



Heriot-Watt University  
Research Gateway

# A novel Physarum-inspired competition algorithm for discrete multi-objective optimisation problems

## Citation for published version:

Awad, A, Coghill, GM & Pang, W 2023, 'A novel Physarum-inspired competition algorithm for discrete multi-objective optimisation problems', *Soft Computing*, vol. 27, no. 20, pp. 14699-14719.  
<https://doi.org/10.1007/s00500-023-08505-1>

## Digital Object Identifier (DOI):

[10.1007/s00500-023-08505-1](https://doi.org/10.1007/s00500-023-08505-1)

## Link:

[Link to publication record in Heriot-Watt Research Portal](#)

## Document Version:

Publisher's PDF, also known as Version of record

## Published In:

Soft Computing

## Publisher Rights Statement:

© The Author(s) 2023.

## General rights

Copyright for the publications made accessible via Heriot-Watt Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

## Take down policy

Heriot-Watt University has made every reasonable effort to ensure that the content in Heriot-Watt Research Portal complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [open.access@hw.ac.uk](mailto:open.access@hw.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# A novel Physarum-inspired competition algorithm for discrete multi-objective optimisation problems

Abubakr Awad<sup>1</sup> · George M. Coghill<sup>2</sup> · Wei Pang<sup>3</sup>

Accepted: 24 April 2023  
© The Author(s) 2023

## Abstract

Many real-world problems can be naturally formulated as discrete multi-objective optimisation (DMOO) problems. We have proposed a novel Physarum-inspired competition algorithm (PCA) to tackle these DMOO problems. Our algorithm is based on hexagonal cellular automata (CA) as a representation of problem search space and reaction–diffusion systems that control the Physarum motility. Physarum’s decision-making power and the discrete properties of CA have made our algorithm a perfectly suitable approach to solve DMOO problems. Each cell in the CA grid will be decoded as a solution (objective function) and will be regarded as a food resource to attract Physarum. The n-dimensional generalisation of the hexagonal CA grid has allowed us to extend the solving capabilities of our PCA from only 2-D to n-D optimisation problems. We have implemented a novel restart procedure to select the global Pareto frontier based on both personal experience and shared information. Extensive experimental and statistical analyses were conducted on several benchmark functions to assess the performance of our PCA against other evolutionary algorithms. As far as we know, this study is the first attempt to assess algorithms that solve DMOO problems, with a large number of benchmark functions and performance indicators. Our PCA has confirmed our assumption that individual skills of competing Physarum are more efficient in exploration and increase the diversity of the solutions. It has achieved the best performance for the Spread indicator (diversity), similar performance results compared to the strength Pareto evolutionary algorithm (SPEA2) and even outperformed other well-established genetic algorithms.

**Keywords** Discrete multi-objective optimisation · Physarum · Competition · Bio-inspired algorithms · Evolutionary algorithms

## 1 Introduction

Many real-world optimisation problems are multi-objective in nature, which means that the optimal decisions need to be taken in the presence of multiple conflicting objectives

(Deb 2014). Most of the existing research has focused on continuous multi-objective optimisation (CMOO) problems (Zopounidis and Pardalos 2010). However, DMOO problems with discrete variables and black-box objective functions are frequently encountered in a diverse set of disciplines such as engineering (Han et al. 2020), network models (Bergman et al. 2018) transportation (Ghannadpour and Zarrabi 2019), supply chain (Dickersbach 2005), medicine (Wang and Ma 2018) and many combinatorial problems (Zouache et al. 2018). For more details, the reader is kindly requested to go through the survey papers (Liu et al. 2020; Deb 2014; Ehrgott 2006; Ehrgott et al. 2016). Despite the finite number of objective functions and the relative finite search space in these DMOO problems, efficient algorithms are still lacking. The challenge of developing efficient algorithms arises from the discontinuity of objective functions and the discreteness of the optimal solution set.

---

✉ Abubakr Awad  
abubakr.awad@nottingham.ac.uk ; bakr.awad@gmail.com

George M. Coghill  
g.coghill@abdn.ac.uk

Wei Pang  
w.pang@hw.ac.uk

- <sup>1</sup> School of Computer Science, University of Nottingham, Nottingham, UK
- <sup>2</sup> School of Natural and Computing Sciences, University of Aberdeen, Aberdeen, UK
- <sup>3</sup> School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK

The inspiration from biology and nature has been one of the most important and exhaustless sources to develop novel algorithms and innovative techniques to solve decision-making problems over the past decades (Afeq et al. 2011; Kroeker 2011; Yang 2010). Computer scientists are taking inspiration from Physarum intelligent foraging behaviour to come up with many novel bio-inspired models capable of solving many NP-hard problems (Adamatzky 2010).

Swarm intelligence is one of the most interesting approaches for solving multi-objective optimisation (MOO) problems (Mostaghim and Teich 2013; Nebro et al. 2009; Alaya et al. 2007; Zou et al. 2011). These approaches deal with the collective behaviour of decentralised and self-organised systems. Just like social insects and animals, Physarum too exhibits swarm intelligence; it shares many features of collective behaviour such as synchronisation, communication, positive feedback, leadership and response thresholds (Reid and Latty 2016). In many optimisation problems, it is crucial to achieve an optimal balance between social collaboration and population diversity (Wong et al. 2010). A major issue that may rise in some swarm intelligence algorithms is the premature convergence as they focus on social collaboration within the population, and they seldom pay attention to the competition among individuals (Beni and Wang 1993). We believe that the individual skills of competitors are more efficient to achieve a good balance between exploration and exploitation and are an important approach to maintain population diversity and the effectiveness of the search.

We have proposed a novel model to imitate the complex patterns observed in Physarum generated in competition settings (Awad et al. 2019a). This new model is based on hexagonal cellular automata (CA) and reaction–diffusion (RD) systems. Physarum’s decision-making power and the discrete properties of CA have made our algorithm a perfectly suitable approach in solving DMOO problems.

Physarum certainly meets the criteria of an ideal model of decision-maker, as it makes multi-objective foraging decisions (Beekman and Latty 2015). Similar to those organisms that have a central information processing unit like a brain, Physarum balances its nutrient intake based on optimality theory. Physarum is capable of choosing the source that is high in concentration and away from dangerous environment, and it trades off risk against food quality (Latty and Beekman 2010). For searching in a discrete space, CA was a very successful choice. CA can represent a physical system, in which space and time are discrete. The physical grid quantities can take only a finite set of values and any finite number of dimensions. This has motivated us to develop a novel Physarum-inspired optimisation algorithm for DMOO problems. This algorithm is based on the possible heuristics that Physarum uses in complex foraging decisions in competition settings. Instead of using Physarum to optimise swarm algorithms, we explore the potential of competition among

individual Physarum acting as a swarm to accelerate the algorithm’s convergence, guarantee the diversity of solutions and increase the ability to escape from local optima.

## 1.1 Work novelty and contributions

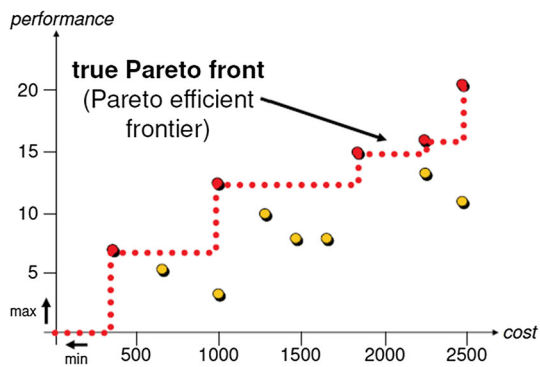
In this research, we proposed a novel autonomous Physarum-inspired competition algorithm (PCA). The competition among the different Physarum individuals has extended the search (exploration) from a single Physarum as in previous Physarum-inspired algorithms to a population-based search as in swarm algorithms, but keeping individual skills. This has facilitated global exploration and kept population diversity. We have considered the  $n$ -dimensional generalisation of the hexagonal grids to extend the solving capabilities of our PCA from only two-dimensional to multi-dimensional optimisation problems. PCA has achieved the best performance for the Spread indicator, inverted generational distance plus (IGD+), the second-best values for the HV and Epsilon indicators after SPEA2, and it even outperformed other well-established genetic algorithms. We have implemented a novel restart procedure which improved the overall performance as it drives the exploitation towards the best solution based on both personal experience and shared information. One of the interesting points in this research is a large number of benchmark functions and the diverse set of performance metrics used.

In this paper, a short overview of background concepts and related technologies, for a better understanding of DMOO problems, is presented in Sect. 2. We will then go further to show how we developed a novel Physarum competition algorithm (PCA) capable of solving DMOO problems (Sect. 3). We have conducted extensive experimental and statistical work on several benchmark functions to assess the performance of our PCA against state-of-the-art evolutionary algorithms (Sect. 4). First, we will provide the experimental results of deep valley problem, which is a discrete 2-D optimisation problems. Second, we will consider the  $n$ -D generalisation of the hexagonal grids to extend the solving capabilities of our PCA from only 2-D to  $n$ -D optimisation problems. Finally, we will give a conclusion of this research in Sect. 5.

## 2 Background concepts and related technologies

### 2.1 Background concepts

The concepts of MOO problems have been defined by many researchers (Deb et al. 2000). However, in this research, we will focus on DMOO problems (Fig. 1). DMOO problems are concerned with a class of mathematical optimisation prob-



**Fig. 1** In this DMOO problem, we are trying to maximise the performance and at the same time minimise the cost. The red dots are the non-dominated solution (Pareto front) while the yellow dots are example of solutions dominated by red ones (this figure is based on Brockhoff (2018) with some modifications)

lems with discrete variables and multiple objectives to be optimised simultaneously. It is hard to optimise objectives simultaneously because they are conflicting. These problems can be mathematically formalised as follows:

$$\begin{aligned}
 &\text{minimise } F(X) = (f_1(X), \dots, f_m(X)), \\
 &\{F : X \rightarrow R^m, m \geq 2\} \\
 &\text{subject to } G(X) = (g_1(X), \dots, g_k(X)), \\
 &\{G : X \rightarrow R^k, G_k(X) \geq 0, k \geq 0\} \\
 &H(X) = (h_1(X), \dots, h_l(X)), \\
 &\{H : X \rightarrow R^l, H_l(X) = 0, l \geq 0\} \\
 &\text{and } X = (x_1, \dots, x_n), \{X \subset Z^n\}
 \end{aligned} \tag{1}$$

In the above equation,  $X$  is an  $n$ -dimensional decision variable (vector) from a discrete (Integer) space.  $F(X)$  is an objective vector composed of  $m$  real-valued objective functions which are either continuous or discrete.  $G(X)$  and  $H(X)$  are the constraint vectors, which consist of real-valued functions representing inequality and equality constraints, respectively.

For more illustration, we have to be familiar with the following concepts:

**Definition 1** (Decision Variables). Decision variables are controllable options that need to be determined in order to solve a problem. The solutions for the optimised problem are generated from the search space of each decision variable. The problem is solved when the best values of the decision variables have been identified.

**Definition 2** (Objective Functions). The objective function of an optimisation problem indicates how much each decision variable contributes to the value to be optimised in the problem.

**Definition 3** (Constraints). Constraints define the possible values that the decision variables of an optimisation problem must be fulfilled while optimising (minimising or maximising)  $F(X)$ . They typically represent resource constraints, or the minimum or maximum levels of an activity.

