An Analysis of the Search Mechanisms of the Bees Algorithm

Baronti, Luca; Castellani, Marco; Pham, Duc

DOI:
10.1016/j.swevo.2020.100746

Citation for published version (Harvard):
Baronti, L, Castellani, M & Pham, D 2020, 'An Analysis of the Search Mechanisms of the Bees Algorithm', *Swarm and Evolutionary Computation*, vol. 59, 100746. https://doi.org/10.1016/j.swevo.2020.100746

An Analysis of the Search Mechanisms of the Bees Algorithm

Luca Baronti[a], Marco Castellani[a], Duc Truong Pham[a]

[a] *Department of Mechanical Engineering, University of Birmingham, United Kingdom*

Abstract

The Bees Algorithm has been successfully applied for over a decade to a large number of optimisation problems. However, a mathematical analysis of its search capabilities, the effects of different parameters used, and various design choices has not been carried out. As a consequence, optimisation of the Bees Algorithm has so far relied on trial-and-error experimentation. This paper formalises the Bees Algorithm in a rigorous mathematical description, beyond the qualitative biological metaphor. A review of the literature is presented, highlighting the main variants of the Bees Algorithm, and its analogies and differences compared with other optimisation methods. The local search procedure of the Bees Algorithm is analysed, and the results experimentally checked. The analysis shows that the progress of local search is mainly influenced by the size of the neighbourhood and the stagnation limit in the site abandonment procedure, rather than the number of recruited foragers. In particular, the analysis underlines the trade-off between the step size of local search (a large neighbourhood size favours quick progress) and the likelihood of stagnation (a small neighbourhood size prevents premature site abandonment). For the first time, the implications of the choice of neighbourhood shape on the character of the local search are clarified. The paper reveals that, particularly in high-dimensional spaces, hyperspherical neighbourhoods allow greater search intensification than hypercubic neighbourhoods. The theoretical results obtained in this paper are in good agreement with the findings of several experimental studies. It is hoped that the new mathematical formalism here introduced will foster further understanding and analysis of the Bees Algorithm, and that the theoretical results obtained will provide useful parameterisation guidelines for applied studies.

Keywords: Bees Algorithm, Optimisation, Statistical Analysis

*Marco Castellani

Email addresses: `l.baronti@bham.ac.uk` (Luca Baronti), `m.castellani@bham.ac.uk` (Marco Castellani), `D.T.Pham@bham.ac.uk` (Duc Truong Pham)

1. Introduction

The *Bees Algorithm* [1] is a nature-inspired intelligent technique that has found application in a wide range of complex optimisation problems [2, 3]. The main idea motivating this algorithm is to model the foraging behaviour of honey bees to address the *exploration* vs *exploitation* trade-off. According to that model, agents simulating *scout bees* randomly explore the solution space looking for areas of high fitness. The scouts that found the most promising solutions recruit (through performing a *waggle dance* [4]) other agents (*forager bees*) for local exploitative search. Local search is conducted in parallel at different *sites*, that is, within neighbourhoods centred on the solutions marked by the scouts.

Despite the initial idea being to maintain a clear separation between the global explorative and local exploitative search efforts, it soon became clear [5] that other factors such as the number of parallel local searches influence the exploration vs. exploitation balance. Several empirical studies [6, 7] tried to shed light on the properties of the Bees Algorithm and related optimisation techniques, and how their parameterisation affects the search effort. However, to the best of the authors' knowledge, a theoretical analysis of the Bees Algorithm behaviour was never attempted.

This paper addresses the above gap in the literature. Understanding in detail the dynamic behaviour of a population-based optimisation algorithm on arbitrarily complex fitness landscapes is extremely complex. For this reason, the literature on nature-inspired algorithms overwhelmingly relies on qualitative biological analogies and empirical comparisons. However, by investigating well-defined cases under a theoretical framework, important insights on the algorithm behaviour can be gained [8]. This study focuses on the local search procedure of the Bees Algorithm, using the clearly delimited boundaries of the site neighbourhood to infer important properties.

The proposed study is timely, as a large number of variants in operators and parameterisations have been developed for this popular algorithm [9]. In the light of the *No Free Lunch Theorem* [10], it is essential to unravel the implications of these different choices of operators and parameters.

The rest of this paper is organised as follows: a general formulation of the Bees Algorithm is given in section 2. The main variants of (section 3) and related algorithms to (section 4) the Bees Algorithm are discussed. The second part of the paper focuses on the local search (section 5) and site abandonment (section 6) procedures. Section 7 analyses the implications of using different neighbourhood shapes. The main findings are discussed in section 8, and the conclusions are drawn in section 9.

