0.33     | Very high            | High or very high if \(n \geq 4\)            |
| Fish systems            | Fish              | 0.17     | High                 | High or very high if \(n \geq 8\)            |
| Artificial bees systems | Employed bee      | 0.33     | Very high            | High or very high if \(n \geq 10\)           |
|                         | Onlooker bee      | 0.50     | Very high            | High or very high if \(n \geq 10\)           |
|                         | Scout bee         | 0.33     | Very high            | High or very high if \(n \geq 10\)           |
| Firefly systems         | Firefly           | 0.5      | Very high            | High or very high if \(n \geq 3\)            |

In Table 3, the corporate specialisation is not a determining factor since it depends on the number of individuals or particles. Nevertheless, it should be noted that if the number of individuals or particles is greater than or equal to the value given in column 5, the corporate specialisation is considered to be high or very high. It should be noted that the thresholds for the values presented in the table were explained in Section 3.
Taking into account the results obtained, it can be observed that the different systems have a specialisation of tasks high or very high, so that the different resources are focused on the achievement of a certain set of tasks. The characterisation of the high degree of specialisation in swarm systems reveals the importance of collaboration among the members of the system, so that the individual work carried out by each element allows to achieve specific aims that are more complex, which are unattainable for a single member. In the same way, the business specialisation is not determined, because the number of organisational elements is not defined, although it is true that the number of groups of tasks to be performed is small enough to consider that the specialisation is high or very high.

Once the specialisation has been measured in these systems, it is interesting to measure the degree of cooperation in different systems considered as a single unit to achieve more complex goals, or to obtain them with less cost [10]. An example of business collaboration in the swarm systems is given in multi-colonies of ants [83–85], where there are several colonies of ants, each one with its own array of pheromones, that cooperate with their neighbouring colonies for the construction of a solution and they share information. This is discussed in the next section.

5. Discussion: Collaboration in Intelligent Systems

The collective intelligence arises from the collaboration or competition of a number of individuals or groups of individuals and from decision making in their processes. The union of specialisation and cooperation to solve a given problem allows to achieve much more complex objectives. Cooperation among intelligent systems like swarm systems allows to perform more complex tasks or tasks with less cost of resources, as in the business networks [8,9]. This cooperation can be measured in the terms listed in Section 3, which gives us an idea of the degree of collaboration among different companies that are related to the benefit of this synergy. This aspect of swarm systems has been analysed in papers such as [11–13], where multi-colonies of ants are considered.

Multi-swarm approach lets a set of swarms to cooperate in order to achieve better solutions. There are two distinct approaches well defined in the literature: Heterogeneous and homogeneous. In the first case, there is a set of swarms that have a behaviour that differs in each swarm [86–90]. In the second case, the swarms have a similar behaviour and the information is exchanged among swarms [91–94]. These types of approaches match with those proposed in business specialisation in humans, in terms of the type of vertical and horizontal cooperation defined in Section 3.

Several authors have successfully used multi-swarm solutions to solve multi-objective optimisation problems. The approach presented in this article is to show a set of problems presented in the literature and to measure the degree of cooperation among the different swarms. Thus, we can demonstrate that the cooperation of different swarms, as an enterprise network, improves the results obtained by an individual swarm.

The following subsections show multi-swarm examples associated with each swarm model described in Section 2 of this article. For each example, the degree of cooperation among the swarms is measured. In addition, the results discussed in such subsections are summarised in Table 4. Several articles define methods that solve a single-objective problem, while other solve multi-objective problems. In the last case, we use $D$ to denote the number of objectives that the multi-objective problem includes.

5.1. Multi-Colonies of Ants

- Gambardella et al. present in [86] the results obtained by multi-colonies of ants applied to solve the vehicle routing problem with time windows. In the proposed solution, the authors seek to simultaneously minimize the number of vehicles and the time of each trip (the first takes precedence over the second one). In this case two colonies are used, so that $N_{SP} = 2$ and $N_{EM} = 2$ and this gives $G_C = 1$ as defined by Equation (14). Taking into account that the objectives of each colony are different, the $G_{CV}$ is medium. The authors conclude in their work that the collaboration is the key factor that improves the results obtained. Therefore, the measure of this collaboration
is important to determine the scale of improvements of the system when such collaboration is implemented in the system.

- Salas et al. propose in [87] a variant of the Ant Colony System algorithm, based on the actual behaviour of ants: The Multi-type Ant Colony System (M-ACS). The M-ACS introduces the use of multiple types of ants organised in colonies, where each colony builds a global solution of the problem. During the process, the colonies cooperate by exchanging information of pheromones, besides competing through a mechanism of repulsion. The experimental results demonstrate that the proposed approach performs better in some cases in terms of effectiveness (final value of the objective function). In this paper the authors use three ant colonies for the resolution of the TSP. In this case there is no vertical cooperation because all the colonies pursue the same objective. The objective of the problem is to find a Hamiltonian cycle: A path that starts and ends at the source node and visits every node exactly once. This is the reason why \( N_{EM} = 3 \), which gives \( G_C = 0.33 \). Therefore, \( G_{CH} \) is high. The results obtained by the authors are competitive compared to those obtained in the specialised literature.