**Definition 4** (Pareto Dominance). From the problem definition 1, let  $a = (a_1, \dots, a_m)$ ,  $b = (b_1, \dots, b_m)$  be two solution vectors obtained from a decision variable (vector).  $a$  is said to dominate  $b$  if  $a_i \leq b_i \forall i = 1, \dots, m$  and  $a \neq b$

## 2.2 Related work

The optimal solution for MOO problems is often not a single one, but rather a set of solutions known as the Pareto optimal set (or Pareto frontier). In this set, no element is superior to the others for all the objectives. Most of the multi-objective optimisation algorithms are based on the theories of Pareto Sort and non-dominated solutions (dominance-based) (Hillermeier 2001). There are two main sets of approaches for generating the Pareto frontier for a MOO problem. The first set of approaches are the criterion-space search methods, which rely on a combination of weighted sums and constraints scalarisations (Kirlik and Sayin 2014; Bektaş 2018) or the augmented weighted Tchebycheff scalarisation (Holzmann and Smith 2018). The second is the decision-space search methods, which are typically based on branch-and-bound search. These methods are developed for mixed-integer linear programmes that operate over the space defined by the original decision variables (Mavrotas and Diakoulaki 1998; Adalgren and Gupta 2017).

Algorithms for solving MOO problems include weighted sum (Audet et al. 2008), continuation methods (Schütze et al. 2008) and evolutionary algorithms (Deb 1999). Multi-objective evolutionary algorithms (MOEAs) are efficient approaches for solving MOO problems (Li et al. 2015). Two of the most well-known MOEAs are the non-dominated sorted genetic algorithm (NSGA-II) (Deb et al. 2000) and the strength Pareto evolutionary algorithm (SPEA2) (Zitzler et al. 2001). GA is based on natural selection and evolution. GA is a very powerful optimisation algorithm in many scientific fields. A continuous genetic algorithm proved to be an efficient solver for systems of second-order boundary value problems (Arqub and Abo-Hammour 2014). It involves initialisation, evaluation, selection, crossover, mutation, replacement and termination. An elegant explanation can be reviewed from Abo-Hammour et al. (2013). SPEA2 contains important operations such as archiving of individuals with good fitness, density estimation and fitness assignment and is able to obtain a population with both “precision” and “diversity”. SPEA2+ attempts to improve the problem space exploration abilities of SPEA2 by adding a more effective crossover mechanism and an algorithm to

maintain diversity in the two object and variable spaces (Zitzler et al. 2001).

These two classical algorithms perform relatively well on MOO problems when the number of objective functions is not more than three. However, their performance significantly degrades for MOO problems when the number of objective functions is greater than four, in which case it becomes many-objective optimisation problems (Ishibuchi et al. 2008). More recent studies have proposed improved variants of dominance-based MOEAs that use reference vector-based niching selection rather than the crowding distance-based selection. NSGA-III (Deb and Jain 2014) is an example of such algorithms. Relaxed-dominance-based MOEAs that are used for the mating and/or environmental selections instead of the exact Pareto-dominance relationship have also been proposed. MOEAs based on  $\epsilon$  dominance ( $\epsilon$ -MOEAs) (Deb et al. 2005) and differential evolution (Arunachalam 2008) are examples of such algorithms.

Most of the above-mentioned techniques were designed for solving CMOO problems (Zopounidis and Pardalos 2010); however, DMOO problems are frequently encountered in a diverse set of engineering and scientific domains (Ehrgott 2006; Zhang et al. 2008). Some existing state-of-the-art methodologies rely on parametric single-objective reformulations of DMOO problems, employing commercial integer programming (IP) solvers as black-box tools for their algorithms (Audet et al. 2008). However, the Pareto frontier cannot be fully recovered by such linear parametric methods in general. Recent studies have tried to address the DMOO problems without tailoring the CMOO algorithm. However, the majority of them are specialised algorithms targeting specific types of DMOO problems (Dickersbach 2005; Gao et al. 2018; Wang 2015; Zouache et al. 2018; Peng et al. 2016; Bergman et al. 2018) and can only solve discrete bi-objective optimisation problems (Boland et al. 2015; Parragh 2018), while only a few of them deal with more than two objective functions (Cacchiani and D'Ambrosio 2017).

Swarm intelligence algorithms are one of the most popular and important approaches for solving multi-objective problems. Examples of swarm-inspired algorithms include particle swarm optimisation (PSO) (Mostaghim and Teich 2013; Nebro et al. 2009), ant colony optimisation (ACO) (Alaya et al. 2007) and artificial bee colony (ABC) (Zou et al. 2011). Recently, Physarum-inspired algorithms have been used for optimising the pheromone matrix initialisation in multi-objective ACO for solving bi-objective travelling salesman problems (Zhang et al. 2016b). The work of Masi and Vasile (2014) was the first attempt to introduce a Physarum-inspired algorithm for DMOO problems. Their algorithm starts by transforming a given DMOO problem into a graph problem, which can be solved as in multi-objective

travelling salesman and vehicle routing problems. However, this approach cannot be generalised well when dealing with generic search problems rather than graph optimisation problems.

### 3 Physarum-inspired competition algorithm for solving DMOO problems

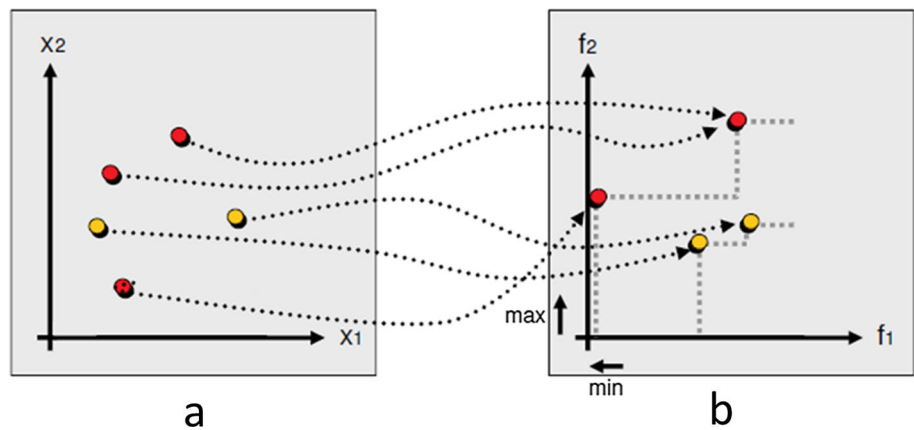
In this section, we present a novel Physarum-inspired competition algorithm (PCA) for DMOO problems. The present algorithm is inspired from our Physarum competition mathematical model (Awad et al. 2019a). This model simulates the foraging behaviour of multiple Physarum competing for multiple food resources, where the Physarum is presented in a cellular automata (CA) grid and the Physarum motility is based on reaction–diffusion (RD) system applied on CA. The Physarum motility over a hexagonal cellular automaton will allow a discrete and effective search. Each cell in the grid will be decoded as a solution (objective function) and will be regarded as a food resource to attract Physarum (Fig. 2).

The algorithm works by identifying the non-dominated solutions for an optimisation problem using the possible heuristics that Physarum uses in complex foraging decisions. The competing Physarum will explore the search space according to the diffusion equations based on the chemoattraction forces towards food resources, and the repulsion negative forces that competing Physarum exert on each other. Physarum will diffuse maximally to cells that provide the best possible trade-offs between the objective functions.

From biology, we understand that Physarum foraging behaviour consists of two simultaneous self-organised processes: expansion (exploration) and contraction (exploitation) (Liu et al. 2017). From the algorithm perspective, the balance between exploration and exploitation process is fundamental for solving DMOO problems (Sun et al. 2018). In the Physarum exploration phase, the competing Physarum will explore the search space according to diffusion equations. In the Physarum exploitation phase, the protoplasmic flow in the body of Physarum plays a great role in developing its intelligence (Tero et al. 2005; Alim et al. 2013). The protoplasmic flux connected to high-quality food resources (non-dominated solutions) tends to increase, while the protoplasmic flux to poor-quality food resources (dominated solutions) tends to decrease as a feedback mechanism (Reid and Latty 2016). This behaviour can be interpreted as a natural attitude in optimising the energy required to feed the organism. The reader can refer to Awad et al. (2021) for more details on Physarum intelligent foraging behaviour in competition settings.



**Fig. 2** **a** Physarum search space is mapped to the **b** objective space, in which each cell in the search space is decoded as a food resource to attract Physarum (this figure is based on Brockhoff (2018) with modifications)



### 3.1 Rules of Physarum-inspired competition algorithm

In order to build our algorithm for solving DMOO problems, we have proposed the following rules:

- (i) The discrete search space of the optimisation problem will be mapped to an  $n$ -dimensional hexagonal CA to solve multi-dimensional problems (Sect. 3.2).
- (ii) Multiple Physarum will be randomly deployed over the search space using either Monte Carlo or Latin Hyper Cube sampling techniques (Sect. 3.4).
- (iii) In the exploration phase, Physarum will explore the search space according to the diffusion mechanism (Eq. 4). We also introduced a feature that maps to real biological behaviour, where the Physarum will stop diffusing if its mass is less than a critical value. This will prevent the Physarum from searching the entire search space in an overly greedy manner.
- (iv) Each cell in the grid will be decoded as a solution and will be regarded as a food resource to attract Physarum. The chemo-attraction forces exerted on Physarum will be a function of the quality of the food resource, which in this case is the degree of optimality of the solution to fulfil the objectives of the problem (Eq. 8).
- (v) Competing Physarum will exert repulsion forces on each other, which will be calculated as a negative force. When two Physarum are competing for the same cell, the one with higher mass will occupy this cell (Eq. 9).
- (vi) The cells that represent food resources engulfed by a Physarum (decoded solutions) will be excluded from further search using Physarum spatial memory (Reid et al. 2012). This memory will increase the searching efficiency and avoid visiting previously explored areas (Reid et al. 2012, 2013; Boussard et al. 2019).
- (vii) The cells that represent unfeasible solutions and do not satisfy the constraint set, will be regarded as walls (obstacles), and the Physarum will stop searching these cells.

- (viii) The procedure of exploring the search space by the competing Physarum will stop when one of the two conditions is satisfied: first when the competing strands of Physarum are unable to find new dominant solutions due to Physarum mass exhaustion (i.e. the mass of Physarum is less than a critical value) and second after a predefined number of iterations is reached.

### 3.2 Hexagonal CA grids

We have considered the Physarum motility over a 2-D, 3-D and  $n$ -D hexagonal cellular automata (Fig. 3), where a set of Physarum ( $P = p_1, p_2 \dots p_m$ ) are competing on a set of chemicals (food resources) ( $\text{CHM} = \text{chm}_1, \text{chm}_2 \dots \text{chm}_n$ ). The state of a cell  $c_{(i,j)}^t$  at iteration  $t$  is described by its type (Eq. 2).

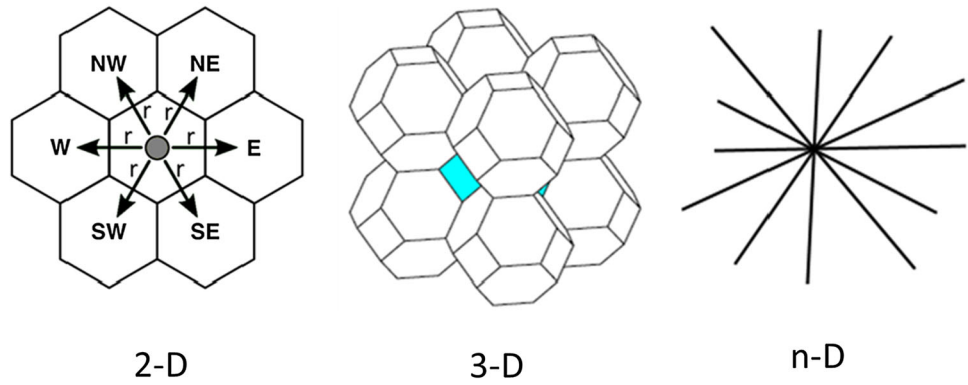
$$\text{CT}_{(i,j)} = \{ \text{"FREE"}, \text{"OBSTACLE"}, \text{"PHYSARUM"}, \text{"CHEMICAL"} \} \quad (2)$$

where  $\text{CT}_{(i,j)}$  represents the cell type.

