Swarm Intelligence (2026) 20:0
https://doi.org/10.1007/s11721-026-00262-9

Set-based particle swarm optimisation convergence

Gary J. Steyn[1] · **Jean-Pierre van Zyl**[1] · **Andries P. Engelbrecht**[1,2]

Received: 28 March 2025 / Accepted: 26 April 2026

Abstract

Set-based particle swarm optimisation is a swarm-based metaheuristic used to solve combinatorial and discrete optimisation problems. Combinatorial and discrete optimisation problems are fundamentally different from real-valued optimisation problems, and present a uniquely challenging problem to solve and as a result many metaheuristics are not suited to solve combinatorial and discrete optimisation problems. Recently published literature has shown set-based particle swarm optimisation to be very adept at solving a wide range of problems including the multi-dimensional knapsack problem, feature selection, portfolio optimisation, polynomial approximation, clustering, and rule induction. Despite these advancements, no comprehensive study exists on the convergence behaviour of the set-based particle swarm optimisation algorithm. This paper performs the first convergence study of the set-based particle swarm optimisation algorithm and outlines the shortcomings of the algorithm in its current form. Through mathematical proofs, empirical experiments, and sensitivity analysis it is found that set-based particle swarm optimisation does not converge, even under ideal conditions.

Keywords Set-based particle swarm optimisation · Swarm diversity · Convergence · Combinatorial optimisation

✉ Gary J. Steyn
23982691@sun.ac.za

Jean-Pierre van Zyl
vanzylj@sun.ac.za

Andries P. Engelbrecht
engel@sun.ac.za

[1] Division of Computer Science, Stellenbosch University, Victoria Street, Stellenbosch 6702, Western Cape, South Africa

[2] Department of Industrial Engineering, Stellenbosch University, Victoria Street, Stellenbosch 6702, Western Cape, South Africa

Published online: 26 May 2026

1 Introduction

Swarm intelligence (SI) methods are popular approaches used to solve difficult optimisation problems. Swarm-based meta-heuristics, *e.g.* particle swarm optimisation (PSO), are often used in cases where traditional optimisation techniques, *e.g.* linear programming, fail (Kennedy & Eberhart, 2001). One of the key aspects needed for the successful application of PSO lies in the control of the exploration-exploitation trade-off. The exploration-exploitation trade-off refers to the balance between searching for new promising areas of the search space, and refining already found favourable solutions. The exploration-exploitation trade-off is inextricably linked to the issue of swarm convergence (Shi & Eberhart, 1998a, b). The importance of the exploration-exploitation trade-off, and the effect of swarm convergence on the trade-off, has lead to extensive research into the convergence behaviour of the PSO algorithm (Cleghorn & Engelbrecht, 2015).

Although the convergence behaviour of PSO is well studied, the behaviour of the lesser-known set-based particle swarm optimisation (SBPSO) variant has no existing corpus of literature. A recent review paper showed that the SBPSO is one of the only PSO-variants for discrete optimisation problems (DOPs) which is entrenched in true set theory, instead of claiming to be "set-based" while making compromises like having fixed length position encoding (Van Zyl & Engelbrecht, 2023). As a result, SBPSO has emerged as a promising SI-based approach to DOPs, and exists alongside a scarce group of other notable discrete PSO variants, such as the geometric particle swarm optimisation (GPSO) and algebraic particle swarm optimisation (APSO) algorithms, proposed by Moraglio et al. (2007) and Baioletti et al. (2017), respectively. Even fewer PSO variants have been developed specifically for pseudo-Boolean combinatorial problems, including the APSO implementation introduced by Santucci et al. (2020), which operates over a bit-string representation of the search space. Therefore, it has become compelling to study the convergence behaviour of SBPSO for equivalent set-based formulations of these problems.

This paper presents the first study on the convergence behaviour of the SBPSO algorithm. The effect of the control parameters of SBPSO, *i.e.* c_1, c_2, c_3, and c_4, is studied and it is shown that the SBPSO does not converge. A mathematical proof shows that the exploration mechanism never allows a sufficient reduction in diversity to allow the swarm to converge, a result which is reproduced in empirical experiments which aim to force convergence. Further, sensitivity analysis (SA) is performed to illustrate the effect of each control parameter on different criteria related to swarm convergence.

The remainder of this paper is outlined as follows: Section 2 presents background information on the SBPSO algorithm and on PSO swarm convergence. Section 3 expands on the concepts of PSO swarm convergence to include set-based swarm convergence, after which toy experiments are performed in Section 4 as a proof of concept. Section 5 performs SA on the acceleration coefficients of SBPSO; finally, the paper is concluded in Section 6.

2 Background

This section provides relevant information regarding the typical behaviours of PSO algorithms, including explanations of convergence, stability and, stagnation. In addition, a brief overview is provided to explain the history and objectives of convergence research with respect to the canonical particle swarm optimisation (CPSO) algorithm. Information regarding SBPSO and SA is also included as context for the contents of the following sections.

The included subsections are stated as follows: Swarm behaviour, Related convergence research, SBPSO, and SA.

2.1 Swarm behaviour

CPSO is the simplest and most extensively studied variant of the PSO algorithm. Despite its apparent simplicity, theoretical analysis of the algorithm's behavior remains challenging due to several implicit characteristics. This difficulty is highlighted by Poli and Broomhead (2007) and can be summarized as follows:

1. PSO consists of many interacting agents,
2. The attractors of each particle evolve over time,
3. Particle dynamics are inherently stochastic, and
4. The behaviour of the algorithm is dependent on the objective function and the resulting search landscape.

The analysis of convergence becomes intractable when each of these characteristics are considered simultaneously. Consequently, convergence studies often rely on simplifying assumptions, e.g. assumptions of stagnation and deterministic particle dynamics.

To understand the findings of this article, as well as contextualize the existing research on CPSO convergence, it is necessary to explicitly differentiate between convergence, stability, and stagnation. The following definitions are provided by Cleghorn and Engelbrecht (2018), where (s_t) is taken to denote the sequence $(\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_t)$, such that $(\boldsymbol{s}_t) \in \mathbb{R}^n$ implies that $\boldsymbol{s}_t \in \mathbb{R}^n \ \forall \ t$.

Definition 1 *(Convergent sequence)* The sequence $(\boldsymbol{s}_t) \in \mathbb{R}^n$ is convergent if there exists an $\boldsymbol{s} \in \mathbb{R}^n$ such that

$$\lim_{t \to \infty} \boldsymbol{s}_t = \boldsymbol{s} \tag{1}$$

Definition 2 *(Order-1 stability)* The sequence $(\boldsymbol{s}_t) \in \mathbb{R}^n$ is order-1 stable if there exists an $\boldsymbol{s}_{\mathbb{E}} \in \mathbb{R}^n$ such that

$$\lim_{t \to \infty} \mathbb{E}[\boldsymbol{s}_t] = \boldsymbol{s}_{\mathbb{E}} \tag{2}$$

where $\mathbb{E}[\boldsymbol{s}_t]$ is the expectation of $\boldsymbol{s}_t$.

Definition 3 *(Order-2 stability)* The sequence $(\boldsymbol{s}_t) \in \mathbb{R}^n$ is order-2 stable if there exists an $\boldsymbol{s}_{\mathbb{V}} \in \mathbb{R}^n$ such that

$$\lim_{t \to \infty} \mathbb{V}[\boldsymbol{s}_t] = \boldsymbol{s}_{\mathbb{V}} \tag{3}$$

where $\mathbb{V}[\boldsymbol{s}_t]$ is the variance of $\boldsymbol{s}_t$.

For unimodal problems, swarm convergence describes the tendency of particle positions to approach a single common point in the search space over time. Swarm diversity describes the difference between particle solutions with respect to the decision space of an optimisation problem, and is frequently used as an indication of swarm convergence.

In the study by Bosman et al. (2014), commonalities were observed in the appearance of the diversity graphs produced by nine continuous PSO variants over experiments on a set of 20 benchmark problems. Specifically, the diversity graphs reflected two general trends, referred to as *phase 1* and *phase 2*. The transition between phase 1 and phase 2 represents a broad shift in the behaviour of the swarm from exploration towards exploitation. As described by Bosman et al. (2014), phase 1 is characterised by high initial diversity due to random initialization of particles, followed by a rapid decrease over subsequent iterations. Phase 2 indicates a continued decrease in diversity, albeit at a slower rate than in phase 1.

Definition 1 describes absolute convergence, which can only ever be attained if particles are assumed to have deterministic dynamics. In the stochastic case, convergence is usually defined with respect to order-1 and order-2 stability (Poli, 2009).

Stability describes the consistency with which particles approach a steady state that adheres to the relevant definition of convergence. This differs from stagnation, which describes a swarm state wherein particles are unable to improve their positions or explore new regions of the search space.

In this study, *stagnation* is used to suggest that the state of the swarm remains unchanged over one or more consecutive iterations. For example, the swarm is said to have stagnated between iterations t and $(t-1)$ if $\boldsymbol{x}_i(t) = \boldsymbol{x}_i(t-1)$, $\boldsymbol{y}_i(t) = \boldsymbol{y}_i(t-1)$, and $\hat{\boldsymbol{y}}_i(t) = \hat{\boldsymbol{y}}_i(t-1)$ for particles $i = 1, \ldots, n_s$.

2.2 Related convergence research

This section discusses the necessity of convergence of PSO swarms and provides background on existing convergence research in real-valued PSO.

2.2.1 The necessity of convergence

The importance of convergence in PSO is typically motivated by two principal considerations: performance enhancement and practical utility from the perspective of the practitioner. Witt (2009) and Leonard et al. (2015) showed that convergence does not always guarantee optimal performance, each focusing on the guaranteed convergence particle swarm optimisation (GCPSO) and angle modulated particle swarm optimisation (AMPSO) respectively. Witt (2009) outlined the issue with performing theoretical runtime analysis on PSO swarms which do not converge, while Leonard et al. (2015) outlined the effect of "inadequate convergence" where particle positions tend to stabilise despite large particle velocities. However, empirical studies have demonstrated that theoretically convergent control parameter configurations in CPSO often yield better results than unconstrained alternatives (Cleghorn & Engelbrecht, 2016; Harrison et al., 2017). In particular, Cleghorn and Engelbrecht (2016) found that theoretically unstable parameters frequently cause CPSO to perform worse than random search, while such outcomes are significantly less likely when stable, convergent configurations are used.

From a practical standpoint, convergence is particularly relevant in real-world problem-solving contexts, since swarm-based optimisation algorithms need to exhibit predictable and reliable search behaviour. Without convergence, the algorithm continues to search indefinitely, which makes it challenging or impossible to determine whether further improvements to the best-found solutions are probable. This uncertainty undermines the utility of the optimiser because it cannot be shown to satisfy the formal definitions of a local or global search algorithm (Van den Bergh & Engelbrecht, 2006). Moreover, unstable parameter configurations can potentially lead to erratic particle dynamics that prevent convergence and degrade performance. For example, in the case of CPSO, it is conceivable that excessively large control parameter values may induce disproportionately large initial step sizes of particles, which could immediately displace the particle attractors far outside the established search bounds and lead to exacerbated particle roaming behavior (Engelbrecht, 2013). Such roaming particles perform fruitless exploration, which wastes computational resources, hinders swarm convergence, and prevents productive refinement of the existing solutions.

Van den Bergh and Engelbrecht (2006) showed that many PSO variants do not consistently operate as effective local or global search algorithms; however, convergence remains a foundational property for the development and further refinement of PSOs. Convergence dispels any doubt about the potential of the optimiser to locate a local or global optimum once the swarm has converged. For example, GCPSO was proposed as an alternative to CPSO that ensures convergence at least to a local optimum (Van den Bergh & Engelbrecht, 2006). Furthermore, Blackwell and Branke (2004) demonstrated that knowledge of a PSOs tendency to prematurely converge can be crucial for the development of strategies to offset the performance-degrading effect of diversity loss in non-stationary environments.