### 5.2. Multi-Colonies of Particles

- Zhang et al. [88] described a method with cooperation among swarms of particles. This method, called Multi-swarm Cooperative Multi-objective Particle Swarm Optimiser, uses a master swarm and multiple slave swarms. Each slave swarm optimises one objective function of the multi-objective problem, while the master swarm explores areas with sparse solutions of the search space. With this purpose, a variant of the PSO algorithm is applied to each type of swarm. The improved PSO algorithm applied to each slave swarm obtains the non-dominated optima of the objective function associated with this swarm. On the other hand, the PSO algorithm applied to the master swarm generates a good Pareto optimal set for the multi-objective problem. In this way, the swarms cooperate to solve the problem.

In this case, \( N_{EM} = D + 1 \) (D slave swarms and a master swarm) and \( N_{SP} = D + 2 \), since there is a service associated with each swarm and the final solution to the problem must also be considered. Therefore, \( G_C = (D + 2) / (D + 1) \). Once the degree of business cooperation has been obtained, the degrees of vertical and horizontal cooperation, that are inverse, can be obtained. In this case, the degree of vertical cooperation is high and the degree of horizontal cooperation is low, since there are as many functions as swarms (each sub-function and the global function) and there are several enterprises \((D + 1)\).

- The MCPSO method proposed by Ben et al. [95] is a multi-swarm solution based on the use of two types of swarms. The population used to solve a problem includes a master swarm and 3 slave swarms. Each slave swarm applies the basic PSO operations independently. On the other hand, the master swarm adjusts its trajectory according to its own experience and the knowledge of the slave swarms. The slaves share information with the master, but the information is not shared with other slaves. The results reported in the article showed that the new method outperforms the original PSO.

The method presented in this article considers four swarms each of which seeks a solution for the same function independently but also collaborates with the others. In this case, \( N_{SP} = 1 \) (a single-objective problem is solved) and \( N_{EM} = 4 \) (3 slaves + 1 master). This generates \( G_C = 0.25 \) and we can conclude that \( G_{CV} \) is low and \( G_{CH} \) is high. In this method, cooperation is clearly horizontal, since the slave swarms solve the same problem independently and then the master applies its own knowledge and that of the slaves to determine the final solution, so there is only one final product or service.
5.3. Multi-Colonies of Fish

- Yazdani et al. present in [90] a method called Parent-Child Artificial Swarm Fish algorithm, for optimisation in dynamic environments. In this case, the objective function includes several peaks that can define a global optimum and the optimal positions may change when the environment changes.

This method uses several swarms that operate simultaneously and independently. There are two basic types of swarms: Parent and child swarms. Parent swarms are defined to find peaks quickly, while child swarms are defined to cover a peak and then track it. Child swarms are divided into two groups: Best child swarm (the child swarm that includes the fish with the best fitness) and non-best child swarms. All swarms apply the basic operations of the Swarm Fish algorithm, but each type of swarm uses different parameter values to achieve its objective.

When the algorithm starts, there are only parent swarms. When a parent swarm converges to a peak, it generates a child swarm that replaces the parent at the peak. Then, this parent is reset and looks for another peak that has not been previously found. Therefore, the total number of swarms increases with time, and this allows to cover all the peaks of the search space with child swarms. When the environment changes, each child swarm tracks the peak on which it has been placed by performing a local search. The parameters used by each type of child swarm to perform this search are different, in order to allow the best child swarm to converge to the optimum and that the remaining child swarms move from a peak to a better one.

The article uses a parent swarm to start the process. Both this swarm and the child swarms that are generated later include two fish. The multimodal functions analyzed include between 1 and 200 peaks. Therefore, the final number of child swarms depends on the problem considered. This causes $N_{EM}$ to take values in $[2, 201]$. On the other hand, $N_{SP}$ takes values in the interval $[2, 201]$, since we must consider the global function together with the number of peaks. As a result, $G = 1$ because the number of swarms (parent + children) is always equal to the number of services (peaks + global function). Therefore, we can conclude that both $G_{CV}$ and $G_{CH}$ are medium. It should be noted that in this case there are as many services or products (functions to optimise) as swarm systems.

- Zhu et al. [96] proposed the Niche Quantum Artificial Fish Swarm Algorithm to solve multimodal problems.

This method considers the basic Swarm Fish algorithm and includes two new mechanisms: Quantum (used to enhance population variety) and niching (used to find multiple optimal solutions). In this case, each peak is considered a niche and a swarm of fish is used to populate each niche.