In the two-dimensional hexagonal CA, the grid is divided into a matrix ( $X \times Y$ ) of identical hexagonal cells, and each cell in the grid has six neighbours. Hexagonal/face-centred cubic (*fcc*) grids and hexagonal/body-centred cubic (*bcc*) grids are the three-dimensional "equivalence" of the two-dimensional hexagonal grids. For the *bcc* grid, the Voronoi neighbourhoods are truncated octahedra having eight hexagonal and six square faces. The space volume was dynamically tessellated by regular truncated octahedron voxels. The hexagonal grid has the densest packing. The truncated octahedron voxels are more "sphere-like", and they have the highest volumetric quotient (the ratio of the polyhedron volume to the volume of its circumsphere) which is much higher than other possible space-filling polyhedrons.

For the  $n$ -dimensional generalisation of the hexagonal and *bcc* grids (Nagy and Strand 2009), the hexagonal faces neighbourhood is defined as follows:

**Fig. 3** Hexagonal neighbourhood of different dimensions



**Definition 5** Let  $\mathbf{p} = (p(1), p(2), \dots, p(n))$  and  $\mathbf{q} = (q(1), q(2), \dots, q(n))$  be two points in  $Z^n$ .  $\mathbf{q}$  is 1-neighbour to  $\mathbf{p}$  if  $q(1) \equiv q(2) \equiv \dots \equiv q(n) \pmod{2}$  and  $|p(i) - q(i)| \leq 1$  for  $1 \leq i \leq n$  and  $\sum_{i=1}^n (p(i) - q(i))^2 \leq n$ .

The position vector of a Physarum in the high-dimensional grid will be defined as follows:

$$P_i^t = [p_{i1}^t, p_{i2}^t, \dots, p_{in}^t], \quad i = 1, 2, \dots, n_p, \quad (3)$$

where  $i$  represents the  $i^{th}$  Physarum individual,  $t$  is the current time step,  $n$  is the dimensionality of the problem space, and  $n_p$  is the population size.

**3.3 Reaction–Diffusion equations (RD)**

The RD equations are applied on Physarum’s mass. It defines the exploration of the available area in the search space by the cytoplasmic material formed by the Physarum plasmodium. We combined both the chemo-attraction force towards chemicals and the repulsion negative forces that competing Physarum exert on each other. Every cell occupied by Physarum at time step ( $t$ ) uses the values of its neighbour cells to calculate the value of the mass at time step ( $t + 1$ ). These rules are executed until a certain number of iteration reached. The Physarum diffusion is affected by the presence of food resources (objective functions) and other competitors as seen in the following equation:

$$PM_i^{t+1} = PM_i^t + \sum_{P_k \in NB_{P_i}} \left\{ \begin{array}{l} (PF * PD)(PM_k^t - PM_i^t) \text{ if } AA_{P_i \rightarrow P_k} = 1, \\ PM_i^t > Diff\_Thr \\ 0, \text{ otherwise} \end{array} \right\}. \quad (4)$$

The variables in Eq. (4) are explained as follows:

- $PM_i^{t+1}$  defines the diffusion of Physarum  $P_i$  mass for the next iteration ( $t + 1$ );
- $PM_i^t$  is the current mass of Physarum  $P_i$  for iteration ( $t$ );
- $PM_k^t$  is the current mass of neighbouring Physarum  $P_k$ ;
- $NB_{P_i}$  is the neighbourhood of Physarum  $P_i$  as in Definition 5;
- PF is the sum of forces affecting Physarum diffusion (defined in Eq. 6);
- PD is the Physarum diffusion coefficient (defined in Eq. 5);
- $AA_{P_i \rightarrow P_k}$  indicates whether Physarum  $P_i$  is available to diffuse towards a neighbouring Physarum  $P_k$  (defined in Eq. 7);
- Diff\_Thr is the threshold in which the Physarum mass must exceed in order to diffuse.

Now, we define Eqs. (5)–(10) as follows:

$$PPD = \frac{nbSize}{10^n} \quad (5)$$

In Eq. (5), nbSize is the neighbourhood’s size of Physarum  $P_i$  (Definition 5) and  $n$  is the dimensionality of the problem space.

$$PF = 1 + AttForce_{P_i \rightarrow P_k} + RepForce_{P_i \leftarrow P_k} \quad (6)$$

In Eq. (6),  $AttForce_{P_i \rightarrow P_k}$  defines the value of attraction forces applied on Physarum  $P_i$  coming from its neighbouring Physarum  $P_k$  (defined in Eq. 8) and  $RepForce_{P_i \leftarrow P_k}$  defines the value of repulsion forces applied on Physarum  $P_i$  exerted by its neighbouring Physarum  $P_k$  (defined in Eq. 9).

$$AA_{P_i \rightarrow P_k} = \begin{cases} 1, & \text{if } P_k \text{ is Empty} \\ 1, & \text{if } PID_i = PID_k \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

In Eq. (7),  $PID_i$  is the ID of Physarum  $P_i$ .

$$AttForce_{P_i \rightarrow P_k} = \sum_{f_i}^{F(X)} \left\{ \frac{f_i \cdot Max - f_i(P_k)}{f_i \cdot Max - f_i \cdot Min} \right\} \quad (8)$$

In Eq. (8),  $F(X)$  is the objective functions and  $f_i \cdot Min$  and  $f_i \cdot Max$  are the local minimal and maximal values obtained by Physarum search for an objective function  $f_i$ , respectively.

$$RepForce_{P_i \leftarrow P_k} = \begin{cases} \frac{PM_{opp(k \Rightarrow i)}}{Total\_Physarum\_Mass} & \text{if } PID_i \neq PID_{opp(k \Rightarrow i)}, \\ 0, & PM_{opp(k \Rightarrow i)} > Rep\_Thr \\ & \text{otherwise} \end{cases} \quad (9)$$

In Eq. (9),  $PM_{opp(k \Rightarrow i)}$  is the neighbour Physarum mass at the opposite direction of Physarum  $P_k$  with reference to the centre point  $c_i$  as defined in Eq. (10) and  $Rep\_Thr$  is a threshold where Physarum must reach to repel neighbouring Physarum.

$$opp_{(k \Rightarrow i)} = c_{2i-k} \quad (10)$$

In Eq. (10),  $opp_{(k \Rightarrow i)}$  is the opposite direction of cell  $c_k$  with reference to the centre point  $c_i$ .

The main steps of the PCA are shown in Algorithm 1.

### 3.4 Physarum deployment strategy

In the Physarum exploration phase, the initial deployment of Physarum over the CA grids may directly affect the search performance, especially when the search space of the optimisation problem is huge (Liu et al. 2015). Most of the swarm-inspired algorithms use random deployment strategy [Monte Carlo sampling (McKay et al. 1979)]. However, in our algorithm design, we used a random deployment strategy (Monte Carlo sampling) and introduced a distributed regional deployment strategy, by using Latin Hypercube sampling technique (LHS) (Lee et al. 2006). LHS is a statistical method used to generate parameter values from a multi-dimensional distribution. In our previous work, LHS has proven to be an effective approach for initialising population (Tian et al. 2019; Usman et al. 2020). A square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. By using LHS, we ensure that the multiple Physarum are evenly distributed all over the search space and thus maximising the coverage.

### Algorithm 1: Physarum-Inspired Competition Algorithm

```

1 GPF = {}; // Initialise empty global Pareto front
  /* Initialise Physarum (solutions)
   {S = P1, P2, ... , Pnp} */
2 foreach Physarum (Pi) do
3   Pi.mass ← init_pm // Initialise Physarum mass
4   Pi.lpf ← {} // Initialise empty local Pareto front for Physarum
5   Randomly place Physarum (position vector) Pi in the search space using either Monto Carlo or Latin Hypercube sampling techniques.
6 end
7 current-iter = 0
8 while current-iter < max-iter do
9   foreach Physarum (Pi) do
10    Add Physarum Pi (solution) to it's LPF (If not dominated by other solutions in Physarum LPF)
    // Initialise Physarum Neighbourhood
11    foreach Physarum Pk in Physarum Pi Neighbourhood ;
    // Definition 5
12    do
13      if PAA(Pi)→(Pk) = 1 ; // Equation 7
14      then
15        Add a new Physarum (Pk) to the search space with initial mass 0
16      end
17    end
18  end
19  foreach Physarum (Pi) do
20    Diffuse Pit to the neighbouring cells ; // Equation 4
21  end
22  if USING RESTART PROCEDURE then
23    if iteration % restart-iteration = 0 then
24      Compute new GPF from each Physarum LPF.
25      Clear old Physarum (Solutions) and create others from this new GPF (as in line 2)
26    end
27  end
28  current-iter = current-iter + 1
29 end
30 Compute GPF from each Physarum LPF.

```

### 3.5 External archive of the non-dominated solutions set

In this research, each Physarum has its own external archive to maintain its local best non-dominated solutions, i.e. local Pareto front (LPF). We use the Pareto dominated comparison mechanism (Reyes-Sierra and Coello 2006) to prune/update this archive. The Pareto dominated comparison mechanism works by comparing the newly decoded solution at  $t_{n+1}$  with the archive at iteration  $t_n$  using the Pareto domination relationship in order to select all non-dominated solutions. These non-dominated solutions are saved into the new archive at iteration  $t_{n+1}$  by concatenating/overwriting the previous archive at iteration  $t_n$ . In our algorithm design, the archive



will initially contain the Physarum initial position as a non-dominated solution. Afterwards, during the Physarum exploration phase, each time a new solution is created (corresponding to a new position of Physarum); it will be added to or discarded from the archive using the Pareto dominated comparison mechanism.

In most of the previous studies, the performance of the MOEAs was evaluated using bounded external archive (BEA) based on non-dominated solutions in the final population at the end of the search (Luong and Bosman 2012). In this research, we have used an unbounded external archive (UEA), which stores all non-dominated solutions found during the search process. This method has been proved to be more practical in real-world applications than the final population method (Tanabe et al. 2017).

### 3.6 Physarum-inspired competition algorithm with a restart mechanism

We introduced a new approach with a restart mechanism (Algorithm 1, line 22). In our algorithm design, after a predetermined number of iterations (*restart-iter*), multiple Physarum will share their personal experiences (i.e. personal bests), which can be interpreted as a communication mechanism among them (Reid and Latty 2016). These local non-dominated solutions from each Physarum archive will be compared together (using the Pareto dominated comparison mechanism) to select the global Pareto frontier (GPF). The cells identified as part of the global Pareto frontier in the external archive will be selected as the new Physarum population. These new Physarum will have their original Physarum's mass which will grow and resume exploration, while the others will potentially vanish. This restart mechanism can avoid stagnation on local optima and drive exploration towards the best solution positions based on both personal experience and shared information.

## 4 Experiments

We have implemented our PCA using Java, and we have used jMetal<sup>1</sup> as a multi-objective optimisation framework for our experiments (Durillo and Nebro 2011). All experiments were carried out on the Maxwell high-performance computing cluster from the University of Aberdeen.

### 4.1 Benchmark functions

A small number of DMOO benchmark functions have been proposed in recent years (multiple-criteria decision-making (MCDM), <http://www.mcdmsociety.org/content/digital-library>;

<sup>1</sup> <http://jmetal.sourceforge.net> (version 5.5).

Mittelmann 2018), and they are only for multi-objective combinatorial problems. To overcome this problem, we attempted to get benefit from the rich pool of CMOO benchmark functions and to transform them into DMOO problems.

One solution for this transformation is the direct discretisation of a CMOO problem's continuous search space. This is achieved by partitioning the numeric continuous variables into several sub-ranges as in Elomaa and Rousu (2004); Riquelme et al. (2015); Abo-Hammour et al. (2014). The goal is to find a set of cut points to partition the range into a small number of intervals, thus transforming the continuous-valued attributes into a finite number of intervals. As an example of this method, assume we have a continuous search space of a certain objective function, and the values of variables are within the range of [0,1]. We will partition (discretise) this search space to equal discrete intervals (DI) of 0.01 (0, 0.01, ..., 0.99, 1). By using this method, the problem will be compatible with the discretised motility of Physarum over the CA grids as described in the Physarum competition mathematical model (Awad et al. 2019a) and in the biological experiments of Shirakawa et al. (2015).