In addition, convergence facilitates more rigorous analysis of algorithmic run-time (Witt, 2009) and supports the design of principled stopping criteria. Hence, convergence enables the formulation of well-founded decisions about when to terminate the algorithm. As outlined by Zielinski and Laur (2007), stopping conditions are often defined with respect to the following types of criteria: either improvement-based or distribution-based. Improvement-based conditions terminate the search when no improvement is observed in particle positions (or personal best solutions) over a predefined number of iterations. Alternatively, distribution-based conditions end the search when particle diversity in the decision space indicates that the swarm has converged. In cases where a fixed number of allocated search iterations is not used to define the stopping condition, non-convergent PSO implementations must typically rely on improvement-based stopping criteria, which require arbitrary threshold parameters and may not reliably indicate whether meaningful progress is still possible.

Finally, understanding convergence behaviour enables the establishment of dependable performance assessments on benchmark problems. Swarm configurations which converge for a benchmark problem, and consistently provides similar solutions, allow trustworthy comparisons to be made between different PSO variants or algorithmic implementations (Van den Bergh & Engelbrecht, 2006).

In the context of discrete pseudo-Boolean optimisation problems, the rugged and discontinuous nature of the fitness landscape often makes it difficult for algorithms to perform fine-grained refinements on the best-known solutions. Such landscapes are characterised by numerous local optima, which can trap search algorithms and hinder incremental improvements (Ochoa et al., 2014). Nevertheless, convergence remains essential for set-based PSO algorithms, as the global optimum corresponds to a specific finite subset of elements, and the global best solution represents the closest current approximation to this optimal set.

Ultimately, convergence remains valuable in set-based PSO algorithms because it allows the search process to focus on the most promising elements that can be used to construct a solution. As the swarm transitions from exploration to exploitation, these high-quality elements are increasingly incorporated into particle positions, refining the search around the global best solution. As in continuous PSO, maintaining non-convergent behaviour in set-based PSO variants is beneficial for dynamic optimisation problems, since premature convergence reduces swarm diversity and limits the ability of the algorithm to track and adapt to changes in the fitness landscape.

2.2.2 Convergence analysis

The necessary and sufficient conditions for CPSO convergence to equilibrium state were derived independently by Trelea (2003) and Van den Bergh and Engelbrecht (2006) under the same set of assumptions. These similar results were later reconciled by Cleghorn and Engelbrecht (2014a), who determined identical convergence conditions under less restrictive assumptions.

The primary objective of CPSO convergence research is to identify the least conservative set of bounded constraints on w, c_1, and c_2 that ensures convergence, while operating under the least restrictive, and fewest number of assumptions. Two main approaches have emerged to derive such bounds. The first approach, introduced by Kadirkamanathan et al. (2006), derives conservative bounds based on Lyapunov stability conditions. The second approach, developed independently by Poli and Broomhead (2007); Poli (2009) and Jiang et al. (2007), derives parameter bounds without relying on Lyapunov analysis. These bounds are henceforth referred to as Poli's bounds.

Both approaches have been extensively scrutinized, with significant research focusing on extending the parameter bounds, deriving them under fewer or alternative assumptions, or validating them empirically. For example, Gazi (2012) extended the bounds of Kadirkamanathan et al. (2006), while Liu (2015) rederived Poli's bounds under different assumptions, showing that the convergence region remains invariant regardless of the social network topology employed by CPSO. Empirical support for the practical use of Poli's bounds can be found in Cleghorn and Engelbrecht (2014b) and Harrison et al. (2017).

Many of the original stability studies recognized the relationship between convergence and the effects of component-wise velocity magnitudes on particle trajectories, for example in the study by Clerc and Kennedy (2002). This relationship is particularly relevant when considering time-invariant self-adaptive PSO algorithms, which rely on introspective feedback from the search process to modify control parameter values, or other aspects of the algorithm, that govern swarm behaviour. A review and empirical convergence analysis of 18 such self-adaptive PSO algorithms is provided by Harrison et al. (2018).

2.3 Set-based particle swarm optimisation

The SBPSO is a variant of the original PSO developed by Langeveld and Engelbrecht (2012) which utilises pure sets to represent particle positions and velocities. The use of set-based particles allows for variable-length solution encodings and makes the SBPSO well-suited for application in combinatorial domains (Van Zyl & Engelbrecht, 2023). The SBPSO algorithm has been used to solve a myriad of DoPs, including the multi-dimensional knapsack problem (MKP) (Langeveld & Engelbrecht, 2012), feature selection (Engelbrecht et al., 2019), portfolio optimisation (Erwin & Engelbrecht, 2021, 2023; Erwin et al., 2020), polynomial approximation (Van Zyl & Engelbrecht, 2021), clustering (Brown & Engelbrecht, 2022; Wet et al., 2023), and rule induction (Van Zyl & Engelbrecht, 2022).

In the subsequent discussion and definitions, the following notation is used: $\mathcal{P}(\mathcal{U})$ denotes the power set of $\mathcal{U}$ (*i.e.* the set containing all possible subsets of $\mathcal{U}$), and $A \times B$ denotes the Cartesian product between the sets A and B. The SBPSO solves problems defined by a universal set, $\mathcal{U} = \{e_n\}_{n \in \{1,\ldots,N_{\mathcal{U}}\}}$, where $N_{\mathcal{U}}$ is the number of elements in $\mathcal{U}$; particle positions, $X_i(t) \subseteq \mathcal{U}$, are used to represent candidate solutions. A particle position is modified using a velocity set, $V_i(t) \subseteq \{+,-\} \times \mathcal{U}$, which dictates the elements that must be added and removed from the current position.

Definition 4 *(The addition of two velocities)* The addition of two velocities, $V_1 \oplus V_2$, is a mapping, $\oplus : \mathcal{P}(\{+,-\} \times \mathcal{U})^2 \Rightarrow \mathcal{P}(\{+,-\} \times \mathcal{U})$, that takes two velocities and yields one velocity. Implemented as set operations, a velocity added to a velocity is interpreted as the union operator:

$$V_1 \oplus V_2 = V_1 \cup V_2. \tag{4}$$

Definition 5 *(The difference between two positions)* The difference between two positions, $X_1 \ominus X_2$, is a mapping, $\ominus : \mathcal{P}(\mathcal{U})^2 \Rightarrow \mathcal{P}(\{+,-\} \times \mathcal{U})$, that takes two positions and yields a velocity. The result is effectively the set operation steps which are required to convert X_2 into X_1:

$$X_1 \ominus X_2 = (\{+\} \times (X_1 \backslash X_2)) \cup (\{-\} \times (X_2 \backslash X_1)). \tag{5}$$

Definition 6 *(The scalar multiplication of a velocity)* The scalar multiplication of a velocity, $\eta \otimes V$, is a mapping, $\otimes : [0,1] \times \mathcal{P}(\{+,-\} \times \mathcal{U}) \Rightarrow \mathcal{P}(\{+,-\} \times \mathcal{U})$, which takes a scalar and a velocity, and yields a velocity. The mapping results in a randomly selected subset of size $\lfloor \eta \times |V| \rfloor$ from V and is expressed as

$$\eta \otimes V = \text{random subset}(V, \eta). \tag{6}$$

Note that $0 \otimes V = \emptyset$ and $1 \otimes V = V$.

Definition 7 *(The addition of a velocity and a position)* The addition of a velocity and a position, $X \boxplus V$, is a mapping, $\boxplus : \mathcal{P}(\mathcal{U}) \times \mathcal{P}(\{+,-\} \times \mathcal{U}) \Rightarrow \mathcal{P}(\mathcal{U})$, that takes a position and velocity and yields the resultant position. The operation is expressed as

$$X \boxplus V = V(X), \tag{7}$$

which involves the application of the operation associated with each v_i from $V = \{v_1, \ldots, v_n\}$ to X by adding or removing each e_i as dictated by the elements in the velocity.

Definition 8 *(The addition of elements)* The addition of elements, $\beta \odot_k^+ A$, to a position *X*(*t*) where A is shorthand for $\mathcal{U} \backslash (X(t) \cup Y(t) \cup \hat{Y}(t))$, is a mapping $\odot_k^+ : [0, |A|] \times \mathcal{P}(\mathcal{U}) \Rightarrow \mathcal{P}(\{+,-\} \times \mathcal{U})$ that takes a scalar and a set of elements and yields a velocity. The operator is implemented by randomly selecting a subset of elements from A, with a size determined by β, to be added to *X*(*t*):

$$\beta \odot_k^+ A = \{+\} \times k\text{-Tournament Selection}(A, N_{\beta,A}), \tag{8}$$

where $N_{\beta,A}$ is the number of elements to be added to *X*(*t*) as defined in Equation (10) and *k* is a user-defined parameter. The pseudocode for the k-tournament selection procedure in a minimisation problem is provided in Algorithm 1.

Input: Set of candidates A, number of selections $N_{\beta,A}$
Output: Selected set V_+
Initialise lists e and s of length k;
$V_+ \leftarrow \emptyset$;
for $n = 1, \ldots, N_{\beta,A}$ **do**
    for $j = 1, \ldots, k$ **do**
        Randomly select $e_j \in A$;
        $s_j \leftarrow f(X_i(t) \cup e_j)$;
    end
    $m \leftarrow \text{argmin}_j(s)$;
    $V_+ \leftarrow V_+ \oplus (\{+\} \times e_m)$;
end
return V_+

Algorithm 1 k-Tournament selection

Definition 9 *(The removal of elements)* The removal of elements, $\beta \odot^- S$, from a position *X*(*t*), where S is shorthand for $X(t) \cap Y(t) \cap \hat{Y}(t)$, is the mapping, $\odot^- : [0, |S|] \times \mathcal{P}(\mathcal{U}) \Rightarrow \mathcal{P}(\{+,-\} \times \mathcal{U})$, that takes a scalar and a set of elements, and yields a velocity. The operator is implemented by randomly selecting a subset of elements from S, with a size determined by β, to be removed from *X*(*t*):

$$\beta \odot^- S = \{-\} \times \left(\frac{N_{\beta,S}}{|S|} \otimes S \right). \tag{9}$$

The number of elements selected, $N_{\beta,S}$, is defined as

$$N_{\beta,S} = \min \left\{ |S|, \lfloor \beta \rfloor + 1_{\{r < \beta - \lfloor \beta \rfloor\}} \right\}, \tag{10}$$

for a random number $r \sim U(0,1)$; $1_{\{\text{bool}\}}$ is 1 if bool is true and 0 if bool is false.
Using the definitions given above, the position and velocity update equations are defined. A particle position, X_i, is updated as

$$X_i(t+1) = X_i(t) \boxplus V_i(t+1), \tag{11}$$

where $\boxplus$ has the function defined in Definition 7. Additionally the velocity, V_i, of a particle is updated as

$$\begin{aligned} V_i(t+1) = c_1 r_{1i}(t) \otimes (Y_i(t) \ominus X_i(t)) \oplus \\ c_2 r_{2i}(t) \otimes \big(\hat{Y}_i(t) \ominus X_i(t)\big) \oplus \\ \big(c_3 r_{3i}(t) \odot_k^+ A_i(t)\big) \oplus \\ \big(c_4 r_{4i}(t) \odot^- S_i(t)\big) , \end{aligned} \tag{12}$$

where $S_i(t)$ and $A_i(t)$ are determined by following Definitions 9 and 8 respectively.