The fish are divided into several swarms to form the niches. Then, the swarms are placed randomly in the search space and operate in parallel.

As a result, the swarms locate multiple solutions (peaks or niches) of the multimodal function. The method uses the restricted competition selection strategy to maintain the niches. This strategy compares the best fish in each swarm and initialises the worst fish considered in each comparison.

The tests described in the article consider 10 swarms ($N_{EM} = 10$) with 10 fish each, which are applied to solve problems with a maximum of 10 peaks. Thus, in this case, the number of selected peaks ranges between 1 and 10 ($N_{SP} \in [1, 10]$), so that the degree of cooperation is modified based on these peaks. As a result, $G \leq 1$. Therefore, the degree of horizontal cooperation will be high in most cases, and will be medium only in case there are as many peaks as swarms.
5.4. Multi-Colonies of Bees

- Zhong et al. propose a multi-objective ABC algorithm [89] that uses three colonies: Two basic colonies and a synthetic colony. The search space is divided into two independent subsets. Then, each basic colony searches for solutions in a subset, while the synthetic colony analyses the entire search space. The method stores in a file the non-dominated solutions found so far; this information can be stored by both types of colonies. The basic colonies share information with the synthetic colony by updating the global best solution; nevertheless, there is no direct communication between the basic colonies. The three colonies use neighbourhood information to generate new food sources, and the synthetic colony also uses the global best food source selected from the file.

The method was applied to problems with two or three objectives. Three colonies with 10 bees each and five food sources were used in the tests.

Given that there is only one global goal which is divided into two sub-objectives, and that there are several colonies, we can set \( N_{EM} = 3 \) and \( N_{SP} = 1 \). In this case we have \( G_C = 0.33 \) and this implies that \( G_{CH} \) is high and \( G_{CV} \) is low. This result makes sense because several colonies pursue the same objective and collaborate to solve the same problem, although internally each colony seeks the solution in its subset.

- Subotic et al. [92] proposed a method called MS-ABC, which considers several swarms of bees. This method divides the population into four swarms of equal size. Each swarm executes the original ABC algorithm and the results of the swarms are shared periodically. The method includes a modification of the ABC algorithm to avoid the loss of the best solution when it is not improved after several attempts.

The article shows results of the application of MS-ABC to 12 single-objective problems. This method was applied considering four swarms with 16 bees each. The article shows that MS-ABC obtains better results than the basic ABC applied to the same problems.

In this case, collaboration among swarms is important to achieve the overall goal. The values of the parameters are \( N_{EM} = 4 \) and \( N_{SP} = 1 \), since the final business objective is the resolution of the local problem of each subgroup. Both parameters give the value 0.25 for \( G_C \). Therefore, \( G_{CV} \) is low and \( G_{CH} \) is high. Since all subpopulations solve the same problem and share information, specialisation in cooperation will be horizontal.

5.5. Multi-Colonies of Fireflies

- Dugonik and Fister proposed a variant of the firefly algorithm called MPFA [93]. This method divides the fireflies into subpopulations that initially have the same size and then applies the operations of the original Firefly algorithm to each subpopulation. In addition, subpopulations communicate with each other and can exchange individuals.

The article compares the results obtained by MPFA and those of the original Firefly algorithm applied to 16 single-objective functions. Both methods consider a population that includes 100 fireflies. Three variants of MPFA are considered, with two, three and four subpopulations with the same number of fireflies (50, 33 and 25 fireflies per subpopulation, respectively). MPFA obtains better results when four subpopulations are considered, and this method also generates better results that the original Firefly algorithm that considers a single population.

Taking into account that the article shows results for three cases with different number of subpopulations, we can compare the degree of cooperation and the efficiency. The value of \( N_{EM} \) is equal to the number of subpopulations, that is, \( N_{EM} = 2 \) for two subpopulations, \( N_{EM} = 3 \) for three subpopulations and \( N_{EM} = 4 \) for four subpopulations. In all cases \( N_{SP} = 1 \). These values generate three possible results for \( G_C \): \{0.5, 0.33, 0.25\}, so that the horizontal cooperation...
is high for all cases and the vertical cooperation is low for all cases. The results are better when four subpopulation are considered and for this case we can conclude that the highest horizontal cooperation is the most efficient.

- Abshouri et al. [94] proposed a multi-swarm firefly method applied to solve multimodal problems. The set of fireflies is divided into several swarms that interact to solve the problem. Each swarm applies the firefly movements defined in the original Firefly algorithm and this allows the swarms to be positioned at different peaks of the search space. There are two operations that define the interaction of swarms: A local interaction between colliding swarms prevents swarms from settling on the same peak, and a global interaction allows all swarms to share information, allowing new peaks to be found. When each swarm has converged to a different peak and there are fewer swarms than peaks, the algorithm restarts the swarm with the worst function value. In addition, when the distance between two swarms is less than a threshold, the swarm with the worst function value is reset.