To generate the true Pareto frontier of our DMOO problems, we used the decision-space search method (Mavrotas and Diakoulaki 1998; Adalgren and Gupte 2017). This method is typically based on the branch-and-bound search strategy developed for multi-objective integer linear programming problems.

In our study, we have selected a wide range of 38 benchmark functions with different characteristics and parameter settings such as DTLZ (Deb et al. 2002), GLT (Zhang et al. 2016a), MOP (Deb 1999) and ZDT (Zitzler et al. 2000). The list of benchmark functions describing the parameters, upper and lower bounds and the discretisation interval is shown in Tables 1 and 2, where  $d$ ,  $m$  and  $c$  indicate the number of decision variables, objectives and constraints, respectively.

### 4.2 Performance assessment

As we are proposing a new optimisation algorithm, it is necessary to compare the performance of our PCA against other MOO algorithms. Assessing the performance of different MOO algorithms usually requires two properties: convergence and diversity. A number of quality indicators for performance assessment have been proposed in literature (Riquelme et al. 2015).

In our study, we have selected Epsilon ( $\epsilon$ ) (Zitzler et al. 2003), which is an indicator for measuring convergence, Spread ( $\Delta$ ) (Li and Zheng 2009), which is an indicator for measuring diversity, and hypervolume (HV) (Zitzler and Thiele 1999) and inverted generational distance plus (IGD+) (Veldhuizen and Lamont 2000; Ishibuchi et al. 2016), which are indicators for measuring both convergence and diversity.

**Table 1** The characteristics of the selected benchmark functions

Problem ID	DMOOProblems	$d$	$m$	$c$	Discrete variables bounds	Discrete interval (DI)	References
1	ConstrEx	2	2	2	$10 < x_0 < 100$ $100 < x_1 < 500$	$DI(x_0) = 0.01$ $DI(x_1) = 0.01$	Deb et al. (2000)
2	DTLZ4	5	3	0	$0 < x_n < 10$	$DI(x_n) = 0.1$	Deb et al. (2002)
3	DTLZ4	2	3	0	$0 < x_n < 100$	$DI(x_n) = 0.01$	
4	DTLZ4	3	3	0	$0 < x_n < 50$	$DI(x_n) = 0.05$	
5	Fonseca	3	2	0	$-40 < x_n < 40$	$DI(x_n) = 0.1$	Fonseca and Fleming (1995)
6	GLT1	2	2	0	$0 < x_0 < 100$ $-100 < x_1 < 100$	$DI(x_0) = 0.01$ $DI(x_1) = 0.01$	Zhang et al. (2016a)
7	GLT2	2	2	0	$0 < x_0 < 100$ $-100 < x_1 < 100$	$DI(x_0) = 0.01$ $DI(x_1) = 0.01$	
8	GLT3	2	2	0	$0 < x_0 < 100$ $-100 < x_1 < 100$	$DI(x_0) = 0.01$ $DI(x_1) = 0.01$	
9	GLT4	2	2	0	$0 < x_0 < 100$ $-100 < x_1 < 100$	$DI(x_0) = 0.01$ $DI(x_1) = 0.01$	
10	GLT5	3	2	0	$0 < x_0 < 50$ $0 < x_1 < 50$ $-50 < x_2 < 50$	$DI(x_0) = 0.05$ $DI(x_1) = 0.05$ $DI(x_2) = 0.05$	
11	Golinski	7	2	11	$26 < x_0 < 36$ $7 < x_1 < 8$ $170 < x_2 < 280$ $73 < x_3 < 83$ $73 < x_4 < 83$ $29 < x_5 < 39$ $50 < x_6 < 55$	$DI(x_0) = 0.1$ $DI(x_1) = 0.1$ $DI(x_2) = 0.1$ $DI(x_3) = 0.1$ $DI(x_4) = 0.1$ $DI(x_5) = 0.1$ $DI(x_6) = 0.1$	Ray (2003)
12	MOP1	5	2	0	$0 < x_n < 10$	$DI(x_n) = 0.1$	Deb (1999)
13	MOP2	5	2	0	$0 < x_n < 10$	$DI(x_n) = 0.1$	
14	MOP3	5	2	0	$0 < x_n < 10$	$DI(x_n) = 0.1$	
15	MOP4	5	2	0	$0 < x_n < 10$	$DI(x_n) = 0.1$	

For Epsilon and IGD+ indicators, the lower the value, the better the computed Pareto fronts, whereas for HV and Spread indicators, the higher the value, the better the computed Pareto fronts.

### 4.3 Experimental design

We have designed four scenarios for our Physarum competition algorithm, where a set of Physarum will be deployed over the search space (cell automaton) using either random sampling technique (Monte Carlo) McKay et al. (1979)) or stratified sampling technique (Latin Hypercube) (Lee et al. 2006), and whether the restart procedure will be applied or not. These four scenarios for our Physarum competition algorithm are as follows:

- (i) *PCAwMC*: A number of Physarum were deployed over the cell automaton using Monte Carlo sampling technique, and the restart procedure was not applied.
- (ii) *PCAwLHS*: A number of Physarum were deployed over the cell automaton using Latin hypercube sampling technique, and the restart procedure was not applied.
- (iii) *PCASwMC*: A number of Physarum were deployed over the cell automaton using Monte Carlo sampling technique, and the restart procedure was applied.
- (iv) *PCASwLHS*: A number of Physarum were deployed over the cell automaton using Latin hypercube sampling technique, and the restart procedure was applied.

**Table 2** The characteristics of the selected benchmark functions

Problem ID	DMOO Problems	$d$	$m$	$c$	Discrete variables bounds	Discrete interval (DI)	References
16	MOP1	2	2	0	$0 < x_n < 100$	$DI(x_n) =$ 0.01	Deb (1999)
17	MOP2	2	2	0	$0 < x_n < 100$	$DI(x_n) =$ 0.01	
18	MOP3	2	2	0	$0 < x_n < 100$	$DI(x_n) =$ 0.01	
19	MOP4	2	2	0	$0 < x_n < 100$	$DI(x_n) =$ 0.01	
20	MOP5	2	2	0	$0 < x_n < 100$	$DI(x_n) =$ 0.01	
21	MOP1	3	2	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	
22	MOP2	3	2	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	
23	MOP3	3	2	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	
24	MOP4	3	2	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	
25	MOP5	3	2	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	
26	MOP6	3	3	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	
27	MOP7	3	3	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	
28	NMMin	2	2	0	$-1000 < x_0 <$ 1000 $-1000 < x_1 <$ 1000	$DI(x_0) = 10$ $DI(x_1) = 10$	Nebro (2014)
29	Srinivas	2	2	2	$-200 < x_n < 200$	$DI(x_n) = 0.1$	Srinivas and Deb (1994)
30	UF6	4	2	0	$0 < x_0 < 10$ $-10 < x_n < 10$	$DI(x_0) = 0.1$ $DI(x_n) = 0.1$	Zhang et al. (2008)
31	Viennet2	2	3	0	$-400 < x_n < 400$	$DI(x_n) =$ 0.01	Deb (1999)
32	Viennet4	2	3	3	$-400 < x_n < 400$	$DI(x_n) =$ 0.01	
33	ZDT1	2	2	0	$0 < x_n < 100$	$DI(x_n) =$ 0.01	Zitzler et al. (2000)
34	ZDT2	2	2	0	$0 < x_n < 100$	$DI(x_n) =$ 0.01	
35	ZDT6	2	2	0	$0 < x_n < 100$	$DI(x_n) =$ 0.01	
36	ZDT1	3	2	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	
37	ZDT2	3	2	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	
38	ZDT6	3	2	0	$0 < x_n < 50$	$DI(x_n) =$ 0.05	

**Table 3** A summary of the parameters of the algorithms used in the experiments

Algorithms	Parameter settings
PCAwMC	max-iter = 250, $N = 100$ , $PM_i^0 = 3000$ , Diff_Thr = 50, Rep_Thr = 50
PCAwLHS	max-iter = 250, $N = 100$ , $PM_i^0 = 3000$ , Diff_Thr = 50, Rep_Thr = 50
PCASwMC	max-iter = 250, restart-iteration = 10, $N = 100$ , $PM_i^0 = 3000$ , Diff_Thr = 50, Rep_Thr = 50
PCASwLHS-RE	max-iter = 250, restart-iteration = 10, $N = 100$ , $PM_i^0 = 3000$ , Diff_Thr = 50, Rep_Thr = 50
NSGA-II	max-ev = 2500, $N = 100$ , $P_c = 0.9$ , $\eta_c = 20.0$ , $P_m = 1/d$ , $\eta_m = 20.0$ , OPERATOR = SBXC + PM + BTS
NSGA-II-45	max-ev = 2500, $N = 100$ , $P_c = 0.9$ , $\eta_c = 20.0$ , $P_m = 1/d$ , $\eta_m = 20.0$ , OPERATOR = SBXC+ PM + BTS
NSGA-III	max-iter = 250, $N = 100$ , $P_c = 0.9$ , $\eta_c = 30.0$ , $P_m = 1/d$ , $\eta_m = 20.0$ , OPERATOR = SBXC+ PM + BTS
SPEA2	max-iter = 250, $N = 100$ , $P_c = 0.9$ , $\eta_c = 20.0$ , $P_m = 1/d$ , $\eta_m = 20.0$ , OPERATOR = SBXC+ PM + BTS

Other MOEAs used for comparison were NSGA-II (Deb et al. 2000), NSGA-II-45,<sup>2</sup> NSGA-III (Deb and Jain 2014) and SPEA2 (Zitzler et al. 2001). These four algorithms are implemented in jMetal. The parameter settings of these algorithms are shown in Table 3, where  $N$ , max-iter and max-ev indicate the population size, maximum number of iterations and the maximum number of function evaluations, respectively.  $P_{\text{Mass}}$ , Diff\_Thr and Rep\_Thr indicate the Physarum initial mass, diffusion threshold used in Eq. (4) and repulsion threshold used in Eq. (9), respectively. OPERATOR indicates the evolutionary operator used in MOEA, where SBX, PM and BTS stand for simulated binary crossover using two parent solutions (integer encoding) (Agrawal et al. 1995), polynomial-based mutation (Coello et al. 2007) and binary tournament selection (Miller and Goldberg 1995), respectively.  $P_c$  and  $P_m$  are the crossover and mutation probability.  $d$  is the number of decision variables.  $\eta_c$  and  $\eta_m$  are the distribution indexes of SBX and PM, respectively.

#### 4.4 Experimental results and statistical analysis

Given an entire computed true Pareto optimal front for a benchmark function, the obtained results were analysed on the basis of the performance indicators as mentioned in Sect. 4.2. In this research, exhaustive experimental studies and extensive statistical analysis were performed to com-

<sup>2</sup> Different implementation of NSGA-II in jMetal with different parameters.

pare the performance of our proposed Physarum competition algorithms with four evolutionary algorithms. A minimum number of 30 independent runs is usually recommended; however, in our research 40 independent runs were executed for each algorithm.

As our experimental results did not follow a normal distribution, the differences between the results were assessed using the median and interquartile range (IQR). More detailed information was obtained by printing these results using box plots. Afterwards, we have applied a nonparametric statistical test, using Kruskal–Wallis for independent samples (also called distribution-free methods) (Kruskal et al. 1952), to test the null hypothesis that all algorithms have the same performance in each indicator for a certain benchmark function. Following the rejection of Kruskal–Wallis test and in order to find statistical significance between each algorithm, post hoc pair-wise analysis was further performed between the results of each algorithm, using the Dunn approach with the Bonferroni correction (Dunn 1964) at a 0.05 significance level.