There are four components to the velocity update equation, with the influence of each controlled by an acceleration coefficient, c_k. Each c_k remains constant for all particles with $c_1, c_2 \in [0,1]$ and $c_3, c_4 \in [0, N_{\mathcal{U}}]$, and each r_{1i}, r_{2i}, r_{3i}, r_{4i} independently drawn from the distribution $U(0,1)$. The coefficient c_1 controls the attraction to the personal best position $(Y_i(t) \ominus X_i(t))$, the coefficient c_2 controls the attraction to the global best position $(\hat{Y}_i(t) \ominus X_i(t))$. Additionally, the coefficient c_3 manages the level of influence of the addition operator $(\odot_k^+ A_i(t))$ and the coefficient c_4 governs the effect of the removal operator $(\odot^- S_i(t))$.

The attractions to the personal and global best positions are concepts from the real-valued PSO. These two attractors are, fundamentally, exploitation mechanisms since these two velocity components encourage particles to search areas of the problem space which have previously shown promise. The swarm of the real-valued PSO is encouraged to explore new areas of the search space through the use of a momentum term. However, no equivalent analogy for the momentum term in the SBPSO exists, since the concept of direction does not exist in a set-based environment. As a result, the addition and removal operators are used to inject diversity into the swarm and encourage exploration. The pseudocode for SBPSO is thus provided in Algorithm 2.

```
Input: Objective function f, universal set 𝒰
Output: Best solution found Ŷ_i
Let n_s be the number of particles in the swarm;
for i = 1, ..., n_s do
    Let X_i(0) ← a small subset of 𝒰;
    Let V_i(0) ← ∅;
    Calculate f(X_i(0));
    Let Y_i(0) ← X_i(0);
    Let f(Y_i(0)) ← ∞;
    Let f(Ŷ_i(0)) ← ∞;
while stopping conditions are not met do
    for i = 1, ..., n_s do
        if f(X_i(t)) < f(Y_i(t)) then
            Y_i(t) ← X_i(t);
        if f(X_i(t)) < f(Ŷ_i(t)) then
            Ŷ_i(t) ← X_i(t);
    for i = 1, ..., n_s do
        Update V_i(t) according to Equation (12);
        Update X_i(t) according to Equation (11);
    t ← t + 1;
return Ŷ_i
```

Algorithm 2 Set-based particle swarm optimisation

2.4 Sensitivity analysis

SA is the study of the relation of the uncertainty in the model output to the uncertainty to the model input (Saltelli et al., 2004). There are two main approaches to SA, namely local sensitivity analysis (LSA) and global sensitivity analysis (GSA), of which GSA is preferred due to LSA being only informative at the base point where the derivatives are computed (Saltelli et al., 2008). GSA methods can further be classified as non-parametric, variance-based, or density-based methods (Borgonovo & Plischke, 2016).

A popular variance-based GSA method is Sobol' SA. Sobol' SA performs quasi-Monte Carlo (MC) sampling, referred to as Sobol' sequences (SS) sampling, on which analysis of variance (ANOVA) is performed and the Sobol' sensitivity indices are calculated. Borgonovo (2007) proposed a density-based sensitivity indicator which compares the probability density function (PDF) of the output variable (f_Y) against the PDF of the output variable when conditioned on one of the input variables ($f_{Y|X_i}$), *i.e.* the sensitivity indicator calculates the area between the two PDFs, as

$$s(X_i) = \int_{-\infty}^{\infty} |f_Y(y) - f_{Y|X_i}(y)| dy. \tag{13}$$

The difference between the two PDFs is then used to calculate the moment independent delta sensitivity index, δ_i, for variable X_i as

$$\delta_i = \frac{1}{2}\mathbb{E}_{X_i}\left[s(X_i)\right], \tag{14}$$

where $\mathbb{E}_{X_i}\left[s(X_i)\right]$ is the expected value of the area between the two PDFs. The density-based approach by Borgonovo (2007) was expanded by Plischke et al. (2013) to include a framework that can estimate both variance-based and density-based sensitivity indices; this includes both the delta index and Sobol' index.

The specific implementation of Plischke's estimation for Sobol' indices used in this paper is | SALib|[1] by Herman and Usher (2017); Iwanaga et al. (2022). The background literature for the implementation is summarised in the remainder of this subsection, as defined by Plischke et al. (2013).

Pearson (1905) introduced the correlation ratio, η_i^2, which serves as the basis for several global sensitivity indices, most notably those proposed by Sobol (1993) and Wagner (1995), despite differences in their underlying assumptions. The correlation ratio is defined as

$$\eta_i^2 = \frac{\mathbb{V}\left[\mathbb{E}\left[Y|X_i\right]\right]}{\mathbb{V}\left[Y\right]} = 1 - \frac{\mathbb{E}\left[\mathbb{V}\left[Y|X_i\right]\right]}{\mathbb{V}\left[Y\right]}, \tag{15}$$

where $\mathbb{E}$ denotes the expected value and $\mathbb{V}$ denotes the variance.

The density-based indices are found by partitioning the input space into M classes, from which the conditional PDFs are calculated. Given the input space $\mathcal{X}$, output space $\mathcal{Y}$, and function $g : \mathcal{X} \to \mathcal{Y},\ g(\boldsymbol{x}) = y$, the following is defined in order to determine the sensitivity of g on $\mathcal{X}$ with respect to $\mathcal{Y}$. A subspace of $\mathcal{X}$ is defined by fixing input dimension $i \in \{1, \ldots, d\}$, which results in the subspace $\mathcal{X}_i$. Additionally, let X_i be a realisation of the random vectors on the space $\mathcal{X}_i$, which means that vectors used as input to g are sampled from this realisation (*i.e.* $\boldsymbol{x} \in X_i$). The subspace $\mathcal{X}_i$ is partitioned into M classes as $\mathcal{P} = \{\mathcal{C}_m | m = 1, \ldots, M\}$ with $\left(\bigcup_{m=1}^{M} \mathcal{C}_m = \mathcal{X}_i\right) \wedge (\mathcal{C}_m \cap \mathcal{C}_{m\prime} = \emptyset \mid m \neq m\prime)$. The probability that X_i falls into a class $\mathcal{C}_m$ is defined as the integral over the partition $\mathcal{P}$ of the PDF f_{X_i}:

$$P_{X_i}(\mathcal{C}_m) = \int_{\mathcal{P}} f_{X_i}(\boldsymbol{x}) d\boldsymbol{x}. \tag{16}$$

The class-conditional density of the output variable given the partitioning is defined as a result of the total probability theorem as

$$f_{Y|\mathcal{C}_m}(y) = \frac{\int_{\mathcal{P}} f_{Y|X_i}(y) \cdot f_{X_i}(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\mathcal{P}} f_{X_i}(\boldsymbol{x}) d\boldsymbol{x}} = \frac{\int_{\mathcal{P}} f_{X_i,Y}(\boldsymbol{x}, y) d\boldsymbol{x}}{P_{X_i}(\mathcal{C}_m)} \tag{17}$$

where the shorthand $f_{Y|X_i}(y) = f_{Y|X_i=\boldsymbol{x}}(y)$ is used.

To obtain the expected value of the output conditioned on a realisation of the subspace of $\mathcal{X}_i$, the integral of $f_{Y|X_i}$ is calculated as

[1] https://github.com/salib/salib

$$\mathbb{E}\left[Y|X_i\right] = \int_{\mathcal{Y}} y \cdot f_{Y|X_i}(y)dy \tag{18}$$

For the calculation of the variance of the expected value, consider the weighted L^2-distance between $\mathbb{E}\left[Y|X_i\right]$ and $\mathbb{E}\left[Y\right]$, *i.e.*

$$\begin{aligned} &\int \left(\int \left(y \cdot f_{Y|X_i}(y) - y \cdot f_Y(y) \right) dy \right)^2 \cdot f_{X_i}(\boldsymbol{x})dx \\ &= \int \left(\mathbb{E}\left[Y|X_i\right] - \mathbb{E}\left[Y\right] \right)^2 f_{X_i}(x)dx \\ &= \mathbb{V}\left[\mathbb{E}\left[Y|X_i\right]\right]. \end{aligned} \tag{19}$$

This L^2-distance is equivalent to the numerator in Equation (15).

From the partitions defined for $\mathcal{P}$, the correlation ratios can be calculated by directly counting the values in each partition, instead of through integration:

$$\hat{\eta}_i^2 = \frac{\sum_{m=1}^{M} n_m(\bar{y}_m - \bar{y})^2}{\sum_{j=1}^{n_Y}(y_j - \bar{y})^2}, \tag{20}$$

where $\bar{y}_m = \frac{1}{n_m}\sum_{j;x_j \in \mathcal{C}_m} y_i$, $n_m \sum_{j;x_j \in \mathcal{C}_m} 1$ is the number of realisations in class $\mathcal{C}_m$, and n_Y is the total number of realisations in all partitions. The first-order Sobol' indices are estimated from the correlation ratios counted by partition in Equation (20).

3 Set-based swarm convergence

At the time of writing this paper, there is scant published research on the analysis of the behaviour of the SBPSO algorithm. The studies by Langeveld and Engelbrecht (2012), and Wet et al. (2023), include respective empirical studies on the relationship between distinct control parameter configurations and performance in solving multi-dimensional knapsack and clustering problems. In contrast to the standard PSO algorithm, which has been thoroughly scrutinized theoretically and empirically, there are aspects of SBPSO that require further analysis to be understood in a meaningful capacity. Convergence is arguably the foremost of these, and is the subject of the theoretical study in this section.

To prove that a particular control parameter configuration induces convergence in standard PSO, it is necessary to demonstrate that the parameter values adhere to a set of necessary and sufficient convergence conditions. Instead of following this approach, this study assumes that convergence is not guaranteed, except in the trivial case where $c_3 = c_4 = 0$. The objective is therefore to infer the necessary convergence conditions for SBPSO, and to prove by contradiction that they are not expected to hold in the non-trivial case.

3.1 Generalized convergence for sequences of sets

Let $(R_t){:=}(R_1, R_2, \ldots, R_t)$ denote a sequence of sets. The sequence is defined with respect to $\mathcal{U}$ as $(R_t) \in \mathcal{U}$, with each $R_t \subseteq \mathcal{U}\ \forall\ t$. Further let there exists a constant set $R \subseteq \mathcal{U}$. To define convergence of the former sequence in terms of the constant set, it is necessary to generalize the convergence definitions of real-valued sequences to sets. The following definitions are the set-based equivalents of the well established convergence definitions given in Definitions 1, 2, and 3. The analogies for set-based convergence are defined in terms of the symmetric difference (*i.e.* $Q\Delta T = (T \setminus Q) \cup (Q \setminus T)$).

Definition 10 *(Set-based Convergent Sequence)* A sequence $(R_t) \in \mathcal{U}$ is convergent if there exists a constant set $R \subseteq \mathcal{U}$ such that

$$\lim_{t\to\infty} |R_t \Delta R| = 0, \tag{21}$$

Definition 11 *(Set-based Order-1 Stability)* The sequence $(R_t) \subseteq \mathcal{U}$ is order-1 stable if there exists $R_{\mathbb{E}} \subseteq \mathcal{U}$ such that

$$\lim_{t\to\infty} \mathbb{E}[|R_t \Delta R_{\mathbb{E}}|] = 0, \tag{22}$$

where $\mathbb{E}[R_t]$ is the expectation of R_t.

Definition 12 *(Set-based Order-2 Stability)* The sequence $(R_t) \subseteq \mathcal{U}$ is order-2 stable if there exists $R_{\mathbb{V}} \subseteq \mathcal{U}$ such that

$$\lim_{t\to\infty} \mathbb{V}[|R_t \Delta R_{\mathbb{V}}|] = 0, \tag{23}$$

where $\mathbb{V}[R_t]$ is the variance of R_t.
The remainder of this section comprises a mathematical proof and related derivations, which suffice as an argument to disprove the possibility of absolute convergence in SBPSO, except in the trivial case where c_3 and c_4 are equal to 0.