The algorithm was applied to problems including 10 peaks at most. The population size was set to 100 and 10 swarms were defined. In this case, there are the same number of swarms and peaks, that is, \( N_{EM} = 10 \) and \( N_{SP} = 10 \), and \( G_C = 1 \). In these terms, \( G_{CV} \) and \( G_{CH} \) are considered medium.

| Swarm Type | Reference | \( N_{EM} \) | \( N_{SP} \) | \( G_C \) | \( G_{CV} \) | \( G_{CH} \) |
|------------|-----------|-------------|-------------|-----------|-----------|-----------|
| ants [86]  | [87]      | 2           | 2           | 1         | medium    | medium    |
| particles  | [88]      | \( D + 1 \) | \( D + 2 \) | 0.33      | high      | low       |
|            | [95]      | 4           | 1           | 0.25      | high      | low       |
| fish [90]  | [96]      | \( D + 1 \) | \( D + 1 \) | 1         | medium    | medium    |
|            |           | 10          | \([1,10]\)  | \([0.1,1]\) | low/medium| high/medium|
| bees [89]  | [92]      | 3           | 1           | 0.33      | low       | high      |
|            | [93]      | 4           | 1           | 0.25      | low       | high      |
| fireflies  | [93]      | \{2,3,4\}  | 1           | \{0.5,0.33,0.25\} | low     | high      |
|            | [94]      | 10          | 10          | 1         | medium    | medium    |

### 5.6. Interpretation of Results

The previous discussion shows that cooperation among the different swarms to achieve objectives in a more efficient way and its measure are two important concepts to classify this form of cooperation and to compare its performance with the measure of the cooperation.

The results obtained for the multi-swarm methods described depend on the type of problem they are trying to solve. The systems that solve single-objective problems have high horizontal cooperation, because each system focuses on obtaining the best possible global result. When considering the systems that solve multi-objective problems, the results obtained for \( G_{CH} \) and \( G_{CV} \) depend on whether each swarm solves an objective of the multi-objective problem or if all the swarms solve the global problem. The systems that include the same number of swarms as the number of sub-objectives of the problem have medium \( G_{CH} \) and \( G_{CV} \), because the systems are not positioned in a clear and specific way to solve either the global problem or different local problems. On the other hand, vertical cooperation is high when the system includes fewer swarms than sub-objectives associated with the problem, because the achievement of partial objectives allows the achievement of a global objective.
6. Conclusions

In swarm systems their members are autonomous, they do not need a leader to indicate the way forward, because their operations (the basic operations that each individual in a swarm can perform, which are the same for all individuals) are well defined and are geared towards achieving the goal of the global system. While each individual of the swarm performs very basic operations, collaboration among all individuals allows the swarm to solve difficult problems.

However, the specialisation in swarm systems is an horizontal specialisation, taking into account that there are no monitoring tasks performed by an external agent. Therefore, we can define an artificial swarm system organisation as an organisation where the control and monitoring are realised by its members; it captures and builds knowledge to understand itself and its environment and it constantly and consistently develops the dynamic knowledge base to transfer it [9]. Intelligence and business organisation have a direct relationship with the swarm systems, so that the optimisation capability offered by this type of systems is used to improve the processes and the organisation of the companies.

In this way, the collaboration of the agents involved within systems, as well as the collaboration of the systems with other systems as an enterprise network, is basic to achieve objectives that a unit could not achieve. The definition and measure of the cooperation among the different agents and the specialisation of a company or a swarm system allows us to elaborate a classification of these systems and to relate the improvement in the achievement of its results with this collaboration.

To obtain these measures, the thresholds for the specialisation and the cooperation proposed in [72] have been considered, and they have been applied to the systems analysed.

In addition, a study of cooperation has been carried out, based on multi-swarm systems working together to achieve a global objective. For this purpose, we have studied the results obtained in a set of research papers, concluding that a greater degree of cooperation implies an improvement in the results.

In summary, the main contribution of this article is the novel proposal of a measurement of specialization in swarm systems. Future research objectives include the use of our proposed measuring as some guideline in order to know which specialized system must be applied to solve a particular problem. It should be noted that, to develop this idea, it is necessary to have previously carried out a complete study of many of these systems and the different cases to which they have been applied.

Author Contributions: J.-Á.R. and M.-L.P.-D. conceived and designed the experiments; M.-L.P.-D. performed the formal analysis; J.-Á.R. performed the experiments; J.-Á.R. and M.-L.P.-D. wrote and reviewed the article. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Samuel Solórzano Barruso Memorial Foundation of the University of Salamanca (grant number FS/102015).