#### 4.5 Experimental results on two-dimensional optimisation problems

For illustrating our results in the discrete 2-D optimisation problem, the deep valley bi-objective optimisation problem was selected as its continuous counterpart MOO problem is considered to be difficult for evolutionary algorithms (Deb 1999). The feasible set of this problem is the integer points

**Table 4** The statistical pair-wise analysis of PCAwMC VS MOEA algorithms with respect to performance indicators (HV, Epsilon, IGD+ and Spread) in deep valley problem

PCAwMC	HV	Epsilon	IGD+	Spread
NSGA-II (Sig.)	▲ (0.040)	– (1.000)	▲ (0.005)	▲ (0.001)
NSGA-III (Sig.)	▲ (0.001)	▲ (0.001)	▲ (0.001)	▲ (0.001)
SPEA2 (Sig.)	▽ (0.001)	▽ (0.008)	– (0.087)	▲ (0.001)

Three different symbols are used: “–” indicates that there is no statistical significance between the algorithms and “▲” and “▽” were used when PCAwMC achieves statistically better or worse results, respectively, than the MOEA algorithms

within the square  $[0, 100] \times [0, 100]$ , and the Pareto front is located at the bottom of a deep valley from the dominated regions. Statistical analysis was performed to compare the performance of our proposed Physarum competition algorithms (PCAwMC) in comparison with the three state-of-the-art evolutionary algorithms (NSGA-II, NSGA-III and SPEA2). The box plot (Fig. 4) of all indicators showed that our algorithm achieved the best performance for the Spread indicator about 1.9 compared to 0.8, 0.4 and 0.3 for NSGA-II, NSGA-III and SPEA2, respectively. These results analyses clearly demonstrate the ability of our algorithm to provide a better spread of solutions with good convergence behaviour. It has also confirmed our assumption that individual skills of competing Physarum are more efficient in exploration, which increases the diversity of the solutions. As regards the other performance indicators, PCA has achieved the second-best values for IGD+, HV and Epsilon indicators that are only exceeded by SPEA2. The pair-wise analysis (Table 4) showed that PCA spread performance is highly significant ( $P < 0.001$ ) than all other algorithms. PCA outperformed the NSGA-II and NSGA-III for other indicators. PCA achieved nearly similar IGD+ performance as SPEA2 with no statistical difference, it has been only exceeded by SPEA2 for HV and Epsilon indicators.

#### 4.6 Experimental results on $n$ -dimensional optimisation problems

In this subsection, we provide an extended version of our earlier conference poster paper (Awad et al. 2019b). We have extended the discrete search space of the Physarum from 2-D problems as deep valley problem to  $n$ -D generalisations of the hexagonal CA grids to solve multi-dimensional, multi-objective optimisation problems.

Exhaustive experimental study and extensive statistical analysis was performed to compare the performance of four of our proposed Physarum competition algorithms (i.e.

PCAwMC, PCAwLHS, PCASwMC and PCASwLHS) in comparison with the four evolutionary algorithms (NSGA-II, NSGA-II-45, NSGA-III and SPEA2) on a benchmark suite of 38 problems (as indicated in Sect. 4.1) ranging from two dimensions to seven dimensions with up to 10 objectives and 5 constraints.

There are some previous benchmarking studies for MOEAs (Li et al. 2013; Bezerra et al. 2018); however, to the best of our knowledge, an exhaustive benchmarking study has never been performed. As far as we know, this study is the first attempt to assess algorithms that solve DMOO problems, with a large number of benchmark functions and performance indicators.

The complete and detailed results of the performance indicators ranking with respect to the 38 benchmark functions for each algorithm are provided in “Appendix” (Table 7).

#### 4.7 The overall performance of PCA versus MOEAs

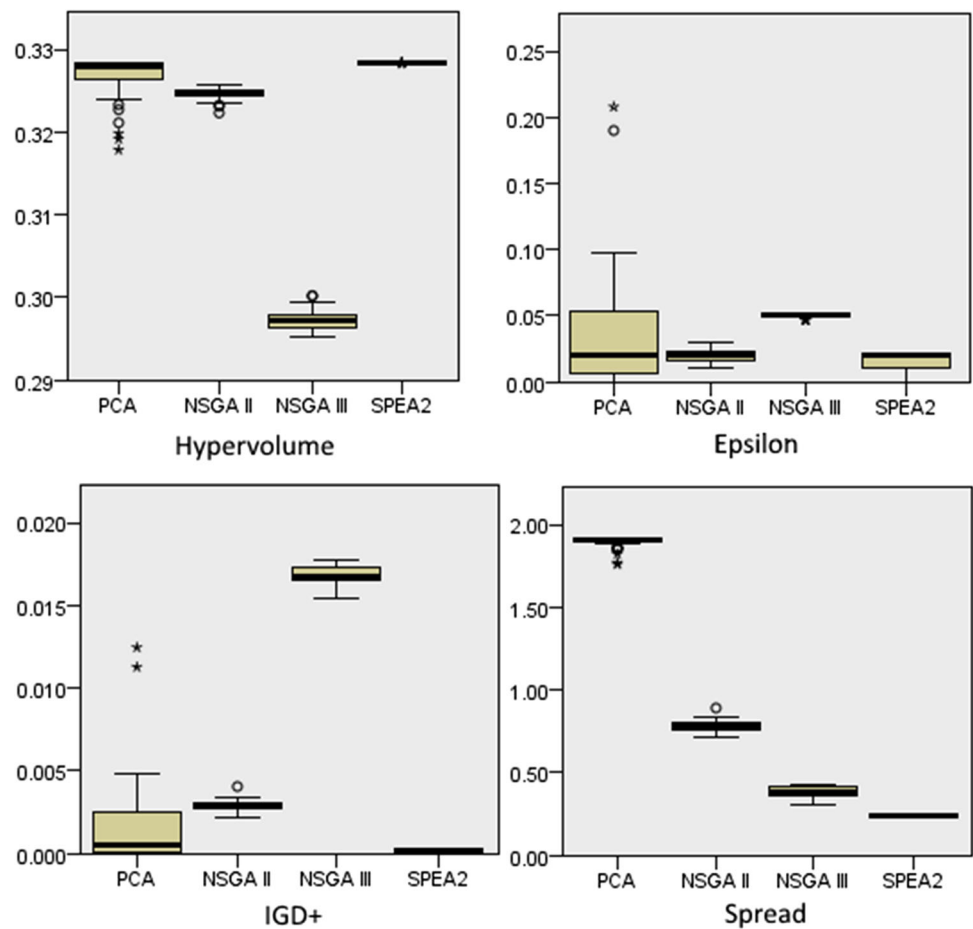
We exhaustively investigate the anytime performance (overall performance) of our four PCAs versus four MOEAs across all benchmark functions collectively to compare algorithms using the given performance indicator values (IGD+, HV, Epsilon and Spread) obtained in 40 runs for each algorithm. For this comparison, we calculated (from the statistical results in Table 7) the mean rank of performance indicators for each algorithm as shown in Table 5.

In addition, we performed a nonparametric statistical test, using Friedman for dependent samples (Friedman 1937), to test the null hypothesis that all algorithms are having the same performance for each indicator across all the benchmark functions. Following the rejection of Friedman test and in order to find statistical significance with respect to each performance indicator, between each algorithm, post hoc pair-wise analysis was further performed between the results, using the Dunn approach with the Bonferroni correction (Dunn 1964) at a 0.05 significance level. Table 6 summarises the pair-wise comparison between algorithms’ performance in terms of HV, IGD+, Epsilon and Spread indicators, respectively, for all benchmark functions.

Looking at the results presented in Tables 5 and 6, it can be seen that PCASwLHS and PCASwMC have achieved the best performance for the Spread and IGD+ indicators and the second-best values for the HV and Epsilon indicators after SPEA2. Furthermore, the PCA with restart procedure algorithms have been proven to outperform NSGA-II, NSGA-II-45 and NSGA-III in terms of all performance indicators. This good performance of the PCAs in terms of the diversity (Spread indicator) has confirmed our hypothesis that the individual skills of the competing individuals are more efficient for avoiding the premature convergence and increasing the ability of escaping from local optima. The combination of the restart procedure and LHS deployment strategy



**Fig. 4** Box plot graph of performance indicators for our PCAs and the four state-of-the-art MOEAs with respect to deep valley problem



**Table 5** The mean rank of performance indicators as sorted from the best performance to the worst in an up-down arrangement with respect to all benchmark functions collectively

HV	IGD+		Epsilon		Spread		
SPEA2	7.24	PCASwLHS	2.66	SPEA2	1.86	PCASwMC	6.11
PCASwLHS	5.63	PCASwMC	2.74	PCASwLHS	3.41	PCASwLHS	5.87
PCASwMC	5.51	SPEA2	3.11	PCASwMC	3.57	NSGA-II	4.95
NSGA-II-45	4.96	NSGA-II-45	4.3	NSGA-II-45	4.03	NSGA-II-45	4.84
NSGA-II	4.93	NSGA-II	4.42	NSGA-II	4.08	PCAwMC	4.33
NSGA-III	3.36	PCAwLHS	5.97	NSGA-III	5.41	PCAwLHS	4.3
PCAwLHS	2.3	PCAwMC	6.13	PCAwLHS	6.82	SPEA2	3.03
PCAwMC	2.07	NSGA-III	6.67	PCAwMC	6.84	NSGA-III	2.58

(PCASwLHS) has improved the overall performance. This is because LHS enables well-distributed exploration covering the whole search space and the restart procedure drives the exploitation towards the best solution based on both the personal experience and the shared information by vanishing unfavourable paths (i.e. dominated solutions).

One of the interesting points in this research is the large number of benchmark functions and the diverse set of performance metrics used. These makes it very challenging for selecting appropriate statistical analysis methods to compare the performance of different algorithms. For this reason, the

vast majority of published articles used only one or two performance metrics and a limited number of benchmark functions (Engelbrecht 2014). Due to the lack of research that covers in detail how to conduct statistical analysis to assess the performance of MOO algorithms, we intend in the future to develop a statistical framework for performing analysis to make it easier for researchers to evaluate their newly proposed algorithms.

**Table 6** The statistical pair-wise performance analysis shows a comparison of our PCAs with four MOEA algorithms for all benchmark problems

Problems	NSGA-II				NSGA-II-45				NSGA-III				SPEA2			
	HV	IGD+	Epsilon	Spread	HV	IGD+	Epsilon	Spread	HV	IGD+	Epsilon	Spread	HV	IGD+	Epsilon	Spread
PCAwMC	▽	▽	▽	-	▽	▽	▽	-	-	-	▽	-	▽	▽	▽	▽
PCAwLHS	▽	▽	▽	-	▽	-	▽	-	-	-	-	-	▽	▽	▽	▽
PCASwMC	-	-	-	-	-	-	-	▲	-	-	-	-	▽	-	▽	-
PCASwLHS	-	-	-	▲	-	-	-	▲	▲	▲	-	-	-	-	-	-

Three different symbols are used: “-” indicates that there is no statistical significance between the algorithms and “▲” and “▽” were used when the algorithm in the column (PCAs) achieves statistically better or worse results, respectively, than the algorithm in the row (MOEA algorithms)

## 5 Conclusion and future work

Many real-world problems can be naturally formulated as discrete multi-objective optimisation (DMOO) problems. To the best of our knowledge, we are the first to present a Physarum competition algorithm for solving discrete multi-objective optimisation problems. This algorithm is based on Physarum decision-making capabilities, and its discrete motility over a hexagonal cellular automaton allowed effective search in discrete search space. The modelling of the Physarum motility is based on the chemo-attraction forces towards food resources (objective functions) and the repulsion negative forces that competing Physarum exert on each other. We have considered the n-dimensional generalisation of the hexagonal grids to extend the solving capabilities of our PCA from only two-dimensional to multi-dimensional optimisation problems. We have implemented a novel restart procedure which improved the overall performance as it drives the exploitation towards the best solution based on both personal experience and shared information. Extensive experiments were conducted on several benchmark functions to assess the performance of our PCA against state-of-the-art evolutionary algorithms. The statistical analysis clearly demonstrated that our algorithm had achieved the best performance for the Spread and IGD+ indicators and the second-best values for the HV and Epsilon indicators after SPEA2. Furthermore, the PCA with restart procedure algorithms has been proven to outperform NSGA-II and NSGA-III in terms of all performance indicators. These results have confirmed our assumption that individual skills of competing Physarum are more efficient in exploration, which increases the diversity of the solutions. These results have demonstrated that PCA is a promising algorithm for solving DMOO problems. In addition, since our algorithm is implemented in Java, it can be easily integrated with the jMetal framework for providing a flexible and configurable PCA.