3.2 Set-based particle swarm optimization convergence

The use of order-1 and order-2 stability conditions in real-valued PSO is motivated by the fact that small perturbations in positions do not affect the objective function values of those positions too greatly. For an objective function f and a position $\boldsymbol{x}$, this is expressed as

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x} + \boldsymbol{\epsilon}),\ \boldsymbol{\epsilon} \ll \mathbf{1}. \tag{24}$$

However, in a combinatorial domain this assumption does not hold. The addition or removal of a singleton from a particle position can have an immense impact on the objective function value of that particle. Because this assumption does not hold, it is necessary to achieve absolute convergence in the SBPSO, not simply stability.

In order for a SBPSO swarm to converge, all particle positions must adhere to the condition described in Equation (21). For this condition to be met, each particle i must have a sequence of unchanging positions, *i.e.* $\lim_{t\to\infty} |X_i(t)\Delta X_i(t+1)| = 0$. The only way in which positions can remain unchanged, is if the velocity of a particle is the empty set, *i.e.* $V_i(t) = \emptyset$.

3.2.1 Particle immobilisation

There is one (highly unlikely) case worthy of note which does not result in swarm convergence, but in particle immobilisation.

Given that during the search process it happens that the particle position becomes the complement of the personal best position, $X_i = Y_i^{\complement}$, or that the particle position becomes the complement of the global best position, $X_i = \hat{Y}_i^{\complement}$, both the addition and removal operators return empty sets; to illustrate, consider the case when $X_i = Y_i^{\complement}$. From Definition 8, the set

$$\begin{aligned} A &= \mathcal{U}\backslash(X(t) \cup Y(t) \cup \hat{Y}(t)) \\ &= \mathcal{U}\backslash(Y^{\complement}(t) \cup Y(t) \cup \hat{Y}(t)) \\ &= \mathcal{U}\backslash\mathcal{U} \\ &= \emptyset. \end{aligned}$$

Similarly from Definition 9, the set

$$\begin{aligned} S &= X(t) \cap Y(t) \cap \hat{Y}(t) \\ &= Y^{\complement}(t) \cap Y(t) \cap \hat{Y}(t) \\ &= \emptyset. \end{aligned}$$

In addition to the position being the complement of either the personal best position or global best position, assume that $c_1 \leq \frac{1}{N_\mathcal{U}}$ and $c_2 \leq \frac{1}{N_\mathcal{U}}$. Since the random variables in the velocity update equation have the range $0 \leq r_{i1}, r_{i2} < 1$, it follows $c_1 r_{1i}(t), c_2 r_{2i}(t) \leq \frac{1}{N_\mathcal{U}}$. The scaling operator from Definition 6 takes the floor, which ensures that $c_1 r_{1i}(t) \otimes (Y_i(t) \ominus X_i(t)) = \emptyset$ and $c_2 r_{2i}(t) \otimes (\hat{Y}_i(t) \ominus X_i(t)) = \emptyset$.

If these two conditions are satisfied simultaneously, the particle will become immobilised and no longer contribute to the search. However, an immobilised particle is not considered to have fulfilled the conditions for convergence. Swarm convergence is considered to be a positive phenomenon, since convergence represents the phase of the search process after which the swarm has explored sufficiently (phase 1) and exploited sufficiently (phase 2). An immobilised particle no longer searches the decision space regardless of the state of the search process; that is, an immobilised particle stagnates even if the swarm is currently in the exploration phase.

3.2.2 The conditions for convergence

The velocity of a particle consists of four components, and each component needs to be the empty set. Assume the ideal situation where the swarm has converged to a global optimum, and therefore that stagnation has occurred. It follows, for each particle *i*, that $X_i = Y_i = \hat{Y}_i$. Since $X_i = Y_i$, the personal best attraction component of the velocity update equation will be empty (*i.e.* $c_1 r_{1i}(t) \otimes (Y_i(t) \ominus X_i(t)) = \emptyset$). Similarly due to $X_i = \hat{Y}_i$, the global best attraction component will also be empty (*i.e.* $c_2 r_{2i}(t) \otimes (\hat{Y}_i(t) \ominus X_i(t)) = \emptyset$).

It is reiterated that $N_{\beta,S}$ and $N_{\beta,A}$ denote the number of elements to be removed from or added to the particle position X_i, respectively, by the element-removal and element-addition operators. These quantities are defined as

$$N_{\beta,S} = \min\left\{|S|, \lfloor\beta\rfloor + 1_{\{r<\beta-\lfloor\beta\rfloor\}}\right\}$$

and

$$N_{\beta,A} = \min\left\{|A|, \lfloor\beta\rfloor + 1_{\{r<\beta-\lfloor\beta\rfloor\}}\right\},$$

where $1_{\{r<\beta-\lfloor\beta\rfloor\}}$ is an indicator function that evaluates to 1 if $r < \beta - \lfloor\beta\rfloor$, and 0 otherwise, with $r \sim U(0,1)$ and β being a scalar. In particular, β_A and β_S are used in the following sections to denote the respective β-variables used to compute $N_{\beta,A}$ and $N_{\beta,S}$, where $\beta_A = r \cdot c_3$ and $\beta_S = r \cdot c_4$, $r \sim U(0,1)$. Similarly, the notation 1_A and 1_S is used to denote the respective indicator functions used to compute $N_{\beta,A}$ and $N_{\beta,S}$. Finally, the additional notation $\beta_{i,A}$, $\beta_{i,S}$, $1_{i,A}$ and $1_{i,S}$ is used to indicate the aforementioned variables specifically as they are considered for particle *i*.

The cardinality of the addition and removal operators are not affected by the state of convergence. In order for $(c_3 r_{3i}(t) \odot_k^+ A_i(t)) = \emptyset$ and $(c_4 r_{4i}(t) \odot^- S_i(t)) = \emptyset$, $N_{\beta,S}(t) = N_{\beta,A}(t) = 0$ as $t \to \infty$. However, the authors show in Section 3.3 that the expected value of the cardinality of the addition and removal operators never approaches zero.

3.3 Expected cardinality of addition and removal operators

SBPSO utilizes two operators to facilitate exploration, namely the addition and removal operators. For absolute convergence to occur, it must necessarily hold that there exists some iteration in the search process wherein each particle permanently ceases to observe additions and removals of elements by the two exploration operators during the position update step. Therefore, the conditions for absolute convergence are derived as follows:

Assume that

$$N_{\beta,S}(t) = N_{\beta,A}(t) = 0, \tag{25}$$

as $t \to \infty$.

Equation (25) is expanded and reorganized as

$$0 = \min(|A|, \lfloor\beta_A\rfloor + \mathbb{1}_{\{r<\beta_A-\lfloor\beta_A\rfloor\}})$$
$$= \min(|S|, \lfloor\beta_S\rfloor + \mathbb{1}_{\{r<\beta_S-\lfloor\beta_S\rfloor\}}),$$

where $\beta_A = r \cdot c_3$ and $\beta_S = r \cdot c_4$, $r \sim U(0, 1)$. Let $\mathbb{1}_A$ and $\mathbb{1}_S$ denote $\mathbb{1}_{\{r<\beta_A-\lfloor\beta_A\rfloor\}}$ and $\mathbb{1}_{\{r<\beta_S-\lfloor\beta_S\rfloor\}}$, respectively. It follows from the previous result that at least one of the following two equations must hold:

$$0 = \min(|S| + \lfloor\beta_A\rfloor + \mathbb{1}_A, |A| + \lfloor\beta_S\rfloor + \mathbb{1}_S) \tag{26}$$

$$0 = \min(|S| + |A|, \lfloor\beta_S\rfloor + \mathbb{1}_S + \lfloor\beta_A\rfloor + \mathbb{1}_A). \tag{27}$$

In the following explanation, the operands of the min-function are referenced in order from left to right:

3.3.1 Suppose Equation (26) holds

It is therefore implied that the global best solution is equal to $\emptyset$ or $\mathcal{U}$. If this condition holds, it further implies that only one of either the first or second operands of the min-function will ever equal 0, and this same operand must equal 0 for each particle. This follows from the fact that $|S|$ and $|A|$ can never simultaneously equal 0 when $N_{\mathcal{U}} > 0$. Furthermore, in the operand where $|S|$ or $|A|$ equals 0, it must also hold, for each particle, that the corresponding β-value is strictly less than 1, and the indicator function ($\mathbb{1}_{\{r<\beta-\lfloor\beta\rfloor\}}$) should resolve to 0. It is motivated that the probability of this happening is infinitesimal when n_s is sufficiently large, because the probability of $\mathbb{1}$ resolving to 0 for each particle is at most $r_i^{n_s}$ where $r_i \sim U(0, 1)$, for $i = 1, \ldots, n_s$.

In other words, Equation (26) holds if, for each particle *i*, either

$$(X_i(t) = Y_i(t) = \hat{Y}_i(t) = \emptyset) \wedge (\lfloor\beta_{i,A}\rfloor = \mathbb{1}_{i,A} = 0) \tag{28}$$

or

$$(X_i(t) = Y_i(t) = \hat{Y}_i(t) = \mathcal{U}) \wedge (\lfloor\beta_{i,S}\rfloor = \mathbb{1}_{i,S} = 0) \tag{29}$$

holds.

3.3.2 Suppose Equation (27) holds

It is therefore implied that the second operand of the min-function always resolves to 0 for each particle, because $|S| + |A|$ cannot ever equal 0 when $N_{\mathcal{U}} > 0$. Alternatively expressed, Equation (27) holds if for each particle *i*,

$$(X_i(t) = Y_i(t) = \hat{Y}_i(t)) \wedge (\hat{Y}_i(t) \notin \{\emptyset, \mathcal{U}\}) \wedge (\lfloor\beta_{i,S}\rfloor + \mathbb{1}_{i,S} + \lfloor\beta_{i,A}\rfloor + \mathbb{1}_{i,A} = 0). \tag{30}$$

Let *c*, $\lfloor\beta\rfloor$ and $\mathbb{1}_{\{r<\beta-\lfloor\beta\rfloor\}}$ denote the respective acceleration coefficient, beta variable, and indicator function that corresponds to an arbitrary choice between either the addition or removal operator. In other words, this notation is equally relevant to both the addition and removal operators. To determine whether the result in Equation (30) is generally expected, $\mathbb{E}(\lfloor\beta\rfloor)$ and $\mathbb{E}(\mathbb{1}_{\{r<\beta-\lfloor\beta\rfloor\}})$ are derived, each in three distinct cases in the following subsections.

3.3.3 Derivation of $\mathbb{E}(\lfloor\beta\rfloor)$

Case ($c = 0$)

$$\mathbb{E}(\lfloor\beta\rfloor) = \mathbb{E}(\lfloor r \cdot 0\rfloor) = \mathbb{E}(\lfloor 0\rfloor) = 0,$$

where $r \sim U(0, 1)$.