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References
1. Del Ser, J.; Osaba, E.; Molina, D.; Yang, X.S.; Salcedo-Sanz, S.; Camacho, D.; Das, S.; Suganthan, P.N.; Coello, C.A.; Herrera, F. Bio-inspired computation: Where we stand and what is next. Swarm Evol. Comput. 2019, 48, 220–250. [CrossRef]
2. Nedjah, N.; Macedo, L. Swarm Intelligent Systems; Springer: Heidelberg, Germany, 2006; Volume 26.
3. Binitha, S.; Sathya, S.S. A survey of bio inspired optimization algorithms. Int. J. Soft Comput. Eng. 2012, 2, 137–151.
4. Talbi, E.G. Metaheuristics: From Design to Implementation; John Wiley & Sons: Hoboken, NJ, USA 2009; Volume 74.
5. Boussaïd, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. Inf. Sci. 2013, 237, 82–117. [CrossRef]
6. Nesmachnow, S. An overview of metaheuristics: Accurate and efficient methods for optimisation. Int. J. Metaheuristics 2014, 3, 320–347. [CrossRef]
7. Kar, A.K. Bio-inspired computing—A review of algorithms and scope of applications. *Exp. Syst. Appl.* 2016, 59, 20–32. [CrossRef]

8. Becerra, F. Las redes empresariales y la dinámica de la empresa: Aproximación teórica. *INNOVAR Revista de Ciencias Administrativas y Sociales* 2008, 18. Available online: http://www.redalyc.org/pdf/818/81803203.pdf (accessed on 14 September 2019).

9. Toca Torres, C.E. Inteligencia colectiva: Enfoque para el análisis de redes. *Estudios Gerenc.* 2014, 30, 259–266. [CrossRef]

10. López, C. Redes Empresariales: Experiencias en la Región Andina; Editorial Minka: Lima, Peru, 2003.

11. Marulanda, J.; Alfonso, W.; Caicedo, E. Competitive multi-swarm system in adaptive resource allocation for a multi-process system. *Revista Facultad de Ingeniería Universidad de Antioquia* 2013, 66, 168–180.

12. Middendorf, M.; Reischle, F.; Schmeck, H. Multi colony ant algorithms. *J. Heuristics* 2002, 8, 305–320. [CrossRef]

13. Melo, L.; Pereira, F.; Costa, E. (Eds.) MC-ANT: A multi-colony ant algorithm. In *Proceedings of the 9th International Conference on Artificial Evolution*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 25–36.

14. Stützle, T.; López-Ibáñez, M.; Dorigo, M. A concise overview of applications of ant colony optimization. In *Wiley Encyclopedia of Operations Research and Management Science*; Wiley Online Library: New York, NY, USA, 2010.

15. Dorigo, M.; Stützle, T. Ant colony optimization: overview and recent advances. In *Handbook of Metaheuristics*; Springer: Cham, Switzerland, 2019; pp. 311–351.

16. Zhang, Y.; Wang, S.; Ji, G. A comprehensive survey on particle swarm optimization algorithm and its applications. *Math. Probl. Eng.* 2015, 2015, 931256. [CrossRef]

17. Cheng, S.; Lu, H.; Lei, X.; Shi, Y. A quarter century of particle swarm optimization. *Complex Intell. Syst.* 2018, 4, 1–13. [CrossRef]

18. Neshat, M.; Sepidnam, G.; Sargolzaei, M.; Toosi, A.N. Artificial fish swarm algorithm: A survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artif. Intell. Rev.* 2014, 42, 965–997. [CrossRef]

19. Karaboga, D.; Gorkemli, B.; Ozturk, C.; Karabog, N. A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.* 2014, 42, 21–57. [CrossRef]

20. Agarwal, S.K.; Yadav, S. A Comprehensive Survey on Artificial Bee Colony Algorithm as a Frontier in Swarm Intelligence. In *Ambient Communications and Computer Systems*; Springer: Singapore, 2019; pp. 125–134.

21. Fister, I.; Yang, X.S.; Fister, D.; Fister, I., Jr. Firefly algorithm: A brief review of the expanding literature. In *Cuckoo Search and Firefly Algorithm: Theory and Applications, Studies in Computational Intelligence*; Springer: Heidelberg, Germany, 2014; Volume 516, pp. 347–360.

22. Tilahun, S.L.; Ngnotchouye, J.M.T. ‘Firefly algorithm for discrete optimization problems: A survey. *KSCE J. Civil Eng.* 2017, 21, 535–545. [CrossRef]

23. Dorigo, M.; Gambardella, L. A Ant System: An Autocatalytic Optimizing Process; Tech. Report 91-016; Dipartamento di Elettronica e Informazion, Politecnico di Milano: Milan, Italy, 1991.

24. Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Dip. Elettronica, Politecnico di Milano, Milan, Italy, 1992.

25. Dantzig, G.; Fulkerson, R.; Johnson, S. Solution of a large-scale traveling salesman problem. *J. Oper. Res. Soc. Am.* 1954, 2, 393–410. [CrossRef]

26. Reinelt, G. *The Traveling Salesman. Computational Solutions for TSP Applications*; Lecture Notes in Computer Science, no. 840; Springer-Verlag: Berlin/Heidelberg, Germany, 1994

27. Dorigo, M.; Gambardella, L. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* 1997, 1, 53–66. [CrossRef]

28. Stützle, T.; Hoos, H. (Eds.) The max-min ant system and local search for combinatorial optimization problems. In *Proceedings of the IEEE International Conference on Evolutionary Computation*; Indianapolis, IN, USA, 13–16 April 1997; pp. 309–314.

29. Bullnheimer, B.; Hartl, R.F.; Strauss, C. A new rank based version of the ant system. A computational study. *Cent. Eur. J. Oper. Res. Econ.* 1997, 7, 25–38.

30. Dorigo, M.; Maniezzo, V.; Colomni, A. Ant System: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* 1996, 26, 29–41. [CrossRef]
31. Bullnheimer, B.; Hartl, R.F.; Strauss, C. An improved ant system algorithm for the vehicle routing problem. *Ann. Oper. Res.* 1999, 89, 319–328. [CrossRef]
32. Vlachogiannis, I.G.; Hatzigiannou, N.D.; Lee, K.Y. Ant colony system-based algorithm for constrained load flow problem. *IEEE Trans. Power Syst.* 2005, 20, 1241–1249. [CrossRef]
33. Pérez-Delgado, M.-L. Rank-based ant system to solve the undirected rural postman problem. *Lect. Notes Comput. Sci.* 2009, 5518, 507–514.
34. Pérez-Delgado, M.-L. The undirected rural postman problem solved by the MAX-MIN ant system. *Adv. Intell. Soft Comput.* 2009, 55, 179–187.
35. Pérez-Delgado, M.-L. Elitist ants applied to the undirected rural postman problem. *Adv. Intell. Soft Comput.* 2010, 70, 221–230.
36. Bautista, J.; Fernández, E.; Pereira, J. Solving an urban waste collection problem using ants heuristics. *Comput. Oper. Res.* 2008, 35, 3020–3033. [CrossRef]
37. Pérez-Delgado, M.-L. Artificial ants and packaging waste recycling. *Lect. Notes Comput. Sci.* 2009, 5518, 596–603.
38. Pérez-Delgado, M.-L.; Matos Franco, J.C. Artificial intelligence for picking up recycling bins: A practical application. *Adv. Intell. Soft Comput.* 2009, 55, 392–400.
39. Wang, J.; Wu, X.; Fan, X. A two-stage ant colony optimization approach based on a directed graph for process planning. *Int. J. Adv. Manuf. Technol.* 2015, 80, 839–850. [CrossRef]
40. Rodríguez-Vazquez, K.; Garro, B.A.; Mancera, E. Solid waste collection in Ciudad Universitaria-UNAM using a VRP approach and Max-Min Ant System algorithm. *Lect. Notes Comput. Sci.* 2018, 11288, 76–85.
41. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
42. Kennedy, J. Particle swarm optimization. In *Encyclopedia of Machine Learning*; Sammut, C., Webb, G.I., Eds.; Springer: New York, NY, USA, 2011; pp. 760–766.
43. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* 2007, 1, 33–57. [CrossRef]
44. Salman, A.; Ahmad, I.; Al-Madani, S. Particle swarm optimization for task assignment problem. *Microprocess. Microsyst.* 2002, 26, 363–371. [CrossRef]
45. Pan, Q.K.; Tasgetiren, M.F.; Liang, Y.C. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Comput. Oper. Res.* 2008, 35, 2807–2839. [CrossRef]
46. Chen, A.; Yang, G.; Wu, Z. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *J. Zhejiang Univ. Sci. A* 2006, 7, 607–614. [CrossRef]
47. Zhang, H.; Hui, Q. Parallel multiagent coordination optimization algorithm: Implementation, evaluation, and applications. *IEEE Trans. Autom. Sci. Eng.* 2016, 14, 984–995. [CrossRef]
48. Li, L.X.; Qian, J.X. Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques. *J. Circuits Syst.* 2003, 1, 1–6.
49. Li, X.L.; Lu, F.; Tian, G.H.; Qian, J.X. Applications of artificial fish school algorithm in combinatorial optimization problems. *J. Shandong Univ. Eng. Sci.* 2004, 34, 64–67.
50. Cheng, Y.; Jiang, M.; Yuan, D. Novel clustering algorithms based on improved artificial fish swarm algorithm. In Proceeding of the IEEE Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China, 14–16 August 2009; pp. 141–145.
51. Shen, W.; Guo, X.; Wu, C.; Wu, D. Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm. *Knowl.-Based Syst.* 2011, 24, 378–385. [CrossRef]
52. Zhang, C.; Zhang, F.; Li, F.; Wu, H. (Eds.). Improved artificial fish swarm algorithm. In Proceedings of the IEEE 9th Conference on Industrial Electronics and Applications, Hangzhou, China, 9–11 June 2014.
53. Karaboga D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Tech. Report TR06; Erciyes University, Engineering Faculty, Computer Engineering Department: Melikgazi, Turkey, 2005.
54. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* 2008, 8, 687–697. [CrossRef]
55. Pérez-Delgado, M.L. The color quantization problem solved by swarm-based operations. *Appl. Intel.* 2019, 49, 2482–2514. [CrossRef]
56. Baykasoglu, A.; Özbakir, L.; Tapkan, P. Artificial bee colony algorithm and its application to generalized assignment problem. In *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*; Felix, T.S., Tiwari, M.K., Eds.; Itech Education and Publishing: Vienna, Austria, 2007; pp. 113–144.
57. Singh A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. Appl. Soft Comput. 2009, 9, 625–631. [CrossRef]