For future work, we intend to explore the potential for solving continuous MOOP. This can be achieved by partitioning the numeric continuous variables in the search space into several sub-ranges, thus transforming the continuous-valued attributes into a finite number of discrete intervals. Accordingly, using the CA can avoid a huge amount of computation caused by the huge number of states selected as continuous actions. In our (PCA) algorithm, we have considered the n-dimensional generalisation of the hexagonal grids to extend the solving capabilities of our PCA from only two-dimensional to multi-dimensional optimisation problems without the need to use multiple automata. So, we believe that the generalisation of our algorithm to solve both discrete MOOP, and continuous MOOP is possible with expected good results.

**Funding** Dr. Abubakr Awad was supported by Elphinstone PhD Scholarship (University of Aberdeen). Dr. Wei Pang and Prof. George M. Coghil were supported by the Royal Society International Exchange program (Grant Ref IE160806). This work was performed using the Maxwell Compute Cluster funded by the University of Aberdeen.

**Data availability** Enquiries about data availability should be directed to the authors.

**Declarations**

**Conflict of interest** The authors declare that they have no conflict of interest.

**Research involving human participants and/or animals** This manuscript does not contain any studies with human participants or animals performed by any of the authors.

**Informed consent** Not applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

**Appendix**

See Table 7.

**Table 7** Detailed performance results of the used algorithms across the 38 benchmark problems

Problem ID	PI	PCAwMC	PCAwLHS	PCASwMC	PCASwLHS	NSGA-II	NSGA-II-45	NSGA-III	SPEA2
1	HV	55.45	64.21	234.04	244.93	172.24	170.51	62.13	280.50
1	Epsilon	258.31	244.60	125.46	114.21	136.50	120.04	228.05	56.83
1	IGDP	261.59	254.09	83.10	70.90	152.91	154.28	265.81	41.33
1	Spread	54.54	44.29	279.58	276.71	202.59	203.13	83.30	139.88
2	HV	160.50	160.50	160.50	160.50	160.50	160.50	160.50	160.50
2	Epsilon	234.68	246.69	234.63	246.01	90.51	78.49	76.50	76.50
2	IGDP	232.94	246.49	232.80	249.78	90.48	78.53	76.50	76.50
2	Spread	75.59	80.08	81.71	84.63	272.25	244.49	194.20	251.06
3	HV	64.15	86.33	221.84	237.08	172.96	168.65	165.54	167.46
3	Epsilon	163.73	152.70	73.24	65.03	190.19	217.66	201.36	220.10
3	IGDP	164.56	151.13	81.25	73.19	185.66	219.80	184.13	224.29
3	Spread	249.15	241.25	208.55	196.06	91.24	84.95	192.30	20.50
4	HV	88.34	72.10	143.99	126.05	215.36	190.16	210.63	237.38
4	Epsilon	199.73	210.14	156.94	162.51	123.99	163.59	149.78	117.34
4	IGDP	195.78	190.00	131.84	130.99	139.74	174.40	172.83	148.44
4	Spread	196.86	193.30	226.26	237.23	111.05	114.19	176.60	28.51
5	HV	38.98	42.03	256.50	256.50	163.60	169.40	100.50	256.50
5	Epsilon	282.21	278.79	49.00	49.00	165.55	158.04	217.75	83.66
5	IGDP	281.55	279.45	65.00	65.00	160.79	147.06	220.15	65.00
5	Spread	100.69	116.83	131.99	133.93	280.03	277.31	20.53	222.71
6	HV	68.05	53.80	238.00	238.00	190.90	190.53	66.73	238.00
6	Epsilon	268.09	267.96	51.50	51.50	152.59	144.46	245.10	102.80
6	IGDP	252.78	251.33	93.00	93.00	118.00	105.50	277.40	93.00
6	Spread	198.85	202.96	222.60	248.33	146.39	135.96	31.84	97.08
7	HV	77.73	83.33	257.49	258.55	161.94	163.48	20.50	261.00
7	Epsilon	248.65	247.81	69.66	90.31	158.45	152.76	284.85	31.50
7	IGDP	243.79	237.21	64.86	61.41	157.58	159.65	300.50	59.00
7	Spread	182.93	164.19	273.21	275.35	100.25	92.59	174.99	20.50
8	HV	75.25	78.26	215.96	219.23	222.00	222.00	29.30	222.00
8	Epsilon	247.50	243.74	71.24	80.60	117.80	125.60	287.53	110.00

**Table 7** (continued)

Problem ID	PI	PCAwMC	PCAwLHS	PCASwMC	PCASwLHS	NSGA-II	NSGA-II-45	NSGA-III	SPEA2
8	IGDP	241.33	239.18	103.00	100.00	100.00	100.00	300.50	100.00
8	Spread	186.09	165.39	271.86	275.89	88.91	87.49	184.60	23.78
9	HV	72.00	80.80	233.00	233.00	205.74	197.76	28.70	233.00
9	Epsilon	253.04	266.85	52.00	52.00	148.21	151.11	261.61	99.18
9	IGDP	258.73	257.23	98.00	98.00	100.50	108.00	265.55	98.00
9	Spread	77.81	82.31	189.98	194.76	236.65	222.43	40.38	239.69
10	HV	132.93	122.03	240.99	223.03	64.01	53.31	243.00	204.71
10	Epsilon	187.59	196.44	134.68	135.04	261.14	263.00	24.00	82.13
10	IGDP	128.05	136.08	51.74	57.73	246.81	266.36	261.03	136.21
10	Spread	165.89	189.88	197.86	193.85	226.49	227.85	23.21	58.98
11	HV	158.05	191.69	187.41	150.65	164.45	212.51	37.30	181.94
11	Epsilon	189.80	165.04	176.45	210.74	111.19	81.51	260.83	88.45
11	IGDP	137.35	121.18	125.28	150.81	182.26	123.34	293.69	150.10
11	Spread	76.15	85.39	84.81	79.60	243.13	251.06	276.26	187.60
12	HV	132.74	142.96	146.33	145.24	200.09	179.33	123.04	214.29
12	Epsilon	186.85	172.35	172.50	171.84	123.75	140.75	214.58	101.39
12	IGDP	156.50	156.50	156.50	156.50	156.50	156.50	188.50	156.50
12	Spread	108.66	90.73	96.96	93.03	259.91	269.60	113.15	251.96
13	HV	89.13	79.78	93.51	82.58	229.46	209.00	261.14	239.41
13	Epsilon	216.20	231.01	217.69	231.28	105.75	121.68	67.95	92.45
13	IGDP	157.00	157.00	157.00	157.00	157.00	157.00	185.00	157.00
13	Spread	102.54	132.65	114.43	128.46	272.16	276.76	26.00	231.00
14	HV	129.69	126.50	130.13	134.83	185.48	169.36	210.85	197.18
14	Epsilon	240.05	235.95	236.80	232.75	78.34	105.56	85.59	68.96
14	IGDP	160.50	160.50	160.50	160.50	160.50	160.50	160.50	160.50
14	Spread	89.58	121.11	103.01	122.34	258.13	255.90	70.45	263.49
15	HV	90.50	84.33	77.58	91.39	227.79	228.38	241.78	242.28
15	Epsilon	218.59	231.81	239.18	220.10	90.68	92.84	120.84	69.98
15	IGDP	157.50	157.50	157.50	157.50	157.50	157.50	181.50	157.50
15	Spread	114.99	103.04	122.13	111.25	263.00	262.88	51.10	255.63
16	HV	58.25	54.63	221.70	221.15	179.50	177.88	73.14	297.76
16	Epsilon	242.55	254.21	151.85	149.69	141.00	130.78	186.30	27.63
16	IGDP	263.35	258.44	81.64	81.59	145.45	152.26	251.19	50.09
16	Spread	143.30	141.69	251.53	246.24	208.66	210.79	22.68	59.13
17	HV	70.11	73.91	224.00	220.73	170.99	171.11	60.75	292.40
17	Epsilon	223.05	217.45	138.10	136.08	152.71	151.33	233.36	31.93
17	IGDP	208.65	213.10	74.31	76.84	184.70	152.68	277.81	95.91
17	Spread	147.13	134.44	237.21	227.75	234.44	221.84	53.50	27.70
18	HV	68.33	64.88	239.78	249.25	161.75	164.50	54.63	280.90
18	Epsilon	255.43	258.93	101.36	86.68	139.21	127.64	242.23	72.54
18	IGDP	256.74	256.30	77.33	70.79	158.09	159.28	262.64	42.85
18	Spread	167.15	163.43	192.59	163.60	260.30	253.63	59.26	24.05
19	HV	51.55	54.69	237.34	256.94	171.06	167.44	84.29	260.70
19	Epsilon	264.84	258.63	78.58	59.90	146.65	158.54	249.41	67.46
19	IGDP	215.91	202.04	83.06	70.50	153.64	153.64	280.28	124.94
19	Spread	80.03	82.68	283.43	273.16	194.16	204.66	20.80	145.09