Case ($c \in \mathbb{Z}_{>0}$)

Let $W = \lfloor rc\rfloor$. The variable W can only take values in the set

$$\{0, \ldots, c-1\}.$$

If $b \in \{0, \ldots, c-1\}$, then

$$\begin{aligned}(W = b) &\implies b \le c \cdot r < b + 1 \\ &\implies \frac{b}{c} \le r < \frac{b+1}{c}.\end{aligned}$$

Therefore,

$$P(W = b) = \frac{b+1}{c} - \frac{b}{c} = \frac{1}{c}.$$

The expected value of $\lfloor\beta\rfloor$ is therefore derived as follows:

$$\begin{aligned}\mathbb{E}(\lfloor\beta\rfloor) &= \sum_{b=0}^{c-1} \left(\frac{1}{c}\right) b \\ &= \frac{1}{c}\sum_{b=0}^{c-1} b.\end{aligned}$$

It follows from the formula for the sum of the first $(c-1)$ non-negative integers that

$$\begin{aligned}\sum_{b=0}^{(c-1)} b &= \frac{(c-1)(c-1+1)}{2} \\ &= \frac{(c-1)c}{2}.\end{aligned}$$

In the case where $(c = 0)$, the simplified form of the expected value of $\mathbb{E}(\lfloor\beta\rfloor)$ is expressed as

$$\begin{aligned}\mathbb{E}(\lfloor\beta\rfloor) &= \frac{1}{c}\cdot\frac{(c-1)c}{2}\\ &= \frac{(c-1)c}{2c}\\ &= \frac{(c-1)}{2}.\end{aligned}$$

Case ($c \in \mathbb{R}_{>0} \setminus \mathbb{Z}$)

Let $W = \lfloor rc \rfloor$. The variable W can only take values in the set $\{0, \ldots, \lfloor c \rfloor\}$. If $b \in \{0, \ldots, \lfloor c-1 \rfloor\}$, then

$$\begin{aligned}(W = b) &\implies b \le c\cdot r < b+1\\ &\implies \frac{b}{c} \le r < \frac{b+1}{c},\end{aligned}$$

and

$$P(W = b) = \frac{1}{c}.$$

If $b = \lfloor c \rfloor$, then

$$\begin{aligned}(W = b) &\implies b < c\cdot r < c\\ &\implies \frac{b}{c} < r < 1,\end{aligned}$$

and therefore

$$\begin{aligned}P(W = b) &= 1 - \frac{b}{c}\\ &= 1 - \frac{\lfloor c \rfloor}{c}.\end{aligned}$$

The expected value of $\lfloor\beta\rfloor$ is therefore derived as follows:

$$\mathbb{E}(\lfloor\beta\rfloor) = \left(\sum_{b=0}^{\lfloor c-1 \rfloor} \frac{1}{c} b\right) + \lfloor c \rfloor \left(1 - \frac{\lfloor c \rfloor}{c}\right). \tag{31}$$

It follows from the formula for the sum of the first $\lfloor c-1 \rfloor$ non-negative integers that

$$\sum_{b=0}^{\lfloor c-1 \rfloor} b = \frac{\lfloor c-1 \rfloor \lfloor c \rfloor}{2}, \tag{32}$$

and therefore the result in Equation (31) can simplified algebraically as follows:

$$
\begin{aligned}
\mathbb{E}(\lfloor\beta\rfloor) &= \frac{1}{c}\cdot\frac{\lfloor c-1\rfloor\lfloor c\rfloor}{2} + \lfloor c\rfloor\left(1-\frac{\lfloor c\rfloor}{c}\right)\\
&= \frac{\lfloor c-1\rfloor\lfloor c\rfloor}{2c} + \lfloor c\rfloor - \frac{\lfloor c\rfloor^2}{c}\\
&= \frac{\lfloor c-1\rfloor\lfloor c\rfloor}{2c} + (\frac{2c}{2c})\lfloor c\rfloor - \frac{2\lfloor c\rfloor^2}{2c}\\
&= \frac{\lfloor c-1\rfloor\lfloor c\rfloor + 2\lfloor c\rfloor c - 2\lfloor c\rfloor^2}{2c}\\
&= \frac{(\lfloor c\rfloor-1)\lfloor c\rfloor + 2\lfloor c\rfloor c - 2\lfloor c\rfloor^2}{2c}\\
&= \frac{2\lfloor c\rfloor c - \lfloor c\rfloor^2 - \lfloor c\rfloor}{2c}.
\end{aligned}
$$

3.3.4 Derivation of $\mathbb{E}(\mathbb{1}_{\{r<\beta-\lfloor\beta\rfloor\}})$

The expected value of $\mathbb{1}_{\{r<\beta-\lfloor\beta\rfloor\}}$ is derived as follows:

$$
\begin{aligned}
\mathbb{E}(\mathbb{1}_{\{r<\beta-\lfloor\beta\rfloor\}}) &= \mathbb{E}(\beta-\lfloor\beta\rfloor)\\
&= \mathbb{E}(\beta) - \mathbb{E}(\lfloor\beta\rfloor)\\
&= \mathbb{E}(rc) - \mathbb{E}(\lfloor\beta\rfloor), \quad r\sim U(0,1)\\
&= c\mathbb{E}(r) - \mathbb{E}(\lfloor\beta\rfloor), \quad r\sim U(0,1)\\
&= \frac{c}{2} - \mathbb{E}(\lfloor\beta\rfloor),
\end{aligned}
$$

where $\mathbb{E}(\lfloor\beta\rfloor)$ is summarised as

$$
\mathbb{E}(\lfloor\beta\rfloor) = \begin{cases} 0, & \text{if } c=0,\\ \frac{(c-1)}{2}, & \text{if } c\in\mathbb{Z}_{>0},\\ \frac{2\lfloor c\rfloor c-\lfloor c\rfloor^2-\lfloor c\rfloor}{2c}, & \text{if } c\in\mathbb{R}_{>0}\setminus\mathbb{Z}. \end{cases}
$$

The expected value of $\mathbb{1}_{\{r<\beta-\lfloor\beta\rfloor\}}$ can therefore be simplified algebraically as follows:

$$
\begin{aligned}
\mathbb{E}(\mathbb{1}_{\{r<\beta-\lfloor\beta\rfloor\}}) &= \begin{cases} \frac{0}{2}-0, & \text{if } c=0,\\ \frac{c}{2}-\frac{(c-1)}{2}, & \text{if } c\in\mathbb{Z}_{>0},\\ \frac{c}{2}-\frac{2\lfloor c\rfloor c-\lfloor c\rfloor^2-\lfloor c\rfloor}{2c}, & \text{if } c\in\mathbb{R}_{>0}\setminus\mathbb{Z}, \end{cases}\\
&= \begin{cases} 0, & \text{if } c=0,\\ \frac{1}{2}, & \text{if } c\in\mathbb{Z}_{>0},\\ \frac{c^2-2\lfloor c\rfloor c+\lfloor c\rfloor^2+\lfloor c\rfloor}{2c}, & \text{if } c\in\mathbb{R}_{>0}\setminus\mathbb{Z}. \end{cases}
\end{aligned}
$$

3.4 Motivation of non-convergence

It is motivated in Section 3.3 that the absolute convergence of SBPSO necessarily implies that either of Equation (26) or Equation (27) holds for each particle. Furthermore, Equation (26) will only hold if either Equation (28) or Equation (29) holds, and Equation (27) will only hold if Equation (30) holds.

To explain why absolute convergence is not typically expected, the following sub-condition is considered from Equation (30):

$$\lfloor \beta_{i,S} \rfloor + \mathbb{1}_{i,S} + \lfloor \beta_{i,A} \rfloor + \mathbb{1}_{i,A} = 0,$$

for $i = 1, \ldots, n_s$.

Similarly, the respective sub-conditions from Equation (28) and Equation (29) are provided as

$$\beta_{i,A} = \mathbb{1}_{i,A} = 0$$

and

$$\beta_{i,S} = \mathbb{1}_{i,S} = 0,$$

for $i = 1, \ldots, n_s$.

In each of the stated sub-conditions, it is required for an indicator function ($\nVdash$) to resolve to 0 in order for the broader condition to hold. Therefore, to motivate that absolute convergence is not typically expected, it is sufficient to show that $\mathbb{E}(\mathbb{1}_{\{r<\beta-\lfloor\beta\rfloor\}}) \neq 0$, except in the trivial case where $c = 0$.

In the equation,

$$\mathbb{E}(\nVdash_{\{r<\beta-\lfloor\beta\rfloor\}}) = \begin{cases} 0, & \text{if } c = 0, \\ \frac{1}{2}, & \text{if } c \in \mathbb{Z}_{>0}, \\ \frac{c^2 - 2\lfloor c \rfloor c + \lfloor c \rfloor^2 + \lfloor c \rfloor}{2c}, & \text{if } c \in \mathbb{R}_{>0} \setminus \mathbb{Z}, \end{cases}$$

it is observed that $\mathbb{E}(\nVdash_{\{r<\beta-\lfloor\beta\rfloor\}}) = \frac{1}{2} \neq 0$, when $c \in \mathbb{Z}_{>0}$. The following proof is provided to show that $\mathbb{E}(\nVdash_{\{r<\beta-\lfloor\beta\rfloor\}}) \neq 0$ in the case where $c \in \mathbb{R}_{>0} \setminus \mathbb{Z}$:

Suppose that $c > 0$ and

$$\frac{c^2 - 2\lfloor c \rfloor c + \lfloor c \rfloor^2 + \lfloor c \rfloor}{2c} = 0.$$

It then follows that

$$\begin{aligned} & c^2 - 2\lfloor c \rfloor c + \lfloor c \rfloor^2 + \lfloor c \rfloor = 0 \\ \Rightarrow \quad & c^2 - 2\lfloor c \rfloor c = -\lfloor c \rfloor^2 - \lfloor c \rfloor \\ \Rightarrow \quad & c^2 - 2\lfloor c \rfloor c + \lfloor c \rfloor^2 = -\lfloor c \rfloor \\ \Rightarrow \quad & (c - \lfloor c \rfloor)^2 = -\lfloor c \rfloor, \end{aligned}$$

which clearly does not hold, because the left-hand side will always resolve to a positive non-integer value, but the right-hand side can only ever be a negative integer.

4 Toy experiments

Two toy experiments are proposed to test ideas related to SBPSO stagnation and convergence.

Toy experiment 1 evaluates the effects of different fixed values for c_3 and c_4, as well as three distinct time-varying scheduling strategies for these parameters on convergence. Swarm diversity is quantified using the average Jaccard distance (or Jaccard diversity) between particle positions, as proposed by Erwin and Engelbrecht (2020). The Jaccard diversity is calculated as

$$\bar{d}_J = \frac{\sum_{i=1}^{n_s-1} \sum_{j=i+1}^{n_s} d_J(S_i, S_j)}{\sum_{i=1}^{n_s-1} \sum_{j=i+1}^{n_s} 1}, \tag{33}$$

where $d_J(\cdot)$ is the Jaccard distance, given by

$$d_J(S_i, S_j) = 1 - \frac{|S_i \cap S_j|}{|S_i \cup S_j|}, \tag{34}$$

and S_i and S_j are sets.

Toy experiment 2 is designed to test the relationship between the swarm size (n_s) and aspects of stagnation, including the frequency and duration of observed stagnation periods.

4.1 Experimental design

The SBPSO algorithm was evaluated on the MKP benchmark suite from the operations research library compiled by Beasley (1990); in total 55 test problems were used, each with a known optimum.

For the MKP, a value function is used to determine how suitable a solution is. The value function is the sum of the worth, or profit, of each item added to the knapsack, *i.e.*

$$V(X) = \sum_{j=1}^{n} p_j x_j. \tag{35}$$

The items which can be added to the knapsack are subject to the constraints

$$\sum_{j=1}^{n} w_{ij} x_j \leq C_i \tag{36}$$

where x_j indicates if the j^{th} variable has been selected, p_j is the value of the j^{th} variable, w_{ij} is the weight of variable j for constraint i, and C_i is the maximum value of constraint i.