58. Karaboga, D.; Ozturk, C. A novel clustering approach: Artificial bee colony (ABC) algorithm. Appl. Soft Comput. 2011, 11, 652–657. [CrossRef]

59. Pan, Q.K.; Tasgetiren, M.F.; Suganthan, P.N.; Chua, T.J. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Inf. Sci. 2011, 181, 2455–2468. [CrossRef]

60. Szeto, W.Y.; Wu, Y.; Ho, S.C. An artificial bee colony algorithm for the capacitated vehicle routing problem. Eur. J. Oper. Res. 2011, 215, 126–135. [CrossRef]

61. Yang, X.S. (Ed.). Firefly algorithm for multimodal optimization. In Proceedings of the 5th International Symposium on Stochastic Algorithms: Foundations and Applications; LNCS 5792; Springer-Verlag: Berlin/Heidelberg, Germany, 2009; pp. 169–178.

62. Yang, X.S. Firefly algorithm, Lévy flights and global optimization. In Research and Development in Intelligent Systems XXVI; Bramer, M., Ed.; Springer-Verlag: London, UK, 2010; pp. 209–218.

63. Senthilnath, J.; Omkar, S.N.; Mani, V. Clustering using firefly algorithm: Performance study. Swarm Evol. Comput. 2011, 1, 164–171. [CrossRef]

64. Horng, M.H. Vector quantization using the firefly algorithm for image compression. Exp. Syst. Appl. 2012, 39, 1078–1091. [CrossRef]

65. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. Int. J. Swarm Intell. 2013, 1, 36–50. [CrossRef]

66. Fister, I.; Fister, I., Jr.; Yang, X.S.; Brest, J. A comprehensive review of firefly algorithms. Swarm Evol. Comput. 2013, 13, 34–46. [CrossRef]

67. Bonabeau, E.; Theraulaz, G. Role and variability of response thresholds in the regulation of division of labour in insect societies. In Information Processing in Social Insects; Detrain, C., Deneubourg, J.L., Pasteels, J.M., Eds.; Springer Verlag: Cham, Switzerland; 1999; pp. 141–163.

68. Rusu, M. Smart specialization a possible solution to the new global challenges. Procedia Econ. Financ. 2013, 6, 128–136. [CrossRef]

69. Baum, H.; Schütze, J. An organizational concept for collaborative enterprise networks. Procedia CIRP 2013, 7, 55–60. [CrossRef]

70. Wenseleers, T.; Ratnieks, F.; Billen, J. Caste fate conflict in swarm-founding social hymenoptera: An inclusive fitness analysis. Ecol. Biol. 2003, 16, 647–658. [CrossRef]

71. Mintzberg, H. El Diseño de las Organizaciones Eficientes; Editorial El Ateneo: Buenos Aires, Argentina, 1989.

72. Román, J.A.; Rodríguez, S.; Corchado, J.M.; Carrascosa, C.; Ossowski, S. Specialization: A new way to improve intelligent systems. Int. J. Artif. Intell. 2015, 13, 58–73.

73. Rushing, W. The effects of industry size and division of labor on administration. Admin. Sci. Q. 1967, 12, 273–295. [CrossRef]

74. Samuel, Y.; Mannheim, B. Multidimensional approach toward a typology of bureaucracy. Admin. Sci. Q. 1970, 15, 693–708. [CrossRef]

75. Reimann, B. Dimensions of structure in effective organizations: Some empirical evidence. Acad. Manag. J. 1974, 17, 693–708.