Table 7 continued

Problem ID	PI	PCAwMC	PCAwLHS	PCASwMC	PCASwLHS	NSGA-II	NSGA-II-45	NSGA-III	SPEA2
20	HV	44.71	45.53	214.05	220.10	186.74	179.45	94.75	298.68
20	Epsilon	268.95	269.09	147.79	140.54	106.15	112.26	203.29	35.94
20	IGDP	249.50	252.84	93.53	91.85	138.73	148.60	279.06	29.90
20	Spread	196.33	187.51	253.89	238.65	160.33	166.30	26.59	54.41
21	HV	48.13	45.68	186.09	198.60	202.46	189.00	119.85	294.20
21	Epsilon	264.78	262.58	146.10	126.33	123.73	146.51	165.89	48.10
21	IGDP	250.00	262.00	138.00	126.00	126.00	126.00	130.00	126.00
21	Spread	191.51	196.54	253.61	259.41	131.96	128.45	20.55	101.96
22	HV	141.01	141.15	139.84	155.44	165.91	168.18	86.94	285.54
22	Epsilon	136.98	133.08	213.46	191.19	183.90	157.73	197.49	70.19
22	IGDP	141.31	152.56	137.88	137.88	143.56	139.38	275.31	156.13
22	Spread	206.63	198.30	279.10	270.10	111.85	109.43	20.50	88.10
23	HV	164.33	172.98	142.59	126.90	140.56	173.68	76.55	286.43
23	Epsilon	214.24	224.39	142.74	150.90	152.00	128.51	206.14	65.09
23	IGDP	179.50	183.50	159.50	171.50	187.50	171.50	187.50	43.50
23	Spread	148.20	144.53	282.58	276.14	158.28	152.43	20.81	101.05
24	HV	67.00	64.18	160.43	196.38	182.11	178.48	152.21	283.23
24	Epsilon	252.13	245.70	172.80	141.91	137.54	132.54	162.10	39.29
24	IGDP	143.00	143.00	143.00	143.00	150.95	158.90	239.28	162.88
24	Spread	194.64	200.16	281.00	278.08	116.60	121.60	20.50	71.43
25	HV	60.10	65.80	195.24	211.33	157.88	154.63	142.40	296.64
25	Epsilon	235.99	248.60	145.31	150.84	122.53	148.60	196.66	35.48
25	IGDP	216.25	218.28	130.00	122.50	141.25	145.00	240.73	70.00
25	Spread	193.33	175.44	286.31	256.16	132.56	132.73	20.50	86.98
26	HV	68.05	50.26	198.70	197.33	116.44	117.95	300.25	235.03
26	Epsilon	236.85	242.16	173.95	178.20	190.60	181.16	27.49	53.59
26	IGDP	205.00	217.50	57.05	58.31	197.55	181.30	87.25	280.04
26	Spread	235.18	252.79	236.54	237.48	118.44	121.66	51.09	30.84
27	HV	74.74	89.99	144.11	142.79	131.29	150.13	251.89	299.08
27	Epsilon	212.11	187.96	199.74	208.28	185.74	168.54	80.04	41.60
27	IGDP	230.66	245.41	82.23	81.85	95.49	87.04	174.34	286.99
27	Spread	209.08	204.58	272.76	270.26	121.54	122.91	62.38	20.50
28	HV	46.01	49.26	223.00	223.00	213.00	220.50	86.23	223.00
28	Epsilon	282.20	275.80	90.00	90.00	122.50	110.00	223.50	90.00
28	IGDP	273.61	268.80	100.50	100.50	100.50	100.50	239.09	100.50
28	Spread	183.16	185.24	278.24	279.21	121.63	114.03	20.50	102.00
29	HV	138.18	144.04	280.50	280.50	129.30	121.30	20.50	169.69
29	Epsilon	223.46	227.59	41.64	40.24	166.34	168.48	300.50	115.76
29	IGDP	174.39	169.46	40.50	40.50	210.58	204.65	300.50	143.43
29	Spread	162.25	158.54	238.91	242.05	82.31	78.89	300.50	20.55
30	HV	136.33	147.09	157.75	160.98	153.06	147.58	154.70	226.53
30	Epsilon	182.98	212.84	199.90	198.00	123.04	121.11	181.29	64.85
30	IGDP	198.68	205.18	157.49	167.15	133.06	135.44	193.84	93.18
30	Spread	70.31	85.34	88.91	83.64	247.50	251.45	186.51	270.34
31	HV	52.15	38.79	196.60	206.00	208.90	208.93	92.28	280.36
31	Epsilon	264.83	278.79	118.59	113.98	130.86	122.69	212.10	42.18



Table 7 continued

Problem ID	PI	PCAwMC	PCAwLHS	PCASwMC	PCASwLHS	NSGA-II	NSGA-II-45	NSGA-III	SPEA2
31	IGDP	268.44	279.01	80.74	67.79	148.03	146.46	223.41	70.13
31	Spread	244.48	257.93	108.50	116.66	122.65	138.65	264.09	31.05
32	HV	46.50	65.55	182.68	186.23	177.71	171.59	173.83	279.93
32	Epsilon	239.93	223.05	131.46	114.71	180.04	189.69	166.90	38.23
32	IGDP	177.36	171.40	49.64	42.16	235.98	237.50	265.98	103.99
32	Spread	261.39	279.18	101.96	100.58	142.29	154.34	191.56	52.71
33	HV	39.55	41.45	260.50	260.50	159.58	161.43	100.50	260.50
33	Epsilon	281.18	279.83	40.50	40.50	161.56	159.44	220.50	100.50
33	IGDP	280.38	280.63	60.50	60.50	162.45	158.55	220.50	60.50
33	Spread	229.26	228.00	250.29	252.65	120.89	121.91	60.50	20.50
34	HV	41.03	39.98	261.00	256.51	160.33	162.68	101.49	261.00
34	Epsilon	278.63	281.15	40.50	46.71	140.40	144.96	219.50	132.15
34	IGDP	281.26	279.74	60.00	64.50	160.90	158.10	219.50	60.00
34	Spread	228.83	227.28	246.75	244.11	128.16	124.90	62.50	21.48
35	HV	37.85	43.15	220.50	220.50	220.50	220.50	100.50	220.50
35	Epsilon	283.36	277.64	60.50	60.50	161.61	159.39	220.50	60.50
35	IGDP	284.05	274.96	100.50	100.50	100.50	100.50	222.49	100.50
35	Spread	83.96	85.86	279.28	281.45	181.85	177.10	20.50	174.00
36	HV	40.40	40.60	219.45	214.86	219.65	222.04	102.50	224.50
36	Epsilon	281.74	279.26	47.95	53.05	141.53	135.63	215.35	129.50
36	IGDP	282.06	278.94	97.74	102.45	111.90	99.91	218.50	92.50
36	Spread	227.63	201.04	257.14	236.03	121.23	114.45	20.50	106.00
37	HV	40.71	40.29	225.50	225.50	213.06	210.94	102.50	225.50
37	Epsilon	279.39	281.55	40.50	40.50	152.95	152.91	203.20	133.00
37	IGDP	280.85	280.15	40.50	40.50	146.26	147.05	217.19	131.50
37	Spread	213.88	212.35	248.44	245.14	120.16	121.04	21.50	101.50
38	HV	41.98	39.25	217.99	217.79	221.50	221.50	102.50	221.50
38	Epsilon	276.20	284.45	94.33	94.48	115.43	110.13	218.50	90.50
38	IGDP	276.63	284.18	102.53	102.70	99.00	101.48	218.50	99.00
38	Spread	78.96	78.96	204.45	186.60	244.65	237.23	24.15	229.00

## References

- Abo-Hammour Z, Alsmadi O, Momani S, Arqub OA (2013) A genetic algorithm approach for prediction of linear dynamical systems. *Math Probl Eng* 2013
- Abo-Hammour Z, Arqub OA, Momani S, Shawagfeh N (2014) Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic algorithm. *Discrete Dyn Nat Soc* 2014
- Adamatzky A (2010) *Physarum machines: computers from slime mould*. World Scientific. [https://books.google.co.uk/books?id=Kbs\\_AIDbfU8C](https://books.google.co.uk/books?id=Kbs_AIDbfU8C)
- Adelgren N, Gupte A (2017) Branch-and-bound for biobjective mixed integer programming. arXiv preprint [arXiv:1709.03668](https://arxiv.org/abs/1709.03668)
- Afek Y, Alon N, Barad O, Hornstein E, Barkai N, Bar-Joseph Z (2011) A biological solution to a fundamental distributed computing problem. *Science* 331(6014):183–185. <https://doi.org/10.1126/science.1193210>
- Agrawal RB, Deb K, Agrawal R (1995) Simulated binary crossover for continuous search space. *Complex Syst* 9(2):115–148
- Alaya I, Solnon C, Ghedira K (2007) Ant colony optimization for multi-objective optimization problems. In: 19th IEEE international conference on tools with artificial intelligence (ICTAI 2007), vol 1, pp 450–457. <https://doi.org/10.1109/ICTAI.2007.108>. ID: 1
- Alim K, Amselem G, Peaudecerf F, Brenner MP, Pringle A (2013) Random network peristalsis in physarum polycephalum organizes fluid flows across an individual. *Proc Natl Acad Sci U S A* 110(33):13306–13311. <https://doi.org/10.1073/pnas.1305049110>
- Arqub OA, Abo-Hammour Z (2014) Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. *Inf Sci* 279:396–415
- Arunachalam V (2008) Optimization using differential evolution
- Audet C, Savard G, Zghal W (2008) Multiobjective optimization through a series of single-objective formulations. *SIAM J Optim* 19(1):188–210. <https://doi.org/10.1137/060677513>
- Awad A, Pang W, Lusseau D, Coghill GM (2019a) A hexagonal cell automaton model to imitate physarum polycephalum competitive behaviour. In: The 2018 conference on artificial life: a hybrid of the European conference on artificial life (ECAL) and the interna-

- tional conference on the synthesis and simulation of living systems (ALIFE), pp 203–210. MIT Press
- Awad A, Usman M, Lusseau D, Coghill GM, Pang W (2019b) A physarum-inspired competition algorithm for solving discrete multi-objective optimization problems. In: Genetic and evolutionary computation conference companion (GECCO '19 companion). GECCO '19. ACM, New York. <https://doi.org/10.1145/3319619.3322030>
- Awad A, Pang W, Lusseau D, Coghill GM (2021) A survey on physarum polycephalum intelligent foraging behaviour and bio-inspired applications. *Artif Intell Rev* 2103:00172
- Beekman M, Latty T (2015) Brainless but multi-headed: decision making by the acellular slime mould *Physarum polycephalum*. *J Mol Biol* 427(23):3734–3743. <https://doi.org/10.1016/j.jmb.2015.07.007>
- Bektaş T (2018) Disjunctive programming for multiobjective discrete optimisation. *INFORMS J Comput* 30(4):625–633
- Beni G, Wang J (1993) Swarm intelligence in cellular robotic systems. In: *Robots and biological systems: Towards a new bionics?*, pp 703–712. Springer
- Bergman D, Bodur M, Cardonha C, Cire AA (2018) Network models for multiobjective discrete optimization. arXiv preprint [arXiv:1802.08637](https://arxiv.org/abs/1802.08637)
- Bezerra LC, López-Ibáñez M, Stützle T (2018) A large-scale experimental evaluation of high-performing multi-and many-objective evolutionary algorithms. *Evol Comput* 26(4):621–656
- Boland N, Charkhgard H, Savelsbergh M (2015) A criterion space search algorithm for biobjective integer programming: the balanced box method. *INFORMS J Comput* 27(4):735–754. <https://doi.org/10.1287/ijoc.2015.0657>
- Boussard A, Delescluse J, Pérez-Escudero A, Dussutour A (2019) Memory inception and preservation in slime moulds: the quest for a common mechanism. *Philos Trans R Soc B* 374(1774):20180368
- Brockhoff D (2018) Gecco 2018 tutorial on evolutionary multiobjective optimization. In: *Proceedings of the genetic and evolutionary computation conference companion. GECCO '18*, pp 349–372. ACM, New York. <https://doi.org/10.1145/3205651.3207864>
- Cacchiani V, D'Ambrosio C (2017) A branch-and-bound based heuristic algorithm for convex multi-objective MINLPs. *Eur J Oper Res* 260(3):920–933. <https://doi.org/10.1016/j.ejor.2016.10.015>
- Coello CAC, Lamont GB, Veldhuizen DAV (2007) *Evolutionary algorithms for solving multi-objective problems*, vol 5. Springer (2007)
- Deb K (2014) *Multi-objective optimization*. In: *Search methodologies*, pp 403–449. Springer
- Deb K (1999) Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evol Comput* 7(3):205–230. <https://doi.org/10.1162/evco.1999.7.3.205>
- Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Trans Evol Comput* 18(4):577–601
- Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, vol 1917, pp 849–858
- Deb K, Thiele L, Laumanns M, Zitzler E (2002) Scalable multi-objective optimization test problems. In: *Proceedings of the 2002 congress on evolutionary computation, 2002. CEC'02*, vol 1, pp 825–830. IEEE
- Deb K, Mohan M, Mishra S (2005) Evaluating the e-dominance based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions. *Evol Comput* 13(4):501–525. <https://doi.org/10.1162/106365605774666895>
- Dickersbach JT (2005) *Supply chain management with APO: structures, modelling approaches and implementation of MySAP SCM 4.1*. Springer
- Dunn OJ (1964) Multiple comparisons using rank sums. *Technometrics* 6(3):241–252. <https://doi.org/10.1080/00401706.1964.10490181>
- Durillo JJ, Nebro AJ (2011) jMetal: a Java framework for multi-objective optimization. *Adv Eng Softw* 42(10):760–771
- Ehrgott M (2006) A discussion of scalarization techniques for multiple objective integer programming. *Ann Oper Res* 147(1):343–360. <https://doi.org/10.1007/s10479-006-0074-z>
- Ehrgott M, Gandibleux X, Przybylski A (2016) Exact methods for multi-objective combinatorial optimisation. *Int Ser Oper Res Management Sci* 233:817–850. [https://doi.org/10.1007/978-1-4939-3094-4\\_19](https://doi.org/10.1007/978-1-4939-3094-4_19)
- Elomaa T, Rousu J (2004) Efficient multisplitting revisited: optimapreserving elimination of partition candidates. *Data Min Knowl Disc* 8(2):97–126
- Engelbrecht AP (2014) Fitness function evaluations: A fair stopping condition? In: *2014 IEEE symposium on swarm intelligence (SIS)*, pp 1–8. IEEE
- Fonseca CM, Fleming PJ (1995) Multiobjective genetic algorithms made easy: selection sharing and mating restriction
- Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J Am Stat Assoc* 32(200):675–701. <https://doi.org/10.1080/01621459.1937.10503522>
- Gao C, Chen Z, Li X, Tian Z, Li S, Wang Z (2018) Multiobjective discrete particle swarm optimization for community detection in dynamic networks. *EPL*. <https://doi.org/10.1209/0295-5075/122/28001>
- Ghannadpour SF, Zarrabi A (2019) Multi-objective heterogeneous vehicle routing and scheduling problem with energy minimizing. *Swarm Evol Comput* 44:728–747
- Han Y, Li J, Sang H, Liu Y, Gao K, Pan Q (2020) Discrete evolutionary multi-objective optimization for energy-efficient blocking flow shop scheduling with setup time. *Appl Soft Comput* 93:106343
- Hillermeier C (2001) *Nonlinear multiobjective optimization: a generalized homotopy approach*, vol 135. Springer
- Holzmann T, Smith JC (2018) Solving discrete multi-objective optimization problems using modified augmented weighted Tchebychev scalarizations. *Eur J Oper Res* 271(2):436–449. <https://doi.org/10.1016/j.ejor.2018.05.036>
- Ishibuchi H, Masuda H, Nojima Y (2016) Sensitivity of performance evaluation results by inverted generational distance to reference points. In: *2016 IEEE congress on evolutionary computation (CEC)*, pp 1107–1114. IEEE
- Ishibuchi H, Tsukamoto N, Nojima Y (2008) Evolutionary many-objective optimization: a short review. In: *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, pp 2419–2426. IEEE
- Kirlik G, Sayin S (2014) A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *Eur J Oper Res* 232(3):479–488. <https://doi.org/10.1016/j.ejor.2013.08.001>
- Kroeker KL (2011) Biology-inspired networking. *Commun ACM* 54(10):11–14. <https://doi.org/10.1145/2001269.2001275>
- Kruskal WH, Wallis WA (1952) Use of ranks in one-criterion variance analysis. *J Am Stat Assoc* 47(260):583–621. <https://doi.org/10.1080/01621459.1952.10483441>
- Latty T, Beekman M (2010) Food quality and the risk of light exposure affect patch-choice decisions in the slime mold *Physarum polycephalum*. *Ecology* 91(1):22–27. <https://doi.org/10.1890/09-0358.1>
- Lee J-H, Ko Y-D, Yun I-G, Han K-H (2006) Comparison of Latin hypercube sampling and simple random sampling applied to neu-