Since each test problem has a known optimum, the fitness function is defined to be bounded on $[0, 1]$ with the optimum at 0. The fitness function is

$$f(X) = \begin{cases} 1, & \text{if } \sum_{j=1}^{n} w_{ij} x_j > C_i, \\ 1 - \frac{V(X)}{Y_{\text{opt}}}, & \text{otherwise}, \end{cases} \tag{37}$$

where Y_{opt} is the known optimum for the problem.

4.2 Toy experiment 1

This toy experiment provides empirical evidence that the acceleration coefficients c_3 and c_4 play an integral role in preventing swarm convergence.

4.2.1 Methodology

The selection of fixed values for c_3 and c_4 depends on the given optimisation problem, as these values are expressed as a percentage of $N_{\mathcal{U}}$. Four variants are evaluated, where c_3 and c_4 are set to 25%, 50%, 75%, and 100% of $N_{\mathcal{U}}$, respectively.

Three scheduling approaches for c_3 and c_4 are also compared, including linear scheduling, and scheduling based on concave-up and concave-down functions defined in terms of the natural growth rate.

The experiment consists of independent trials per each of the 55 MKP benchmarks, where each independent trial entails a selection from either one of the four constant values for c_3 and c_4, or one of the three aforementioned scheduling schemes, as well as a stationary parameter configuration for c_1 and c_2. It is clarified that these stationary parameter configurations are acquired by sampling 64 distinct two-dimensional SSs in the range (0, 1). In each independent trial, the diversity results are recorded over 2000 search iterations, and a fixed swarm size of $n_s = 30$ is used.

Figure 1 represents the three scheduling schemes, which define c_3 and c_4 with respect to a decreasing upper bound on the number of additions and removals of MKP items as a proportion of $N_{\mathcal{U}}$. The degree of swarm convergence is measured in terms of the average Jaccard distance between particle positions.

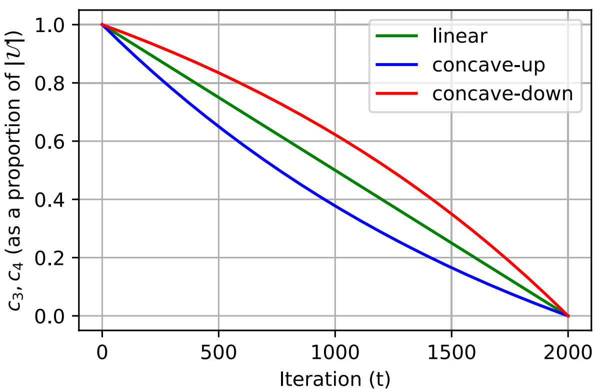

Fig. 1 Scheduling Functions for c_3 and c_4

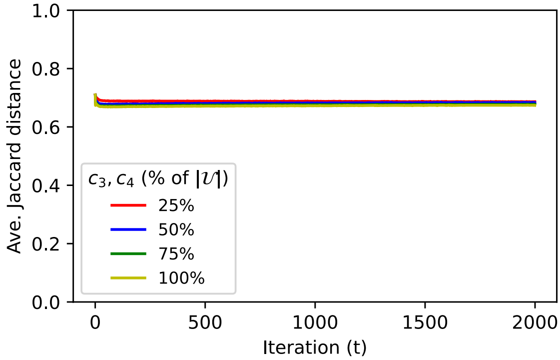

Fig. 2 Average diversity for constant values of c_3 and c_4

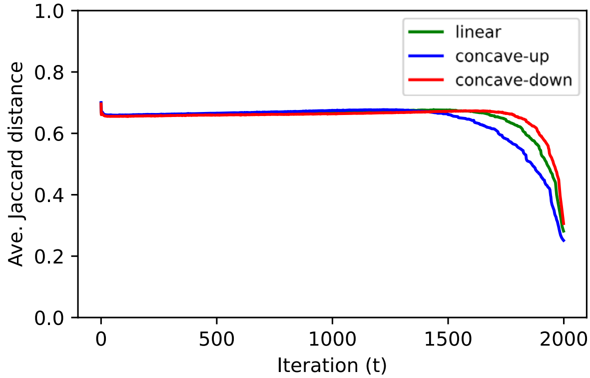

Fig. 3 Average diversity of scheduling approaches

4.2.2 Results

Figure 2 shows the diversity graphs for each of the four SBPSO variants that use constant values of c_3 and c_4, averaged over the different parameter configurations and benchmark problems.

The four graphs in Figure 2 are similar in that they reflect almost entirely constant average diversity for most of the search process. Moreover, the four graphs are observed to occupy values in a similar range. For each variant, a small dip in the average diversity is observed within the first few iterations of the search process. This trend is motivated by the fact that random uniform initialization of particles often results in a high number of disjoint particle positions, and thereafter the diversity is expected to decrease as particles are attracted towards the global best solution and subsequently share an increasing number of common elements.

Figure 3 indicates the diversity graphs of the SBPSO variants for each of the three scheduling schemes for c_3 and c_4.

Neither of the graphs in Figure 3 observe trends that match the descriptions of *phase 1* and *phase 2* provided in Section 2.1. Furthermore, all three graphs indicate a prolonged period of exploration throughout most of the search, preceded by a slight initial drop in diversity during the first few iterations.

Figure 3 shows that, on average, the SBPSO variants with a concave-up scheduling function for c_3 and c_4 begin to approach convergence earlier than those with a linear scheduling function. Similarly, variants with a linear scheduling function approach convergence earlier than those with a concave-down scheduling function.

The results of this experiment allude to the importance of c_3 and c_4 in facilitating convergence. Although absolute convergence is not observed on average for any of the evaluated variants, it is clear that the scheduled approaches for c_3 and c_4 are more effective in instigating swarm exploitation than the variants that use fixed values of c_3 and c_4. Furthermore, the evaluated SBPSO variants that use constant value of c_3 and c_4 are observed to promote a state of perpetual swarm exploration, on average.

4.3 Toy experiment 2

This toy experiment provides empirical evidence that, even if provided with the ideal conditions for convergence, the SBPSO swarm does not converge.

4.3.1 Methodology

As motivated in Section 3.2, the likelihood of stagnation decreases as the swarm size, n_s, assumes a sufficiently large value. The following experiment aims to test this claim empirically by measuring the relationship between the frequency and duration of stagnation periods for different values of n_s.

Let Y_{opt} denote the true optima of an optimisation problem. Three different problem scenarios are considered, including the cases where $Y_{\text{opt}} = \emptyset$, $Y_{\text{opt}} = \mathcal{U}$, and in the typical case where $(Y_{\text{opt}} \subset \mathcal{U}) \wedge (Y_{\text{opt}} \notin \{\emptyset, \mathcal{U}\})$. Three unconstrained minimization problems f_1 , f_2 and f_3 are defined with respect to the objective function,

$$f(X) = \begin{cases} \sum_{x_i \in X} x_i, & \text{if } X \neq \emptyset, \\ 0, & \text{if } X = \emptyset, \end{cases}$$

and the information in Table 1, so that each problem corresponds to one of the three problem scenarios.

There are only four possible solutions for each problem case, namely the solutions $\mathcal{U}$, $\emptyset$, and the two sets that contain either one of the individual elements in $\mathcal{U}$ (e.g. $\{1\}$ and $\{2\}$ in problem f_1). This decision is motivated by the goal to keep $\mathcal{U}$ as small as possible while still allowing for the three types of solutions X_1, X_2, and X_3 where $X_1 = \emptyset$, $X_2 = \mathcal{U}$, and $(X_3 \subset \mathcal{U}) \wedge (X_3 \notin \{\emptyset, \mathcal{U}\})$. Furthermore, the definition of $\mathcal{U}$ restricts the particle positions to solutions within the smallest set of possible solutions, subject to the specification that the solution cases, X_1, X_2 and X_3, are supported.

Table 1 Toy experiment 2 benchmark functions

	Y_{opt}	$\mathcal{U}$
f_1	$\emptyset$	$\{1, 2\}$
f_2	$\{-1\}$	$\{1, -1\}$
f_3	$\{-1, -2\}$	$\{-1, -2\}$

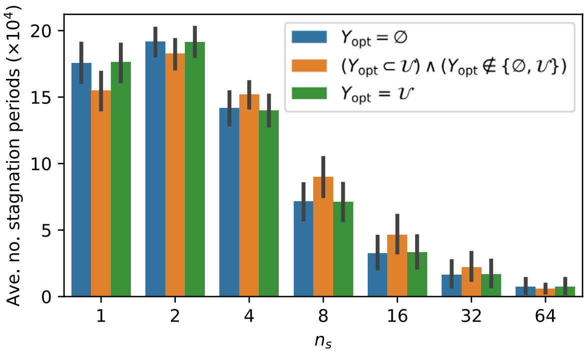

Fig. 4 Number of stagnation periods per swarm size

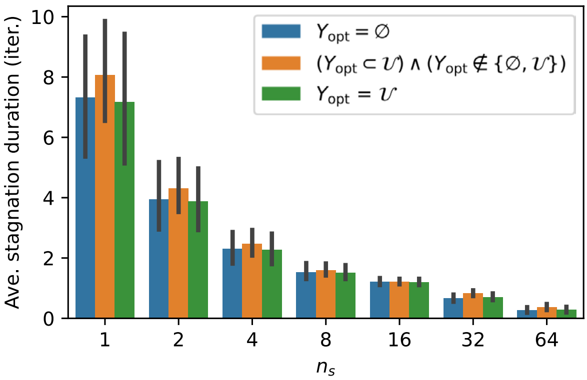

Fig. 5 Average number of iterations stagnated per swarm size

The acceleration coefficients c_1, c_2, c_3, and c_4 are sampled as 96 distinct four-dimensional SSs, where c_1 and c_2 are sampled within the range $(\frac{1}{\mathcal{U}}, 1)$, to prevent particle immobilization, and c_3 and c_4 are sampled within the range (0, 1). The values of c_3 and c_4, being less than 1, ensure that the probability of stagnation is entirely influenced by the resolved values of the indicator functions associated with the element addition and removal operators in the position update equation for each particle.

For each independent run, the neighbourhood best $(\hat{Y}_i)$, personal best (Y_i) and position (X_i) of each particle $i = 1, \ldots, n_s$, are set to the true optimum (Y_{opt}) prior to the start of the search process.

The frequency and duration of stagnation periods are captured over 10^6 iterations, where a stagnation period constitutes any consecutive sequence of iterations where no changes in particle positions are observed. Seven SBPSO variants are considered, each corresponding to a different value of n_s from the set $\{1, 2, 4, 8, 16, 32, 64\}$.

4.3.2 Results

Figure 4 and Figure 5 respectively indicate the average number of stagnation periods and average duration of stagnation periods for each of the evaluated SBPSO variants. Additionally, Figure 6 is provided to indicate the average percentage of the search process where the swarm is considered stagnated. It is clarified that each variant corresponds to a group of three bar plots, which are associated with the three types of problem scenarios.

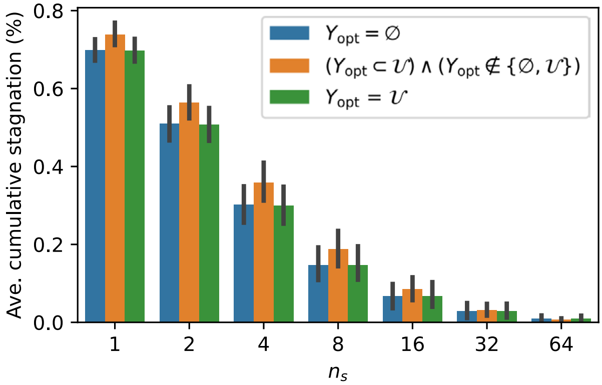

Fig. 6 Percentage of the search stagnated per swarm size

The bar plots in Figure 4 reveal that the average number of stagnation periods were not inherently larger for the variants with smaller values of n_s. Furthermore, it is observed that the variant with only a single-particle swarm experienced fewer stagnation periods on average than the variant with $n_s = 2$. Figure 5 indicates a clear trend in which algorithm variants with smaller values of n_s experience longer periods of stagnation than those with larger values. This result indicates that, although the variant with $n_s = 1$ stagnated less frequently than the variant with $n_s = 2$, the stagnation periods were longer, on average. Figure 6 reveals a distinct trend for all problem cases, highlighting an inverse relationship between the swarm size and the cumulative number of stagnation iterations, as a percentage of the 10^6 allocated search iterations.