76. Arkin, R. Behavior Based Robotics; MIT Press: Cambridge, MA, USA, 1998.

77. Arkin, R.; Balch, T. Behavior-based formation control for multi-robot teams. IEEE Trans. Robot. Autom. 1999, 14, 926–939.

78. Balch, T. Measuring robot group diversity. In Robot Teams: From Diversity to Polymorphism; Balch, T., Parker, E., Eds.; A.K. Peters, Ltd.: Natick, MA, USA, 2002; pp. 93–135.

79. Chai, L.; Chen, J.; Han, Z.; Di, Z.; Fan, Y. (Eds.). Emergence of specialization from global optimizing evolution in a multi-agent system. In Proceedings of the 7th International Conference on Computer Science; Springer: Beijing, China, 2007; pp. 98–105.

80. Nitschke, G.; Schut, M.; Eiben, A. Emergent specialization in biologically inspired collective behavior systems. In Intelligent Complex Adaptive Systems; Yang, A., Ed.; IGI Publishing: New York, NY, USA, 2007; pp. 100–140.

81. Li, L.; Martinoli, A.; Abu-Mostafa, Y.S. Learning and measuring specialization in collaborative swarm systems. Adapt. Behav. 2004, 12, 199–212. [CrossRef]
82. O’Donnell, S.; Jeanne, R.L. Forager specialization and the control of nest repair in Polybia occidentalis Olivier (Hymenoptera: Vespidae). *Behav. Ecol. Sociobiol.* **1990**, *27*, 359–364. [CrossRef]
83. Alba, E. (Ed.). *Parallel Metaheuristics: A New Class of Algorithms*; Wiley-Interscience: Hoboken, NJ, USA, 2005.
84. Pedemonte, M.; Nesmachnow, S.; Cancela, H. A survey on parallel ant colony optimization. *Appl. Soft Comput.* **2011**, *11*, 5181–5197. [CrossRef]
85. Randall, M.; Lewis, A. A parallel implementation of ant colony optimization. *J. Parallel Distrib. Comput.* **2002**, *62*, 1421–1432. [CrossRef]
86. Gambardella, L.M.; Taillard, E.; Agazzi, G. *A Multiple Ant Colony System for Vehicle Routing Problem with Time Windows*; Tech. Report IDSIA-06-99; IDSIA: Lugano, Switzerland, 1999.
87. Salas, Y.C.; Pérez, N.C.; Betancur, J.F. (Eds.). Optimization based on multi-type ants for the Traveling Salesman Problem. In Proceedings of the 9th Computing Colombian Conference, Pereira, Colombia, 3–5 September 2014; pp. 144–149.
88. Zhang, Y.; Gong, D.W.; Ding, Z.H. Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer. *Exp. Syst. Appl.* **2011**, *38*, 13933–13941. [CrossRef]
89. Zhong, Y.B.; Xiang, Y.; Liu, H.L. A multi-objective artificial bee colony algorithm based on division of the searching space. *Appl. Intell.* **2014**, *41*, 987–1011. [CrossRef]
90. Yazdani, D.; Sepas-Moghaddam, A.; Dehban, A.; Horta, N. A novel approach for optimization in dynamic environments based on modified artificial fish swarm algorithm. *Int. J. Comput. Intell. Appl.* **2016**, *15*, 1–23. [CrossRef]
91. Michel, R.; Middendorf, M. An ACO algorithm for the shortest common supersequence problem. In *New Ideas in Optimization*; Corne, D., Eds.; McGraw-Hill: Maidenhead, UK, 1999; pp. 51–61.
92. Subotic, M.; Tuba, M. Parallelized multiple swarm artificial bee colony algorithm (MS-ABC) for global optimization. *Stud. Informa. Control* **2014**, *23*, 117–126. [CrossRef]
93. Dugonik, J.; Fister, I. Multi-population firefly algorithm. In Proceedings of the Computer Science Research Conference, Berlin, Germany, 5–6 November 2014; pp. 20–23.
94. Abshouri, A.A.; Meybodi, M.R.; Bakhtiary, A. New firefly algorithm based on multi swarm and learning automata in dynamic environments. In Proceedings of the Third International Conference on Signal Processing Systems (ICSPS 2011), Yantai, China, 27–28 August 2011; pp. 73–77.
95. Niu, B.; Zhu, Y.; He, X.; Wu, H. MCPSO: A multi-swarm cooperative particle swarm optimizer. *Appl. Math. Comput.* **2007**, *185*, 1050–1062. [CrossRef]
96. Zhu, K.; Jiang, M.; Cheng, Y. Niche artificial fish swarm algorithm based on quantum theory. In Proceedings of the IEEE 10th International Conference on Signal Processing, Beijing, China, 24–28 October 2010; pp. 1425–1428.

© 2020 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/).