- ral network modeling of HfO<sub>2</sub> thin film fabrication. *Trans Electr Electron Mater* 7(4):210–214
- Li M, Zheng J (2009) Spread assessment for evolutionary multi-objective optimization. In: International conference on evolutionary multi-criterion optimization, pp 216–230. Springer (2009)
- Li M, Yang S, Liu X, Shen R (2013) A comparative study on evolutionary algorithms for many-objective optimization. In: International conference on evolutionary multi-criterion optimization, pp 261–275. Springer
- Li B, Li J, Tang K, Yao X (2015) Many-objective evolutionary algorithms: a survey. *ACM Comput Surv (CSUR)* 48(1):13
- Liu L, Song Y, Zhang H, Ma H, Vasilakos AV (2015) Physarum optimization: a biology-inspired algorithm for the Steiner tree problem in networks. *IEEE Trans Comput* 64(3):818–831
- Liu Y, Gao C, Zhang Z, Wu Y, Liang M, Tao L, Lu Y (2017) A new multi-agent system to simulate the foraging behaviors of physarum. *Nat Comput* 16(1):15–29. <https://doi.org/10.1007/s11047-015-9530-5>
- Liu Q, Li X, Liu H, Guo Z (2020) Multi-objective metaheuristics for discrete optimization problems: a review of the state-of-the-art. *Appl Soft Comput* 93:106382
- Luong HN, Bosman PA (2012) Elitist archiving for multi-objective evolutionary algorithms: to adapt or not to adapt. In: International conference on parallel problem solving from nature, pp 72–81. Springer
- Masi L, Vasile M (2014) A multi-directional modified physarum algorithm for optimal multi-objective discrete decision making. *Stud Comput Intell* 500:195–212. [https://doi.org/10.1007/978-3-319-01460-9\\_9](https://doi.org/10.1007/978-3-319-01460-9_9)
- Mavrotas G, Diakoulaki D (1998) A branch and bound algorithm for mixed zero-one multiple objective linear programming. *Eur J Oper Res* 107(3):530–541. [https://doi.org/10.1016/S0377-2217\(97\)00077-5](https://doi.org/10.1016/S0377-2217(97)00077-5)
- McKay MD, Beckman RJ, Conover WJ (1979) Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2):239–245
- Miller BL, Goldberg DE (1995) Genetic algorithms, tournament selection, and the effects of noise. *Complex Syst* 9(3):193–212
- Mittelmann HD (2018) Decision tree for optimization software
- Mostaghim S, Teich J (2013) Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: 2003 IEEE swarm intelligence symposium, SIS 2003—proceedings, pp 26–33. <https://doi.org/10.1109/SIS.2003.1202243>
- Multiple criteria decision making (MCDM), I.S.: Problems Digital Library. <http://www.mcdmsociety.org/content/digital-library>
- Nagy B, Strand R (2009) Neighborhood sequences on nd hexagonal/face-centered-cubic grids. In: International workshop on combinatorial image analysis, pp 96–108. Springer
- Nebro A (2014) Bi-objective problem for testing integer encoding. Created by Antonio J. Nebro (antonio@lcc.uma.es) on 03/07/14
- Nebro AJ, Durillo JJ, Garcia-Nieto J, Coello CAC, Luna F, Alba E (2009) SMPSO: a new PSO-based metaheuristic for multi-objective optimization. In: 2009 IEEE symposium on computational intelligence in multi-criteria decision-making (MCDM), pp 66–73 (2009). <https://doi.org/10.1109/MCDM.2009.4938830>. ID: 1
- Parragh SN, Tricoire F (2018) Branch-and-bound for bi-objective integer programming. arXiv preprint [arXiv:1809.06823](https://arxiv.org/abs/1809.06823)
- Peng G, Fang Y, Chen S, Peng W, Yang D (2016) A hybrid multi-objective discrete particle swarm optimization algorithm for cooperative air combat DWT. *Commun Comput Inf Sci* 682:114–119. [https://doi.org/10.1007/978-981-10-3614-9\\_15](https://doi.org/10.1007/978-981-10-3614-9_15)
- Ray T (2003) Golinski’s speed reducer problem revisited. *AIAA J* 41(3):556–558
- Reid CR, Latty T (2016) Collective behaviour and swarm intelligence in slime moulds. *FEMS Microbiol Rev* 40(6):798–806. <https://doi.org/10.1093/femsre/fuw033>
- Reid CR, Latty T, Dussutour A, Beekman M (2012) Slime mold uses an externalized spatial “memory” to navigate in complex environments. *Proc Natl Acad Sci USA* 109(43):17490–17494. <https://doi.org/10.1073/pnas.1215037109>
- Reid CR, Beekman M, Latty T, Dussutour A (2013) Amoeboid organism uses extracellular secretions to make smart foraging decisions. *Behav Ecol* 24(4):812–818. <https://doi.org/10.1093/beheco/art032>
- Reyes-Sierra M, Coello CC (2006) Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *Int J Comput Intell Res* 2(3):287–308
- Riquelme N, Lücken CV, Baran B (2015) Performance metrics in multi-objective optimization. In: Computing conference (CLEI), 2015 Latin American, pp. 1–11. IEEE
- Schütze O, Coello CAC, Mostaghim S, Talbi E-G, Dellnitz M (2008) Hybridizing evolutionary strategies with continuation methods for solving multi-objective problems. *Eng Optim* 40(5):383–402. <https://doi.org/10.1080/03052150701821328>
- Shirakawa T, Sato H, Ishiguro S (2015) Construction of living cellular automata using the physarum plasmodium. *Int J Gen Syst* 44(3):292–304. <https://doi.org/10.1080/03081079.2014.997531>
- Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol Comput* 2(3):221–248
- Sun J, Zhang H, Zhang Q, Chen H (2018) Balancing exploration and exploitation in multiobjective evolutionary optimization. In: Proceedings of the genetic and evolutionary computation conference companion, pp 199–200
- Tanabe R, Ishibuchi H, Oyama A (2017) Benchmarking multi-and many-objective evolutionary algorithms under two optimization scenarios. *IEEE Access* 5:19597–19619
- Tero A, Kobayashi R, Nakagaki T (2005) A coupled-oscillator model with a conservation law for the rhythmic amoeboid movements of plasmodial slime molds. *Physica D* 205(1–4):125–135. <https://doi.org/10.1016/j.physd.2005.01.010>
- Tian X, Pang W, Wang Y, Guo K, Zhou Y (2019) LatinPSO: an algorithm for simultaneously inferring structure and parameters of ordinary differential equations models. *Biosystems* 182:8–16
- Usman M, Pang W, Coghill GM (2020) Inferring structure and parameters of dynamic system models simultaneously using swarm intelligence approaches. *Memet Comput* 12(3):267–282
- Veldhuizen DAV, Lamont GB (2000) Multiobjective evolutionary algorithms: analyzing the state-of-the-art. *Evol Comput* 8(2):125–147
- Wang H (2015) Direct zigzag search for discrete multi-objective optimization. *Comput Oper Res* 61:100–109. <https://doi.org/10.1016/j.cor.2015.03.001>
- Wang S, Ma S (2018) Efficient methods for a bi-objective nursing home location and allocation problem: a case study. *Appl Soft Comput* 65:280–291
- Wong K-C, Leung K-S, Wong M-H (2010) Effect of spatial locality on an evolutionary algorithm for multimodal optimization. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol 6024, pp 481–490. LNCS. [https://doi.org/10.1007/978-3-642-12239-2\\_50](https://doi.org/10.1007/978-3-642-12239-2_50)
- Yang X-S (2010) Nature-inspired metaheuristic algorithms. Luniver Press (2010)
- Zhang Q, Zhou A, Zhao S, Suganthan PN, Liu W, Tiwari S (2008) Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms. Technical report, vol 264

- Zhang H, Zhang X, Gao X-Z, Song S (2016) Self-organizing multiobjective optimization based on decomposition with neighborhood ensemble. *Neurocomputing* 173:1868–1884
- Zhang Z, Gao C, Lu Y, Liu Y, Liang M (2016) Multi-objective ant colony optimization based on the physarum-inspired mathematical model for bi-objective traveling salesman problems. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0146709>
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
- Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput* 8(2):173–195
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. TIK-report 103
- Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VGD (2003) Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans Evol Comput* 7(2):117–132
- Zopounidis C, Pardalos PM (2010) *Handbook of multicriteria analysis*, vol 103. Springer
- Zouache D, Moussaoui A, Abdelaziz FB (2018) A cooperative swarm intelligence algorithm for multi-objective discrete optimization with application to the knapsack problem. *Eur J Oper Res* 264(1):74–88. <https://doi.org/10.1016/j.ejor.2017.06.058>
- Zou W, Zhu Y, Chen H, Zhang B (2011) Solving multiobjective optimization problems using artificial bee colony algorithm. *Discrete Dyn Nat Soc*. <https://doi.org/10.1155/2011/569784>

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