5 Acceleration coefficient sensitivity analysis

To gain a better understanding of how the acceleration coefficients affect the convergence behaviour of the swarm, a SA was performed. The SA was performed on the same MKP problems, using the same objective function, as outlined in Section 4.1. The sensitivity of four different metrics, relevant to swarm convergence, were evaluated against each of the four acceleration coefficients, *i.e.* c_1, c_2, c_3, and c_4. The metrics were selected in order gain insight into how the behaviour of the swarm changes with different values for each coefficient.

5.1 Sensitivity analysis empirical procedure

For each of the 55 benchmark problems, 192 unique SSs were generated, with each SS consisting of a vector of control parameter configurations ($[c_1, c_2, c_3, c_4]$). The resultant 192 four-dimensional were used to perform ANOVA, from which the Sobol' indices were calculated as outline in Section 2.4. The metrics selected for analysis were Jaccard diversity (Q_J), average velocity size (Q_V), diversity rate of change (DRoC) (Q_D), and diversity trend (Q_T). The following subsections define the four metrics used for SA.

5.1.1 Average diversity

The diversity of the swarm *S*, for which SA was performed, is defined as the average Jaccard distance between all particle positions at the final iteration (n_t) of the simulation. Specifically, this corresponds to Equation (33) evaluated at n_t, as given by

$$Q_J(S) = \frac{\sum_{i=1}^{n_s-1} \sum_{j=i+1}^{n_s} d_J(X_i(n_t), X_j(n_t))}{\sum_{i=1}^{n_s-1} \sum_{j=i+1}^{n_s} 1}. \tag{38}$$

5.1.2 Average velocity size

The average size of the velocity sets in the swarm was used to determine how well the swarm had converged by the end of the search. A swarm where all particles have small velocity sets is more likely to have converged than a swarm which has large velocity sets (*i.e.* is still actively exploring). The average velocity set size is simply calculated as

$$Q_V(S) = \frac{1}{n_s} \sum_{i=1}^{n_s} |V_i(n_t)|. \tag{39}$$

5.1.3 Diversity rate of change

The sensitivity of the DRoC metric was used to quantify how each coefficient affects the tendency of the swarm to converge. The DRoC is a well-established metric used to quantify the convergence profile of a PSO swarm into a single value (Bosman et al., 2014).

The DRoC fits a two-piece piecewise linear function to the swarm diversity plotted over time. The piecewise approximation takes the form $y(x) \approx f(x)$ for $i_0 \leq x \leq i_2$ with i_0 and i_2 the start and end iteration numbers of the simulation. Each section of the linear approximation forms part of the function approximation as

$$y(x) = \begin{cases} a_1 + b_1 x \text{ for } i_0 \leq x \leq i_1 \\ a_2 + b_2 x \text{ for } i_1 < x \leq i_2, \end{cases} \tag{40}$$

with $i_0 < i_i < i_2$, $i_0 = 1$, and $i_2 = n_t$. The coefficient b_1 quantifies how quickly the swarm diversity decreases during the initial phase of exploration, hence is used as the DRoC metric. The time scale of the swarm diversity plot is normalised to the range [0, 1] in order to match the range of the average Jaccard diversity of the swarm. As a result, the SA is performed with respect to

$$Q_D(S) = b_1. \tag{41}$$

5.1.4 Diversity trend

The DRoC metric was developed for the analysis of real-valued PSO and has not been established as an appropriate metric for combinatorial domains. Hence, the average trend

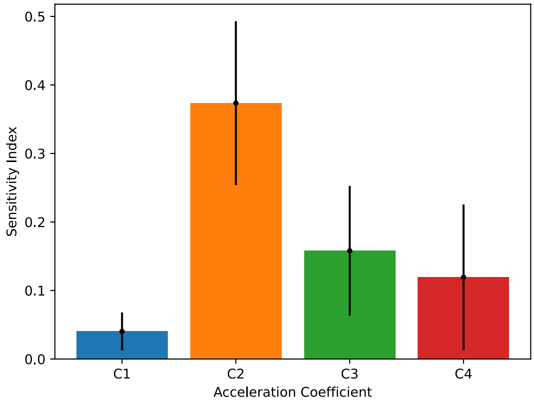

Fig. 7 Sensitivity of swarm diversity

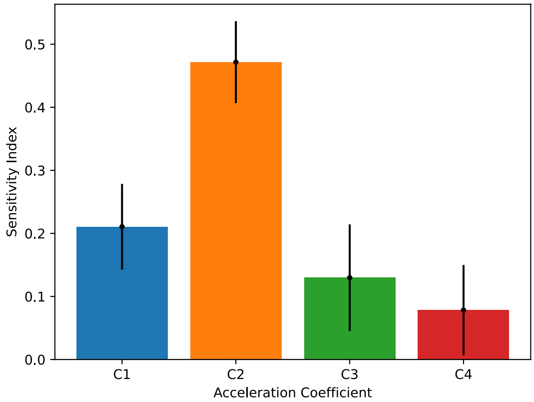

Fig. 8 Sensitivity of swarm velocity

of the swarm diversity (as determined by a one-piece linear approximation) was also used. The one-piece linear approximation of the swarm diversity is used to capture the overall tendency of the diversity, instead of only in phase 2 as given by Bosman et al. (2014). This additional metric is used to gain additional insight which may be missed by Q_D, since DRoC is not established for combinatorial domains. Given that the linear approximation takes the form $y(x) = a_3 + b_3 x$ for $i_0 \leq x \leq i_2$, with $i_0 = 1$ and $i_2 = n_t$, the SA is performed with respect to

$$Q_T(S) = b_3. \tag{42}$$

5.2 Sensitivity analysis results

The SA results show that there is not one control parameter that influences all metrics of swarm convergence. Figure 7 shows that the average diversity of the swarm at the final iteration is most sensitive to c_2, *i.e.* the attraction to the global best.

The average cardinality of the velocity set sizes are most sensitive to c_2, as shown in Figure 8. This figure also shows that the velocity set sizes are not sensitive to c_1.

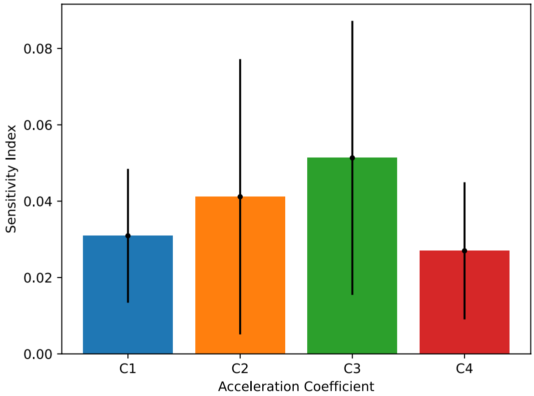

Fig. 9 Sensitivity of swarm DRoC

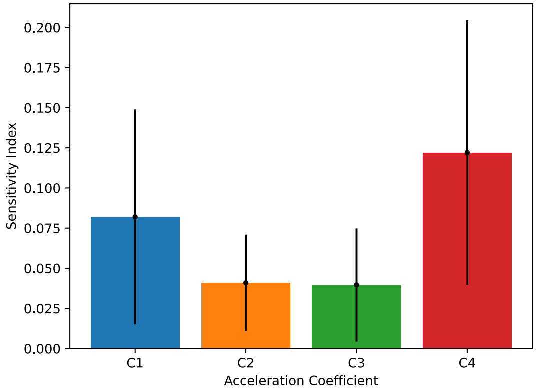

Fig. 10 Sensitivity of swarm diversity trend

For DRoC, as shown in Figure 9, there is a slightly higher than average sensitivity to c_3. However, the sensitivity indices for all acceleration coefficients are less than 0.06.

The average trend of the swarm diversity shows hardly any sensitivity to any of the acceleration coefficients. This indicates that the rate of change of velocity does not change significantly with different control parameter combinations.

The SA results can be broken down into two categories: (1) results that summarise the state of the swarm at the end of the simulation, and (2) results that shed light on what happens during the simulation.

Figures 7 and 8 reveal what happened at the end of the simulation (*i.e.* at iteration n_t); where a low swarm diversity and an average velocity set size of zero both imply possible convergence. Clearly, both these metrics are most sensitive to c_2.

Figures 9 and 10 shed light on what happens during the simulation; the figures show minimal sensitivity to any of the acceleration coefficients, nor any significant differences between the coefficients.

As a further quantitative demonstration of the results in Figures 7 to 10, Table 2 presents the average values of the sensitivity indices for each of the control parameters.

The results of the SA must be interpreted in light of the mathematical proof presented in Section 3.3. There is a low level of sensitivity of all metrics to the c_3 and c_4 control

Table 2 Sensitivity analysis of acceleration coefficients

Quantity	c_1	c_2	c_3	c_4
Q_J	0.0433 ± 0.0380	0.3811 ± 0.1274	0.1659 ± 0.0955	0.1166 ± 0.0979
Q_V	0.2126 ± 0.0623	0.4516 ± 0.0687	0.0488 ± 0.0686	0.0827 ± 0.0662
Q_D	0.0610 ± 0.0454	0.0431 ± 0.0361	0.0389 ± 0.0341	0.1128 ± 0.0670
Q_T	0.0493 ± 0.0340	0.0981 ± 0.0747	0.2742 ± 0.1253	0.0733 ± 0.0888

parameters; this is believed to be because any non-trivial value (*i.e.* $c_3 \neq 0$, $c_4 \neq 0$) results in excessive exploration due to non-convergence of the swarm.

6 Conclusion

This paper studied the convergence behaviour of a variant of particle swarm optimisation (PSO), the set-based particle swarm optimisation (SBPSO) algorithm. It was shown that the SBPSO does not converge, even under ideal conditions. The study was conduced through a combination of thorough mathematical proofs, empirical experiments, and sensitivity analysis (SA). The mathematical proofs proved that the expected cardinality of the velocity sets is never zero, since the addition and removal operators always inject diversity into the swarm. The empirical experiments involved specially crafted toy experiments which recreated ideal convergence conditions for SBPSO, and showed that even then convergence does not take place. The SA revealed to which acceleration coefficients SBPSO is most sensitive given different convergencerelated evaluated metrics, and showed the effect of each coefficient on the diversity and velocity sizes of the swarms. The SBPSO remains a promising PSO-variant for combinatorial domains and discrete optimisation problems (DOPs); therefore this paper is not meant to discourage future development on the algorithm. Instead, the authors recommend that additional effort be expended to rectify the current convergence shortcomings of the algorithm such that DOPs can be solved with greater certainty and repeatability. Future work can include the development of alternative exploration mechanisms which can replace the current addition and removal operators; these new exploration mechanism should allow the swarm to converge without hindering effective exploration of as much of the search space as possible. This paper provides the first in-depth critique of SBPSO, yet the identified shortcomings of the algorithm remain unaddressed. Since this study was limited to analysing the behaviour of SBPSO, the primary unresolved challenge is the development of an improved exploration mechanism that can facilitate swarm convergence. Future analysis of swarm behaviour in SBPSO could incorporate additional diversity metrics, such as the average symmetric difference or the equivalent Hamming distance among particle positions. Moreover, previous studies have shown that parameter configurations in continuous PSO that guarantee convergence are generally associated with improved performance and robustness. It remains necessary, however, to determine whether these convergence-related benefits also extend to discrete PSO variants. Further, the convergence and performance of other PSO variants for DOPs, *e.g.* the algebraic particle swarm optimisation (APSO) and geometric particle swarm optimisation (GPSO), can be researched and compared to that of SBPSO. The APSO algorithm is conveniently abstract, and applicable to any problem defined on a group, *e.g.* $\mathbb{B}^n$ with the exclusive-or ($\underline{\vee}$). The previous successful application of APSO to the NK landscape problem bodes well for the application to more combinatorial problems. In addition, the application of GPSO on the problem of sudoku lends itself to the possibility of a more general framework for combinatorial problem representation. An interesting contribution to the literature would be to compare and unify the three distinct but interrelated approaches of SBPSO, APSO, and GPSO.

Author contributions Conceptualization of the work is attributed to A.P.E. The mathematical convergence proof was contributed by G.J.S. G.J.S and J.-P.v.Z. contributed equally to the remaining sections of the manuscript text. All authors reviewed the manuscript.

Funding Open access funding provided by Stellenbosch University.

Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

Baioletti, M., Milani, A., & Santucci, V. (2017). Algebraic particle swarm optimization for the permutations search space. In *Proceedings of the IEEE congress on evolutionary computation*, IEEE, pp 1587–1594, https://doi.org/10.1109/cec.2017.7969492

Beasley, J. E. (1990). Or-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society, 41*(11), 1069–1072. https://doi.org/10.1057/jors.1990.166

Blackwell, T., & Branke, J. (2004). Multi-swarm optimization in dynamic environments. In *Proceedings of the workshops on applications of evolutionary computation* (pp 489–500). Springer.

Borgonovo, E. (2007). A new uncertainty importance measure. *Reliability Engineering & System Safety, 92*(6), 771–784. https://doi.org/10.1016/j.ress.2006.04.015

Borgonovo, E., & Plischke, E. (2016). Sensitivity analysis: A review of recent advances. *European Journal of Operational Research, 248*(3), 869–887. https://doi.org/10.1016/j.ejor.2015.06.032

Bosman, P., & Engelbrecht, A.P. (2014). Diversity rate of change measurement for particle swarm optimisers. In M. Dorigo, M. Birattari, S. Garnier et al (eds) *Proceedings of the international conference on swarm intelligence*, Lecture Notes in Computer Science, (vol 8667. pp 86–97), Springer.https://doi.org/10.1007/978-3-319-09952-1_8

Brown, L., & Engelbrecht, A. (2022). Set-based particle swarm optimization for data clustering. In *Proceedings of the 6th international conference on intelligent systems, metaheuristics & swarm intelligence*, pp 43–49, https://doi.org/10.1145/3533050.3533057

Cleghorn, C. W., & Engelbrecht, A. P. (2014). A generalized theoretical deterministic particle swarm model. *Swarm Intelligence, 8*(1), 35–59. https://doi.org/10.1007/s11721-013-0090-y

Cleghorn, C. W., & Engelbrecht, A. P. (2015). Particle swarm variants: Standardized convergence analysis. *Swarm Intelligence, 9*(2), 177–203. https://doi.org/10.1007/s11721-015-0109-7

Cleghorn, C. W., & Engelbrecht, A. P. (2018). Particle swarm stability: A theoretical extension using the non-stagnate distribution assumption. *Swarm Intelligence, 12*(1), 1–22. https://doi.org/10.1007/s11721-017-0141-x

Cleghorn, C.W., & Engelbrecht, A.P. (2014b). Particle swarm convergence: An empirical investigation. In *Proceedings of the IEEE congress on evolutionary computation*, IEEE, pp 2524–2530, https://doi.org/10.1109/CEC.2014.6900439

Cleghorn, C.W., & Engelbrecht. A. (2016). Particle swarm optimizer: The impact of unstable particles on performance. In *Proceedings of the IEEE symposium series on computational intelligence*, IEEE, pp 1–7

Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58–73. https://doi.org/10.1109/4235.985692

De Wet, R.M., & Engelbrecht, A. (2023). Set-based particle swarm optimization for data clustering: Comparison and analysis of control parameters. In *Proceedings of the 7th international conference on intelligent systems, metaheuristics & swarm intelligence*, pp 103–110, https://doi.org/10.1145/3596947.3596956

Engelbrecht, A. P., Grobler, J., & Langeveld, J. (2019). Set based particle swarm optimization for the feature selection problem. *Engineering Applications of Artificial Intelligence, 85*, 324–336. https://doi.org/10.1016/j.engappai.2019.06.008

Engelbrecht, A.P. (2013). Roaming behavior of unconstrained particles. In *Proceedings of the BRICS Congress on computational intelligence and 11th Brazilian congress on computational intelligence*, (pp 104–111). IEEE.

Erwin, K., & Engelbrecht, A. (2020a). Diversity measures for set-based meta-heuristics. In *Proceedings of the 7th international conference on soft computing & machine intelligence* (pp 45–50). IEEE.

Erwin, K., & Engelbrecht, A.P. (2020b). Set-based particle swarm optimization for portfolio optimization. In M. Dorigo, T. Stützle, M. J. Blesa et al (eds) *Proceedings of the 12th international conference on swarm intelligence*, (pp 333–339). Springer, https://doi.org/10.1007/978-3-030-60376-2_28

Erwin, K., & Engelbrecht, A. (2021). Set-based particle swarm optimization for portfolio optimization with adaptive coordinate descent weight optimization. In *Proceedings of the IEEE symposium series on computational intelligence* (pp 01–08). IEEE. https://doi.org/10.1109/SSCI50451.2021.9659541

Erwin, K., & Engelbrecht, A. (2023). Multi-guide set-based particle swarm optimization for multi-objective portfolio optimization. *Algorithms, 16*(2), Article 62. https://doi.org/10.3390/a16020062

Gazi, V. (2012). Stochastic stability analysis of the particle dynamics in the PSO algorithm. In *Proceedings of the IEEE international symposium on intelligent control*, (pp 708–713). IEEE, https://doi.org/10.1109/ISIC.2012.6398264

Harrison, K. R., Engelbrecht, A. P., & Ombuki-Berman, B. M. (2018). Self-adaptive particle swarm optimization: A review and analysis of convergence. *Swarm Intelligence, 12*(3), 187–226. https://doi.org/10.1007/s11721-017-0150-9

Harrison, K.R., Ombuki-Berman, B.M., & Engelbrecht, A.P. (2017). Optimal parameter regions for particle swarm optimization algorithms. In *Proceedings of the IEEE congress on evolutionary computation*, (pp 349–356). IEEE, https://doi.org/10.1109/CEC.2017.7969333

Herman, J., & Usher, W. (2017). SALib: An open-source Python library for sensitivity analysis. *The Journal of Open Source Software*. https://doi.org/10.21105/joss.00097

Iwanaga, T., Usher, W., & Herman, J. (2022). Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses. *Socio-Environmental Systems Modelling, 4*, 1–15. https://doi.org/10.18174/sesmo.18155

Jiang, M., Luo, Y. P., & Yang, S. Y. (2007). Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters, 102*(1), 8–16. https://doi.org/10.1016/j.ipl.2006.10.005

Kadirkamanathan, V., Selvarajah, K., & Fleming, P. J. (2006). Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation, 10*(3), 245–255. https://doi.org/10.1109/TEVC.2005.857077

Kennedy, J., & Eberhart, R. C. (2001). *Swarm intelligence*. Morgan Kaufmann.

Langeveld, J., & Engelbrecht, A. P. (2012). Set-based particle swarm optimization applied to the multidimensional knapsack problem. *Swarm Intelligence, 6*(4), 297–342. https://doi.org/10.1007/s11721-012-0073-4

Leonard, B. J., Engelbrecht, A. P., & Cleghorn, C. W. (2015). Critical considerations on angle modulated particle swarm optimisers. *Swarm Intelligence, 9*(4), 291–314. https://doi.org/10.1007/s11721-015-0114-x

Liu, Q. (2015). Order-2 stability analysis of particle swarm optimization. *Evolutionary Computation, 23*(2), 187–216. https://doi.org/10.1162/EVCO_a_00129

Moraglio, A., Di Chio, C., & Poli, R. (2007). Geometric particle swarm optimisation. In *Proceedings of the 10th European conference on genetic programming*, (pp 125–136). Springer, https://doi.org/10.1007/978-3-540-71605-1_12

Ochoa, G., Verel, S., Daolio, F., Tomassini, M. (2014). Local optima networks: A new model of combinatorial fitness landscapes. In *Recent advances in the theory and application of fitness landscapes*, (pp. 233–262). Springer.

Pearson, K. (1905). *On the general theory of skew correlation and non-linear regression*. 14, Dulau and Company

Plischke, E., Borgonovo, E., & Smith, C. L. (2013). Global sensitivity measures from given data. *European Journal of Operational Research, 226*(3), 536–550. https://doi.org/10.1016/j.ejor.2012.11.047

Poli, R. (2009). Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Transactions on Evolutionary Computation, 13*(4), 712–721. https://doi.org/10.1109/TEVC.2008.2011744

Poli, R., & Broomhead, D. (2007). Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation. In *Proceedings of the 9th annual conference on genetic and evolutionary computation*, pp 134–141, https://doi.org/10.1145/1276958.1276977

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S. (2008). *Global Sensitivity Analysis: The Primer*. John Wiley & Sons

Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M. (2004). *Sensitivity analysis in practice: A guide to assessing scientific models*, vol 1. Wiley Online Library

Santucci, V., Baioletti, M., & Milani, A. (2020). An algebraic framework for swarm and evolutionary algorithms in combinatorial optimization. *Swarm and Evolutionary Computation, 55*, Article 100673.

Shi, Y., & Eberhart, R.C. (1998a). A modified particle swarm optimizer. In *Proceedings of the 1998 IEEE international conference on evolutionary computation*, pp 69–73

Shi, Y., & Eberhart, R.C. (1998b). Parameter selection in particle swarm optimization. In *Proceedings of the 7th international conference on evolutionary programming*, (pp 591–600). Springer.

Sobol, I. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling & Computational Experiment, 1*, 407–414.

Trelea, I. C. (2003). The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters, 85*(6), 317–325. https://doi.org/10.1016/S0020-0190(02)00447-7

Van den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences, 176*(8), 937–971. https://doi.org/10.1016/j.ins.2005.02.003

Van Zyl, J. P., & Engelbrecht, A. P. (2023). Set-based particle swarm optimisation: A review. *Mathematics*. https://doi.org/10.3390/math11132980

Van Zyl, J.P., & Engelbrecht, A.P. (2021). Polynomial approximation using set-based particle swarm optimization. In *Proceedings of the 12th international conference on swarm intelligence*, (pp 210–222). Springer, https://doi.org/10.1007/978-3-030-78743-1_19

Van Zyl, J.P., & Engelbrecht, A.P. (2022). Rule induction using set-based particle swarm optimisation. In *Proceedings of the IEEE congress on evolutionary computation*, (pp 1–8). IEEE, https://doi.org/10.1109/CEC55065.2022.9870360

Wagner, H. M. (1995). Global sensitivity analysis. *Operations Research, 43*(6), 948–969. https://doi.org/10.1287/opre.43.6.948

Witt, C. (2009). Why standard particle swarm optimisers elude a theoretical runtime analysis. In *Proceedings of the tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, pp 13–20

Zielinski, K., Laur, R. (2007). Stopping criteria for a constrained single-objective particle swarm optimization algorithm. *Informatica* 31(1)