2. Formal Definition of the Bees Algorithm

The Bees Algorithm iteratively looks for better solutions to a specified optimisation problem. The algorithm is terminated when a given stopping criterion is met (e.g. a pre-set number of optimisation cycles has elapsed, a solution of satisfactory quality is found). Despite minor differences, the notation concerning the main parameters and operators of the Bees Algorithm is consistent in the literature. With some minor changes, it is also used in this paper:

- **ns** number of scout bees used *only* in the global search;
- **nb** number of sites where local search is performed;
- **nr** number of recruited forager bees for each of the nb sites;
- **stlim** number of cycles of local stagnation before a site is abandoned;
- **ngh** initial neighbourhood size of the nb sites;
- α neighbourhood shrinking parameter ($0 < \alpha < 1$);

In the standard Bees Algorithm, the parameter ns describes the total number of scouts used for random exploration (here ns) plus the number of scouts (nb) marking the neighbourhoods (sites) selected for local search. That is, $ns^{standard} = ns + nb$. Also, it is customary to allocate a larger number of foragers (nre) to the very best $ne < nb$ (*elite*) sites, and less ($nrb < nre$) to the remaining $nb - ne$ best sites. This distinction is not necessary for the analysis proposed in this paper, and for the sake of compactness is dropped. Henceforth, the parameter nr will refer likewise to nre or nrb.

In this study only continuous optimisation is considered, and each solution is represented by an N-dimensional vector of real-valued decision variables $s^g = \{s^g[1], ..., s^g[N]\} \in \mathbb{R}^n$. The solutions are evaluated by a fitness function $\mathscr{F}$ specific to the problem domain, which the algorithm aims to maximise. The analysis of this paper is equally valid for a minimisation problem ($\min\{\mathscr{F}(\cdot)\} \equiv \max\{-\mathscr{F}(\cdot)\}$).

In this paper, each of the $s \in \{s^{(1)}, \dots, s^{(nb)}\}$ nb sites selected for local search is denoted by a centre s^g and two additional variables: the *time-to-live* integer variable s^{ttl}, and the local search *edge* s^e. The time-to-live variable s^{ttl} is a counter that indicates the number of remaining cycles of stagnation before the site is abandoned. The edge s^e defines the current spatial extent (henceforth called *search scope*) of the local search.

For the sake of simplicity, unless otherwise stated, all the decision variables will be henceforth defined in the same interval. Accordingly, s^e and ngh are scalars, and the search scope at a given site s is delimited by a *hypercube* $\mathcal{C}$ of edge s^e centred in the solution s^g. Hereafter, this region will be indicated as $\mathcal{C}(s^g, s^e)$. Local search is performed uniformly sampling nr solutions inside $\mathcal{C}(s^g, s^e)$. In the general case that the interval of definition is not equal for all parameters, s^e and ngh will be defined as vectors of size N. In this case, local search is performed inside a *box* (i.e. an N-*orthotope*) of edges $s^e = \{s^e[1], \dots, s^e[N]\}$ centred in s^g. When relevant, the consequences of using box sampling rather than cubic sampling will be discussed.

The algorithm steps are described in box 1. Except for minor changes (i.e. no *elite* sites), the procedure described in box 1 can be regarded as the Standard Bees Algorithm (*SBA* [5]). In the neighbourhood shrinking procedure

Bees Algorithm: Main Steps

1. (**Initialisation**) The initial population P is created sampling $ns+nr\cdot nb$ solutions at random across the N-dimensional solution space. Each solution s^g marks the centre of a neighbourhood (*site*) s of edge s^e (i.e. a *search scope* $\mathcal{C}(s^g, s^e)$). The two variables s^{ttl} and s^e are initialised for each site as follows: $s^{ttl} = stlim$ and $s^e = ngh \cdot (M - m)$, where M and m are respectively the upper and lower limit of the interval of definition of the variables;
2. (**Selection**) The best nb sites $s^{(1)}, \ldots, s^{(nb)}$ centred on the solutions of highest fitness are selected from P for local search (**Waggle Dance**), the others are removed from the population;
3. (**Local Search**) For each of the nb sites $s \in \{s^{(1)}, \ldots, s^{(nb)}\}$ in P selected for local search, the following steps are performed:
   (a) If $s^{ttl} = 0$ the site is abandoned (**Site Abandonment**), and nr new solutions are randomly sampled across the search space. The best v of these nr solutions is used as the centre for a new site s, which is initialiased with $s^{ttl} = stlim$ and $s^e = ngh \cdot (M - m)$;
   (b) (**Foraging**) If $s^{ttl} > 0$, nr solutions $v_1, \ldots, v_{nr}$ are randomly sampled with uniform distribution within $\mathcal{C}(s^g, s^e)$. The solution v of highest fitness is selected, whilst the other solutions are discarded:
      i. if $\mathscr{F}(v) > \mathscr{F}(s^g)$, v replaces s^g as the site centre ($s^g = v$). The time to live is set to $s^{ttl} = stlim$ and the edge is kept unchanged (s^e);
      ii. If $\mathscr{F}(v) \leq \mathscr{F}(s^g)$, the local search is said to stagnate. The edge is reduced by a constant factor $s^e = \alpha s^e$ (**Neighbourhood Shrinking**), and the time to live is reduced to $s^{ttl} = s^{ttl} - 1$;
4. (**Global Search**) ns new (*scout*) solutions $v_1, \ldots, v_{ns}$ are uniformly sampled in the search space. They become the centres of ns new sites, each initialised with $s^{ttl} = stlim$ and $s^e = ngh \cdot (M - m)$.
5. (**Population Update**) The nb scouts marking the centres of the sites evolved via local search (3) and the ns scouts created by global search (4) make up the new population P;
6. (**Stopping Criterion**) if the stopping criterion is not met go back to step 2, otherwise return the solution of highest fitness found;

Algorithm 1: The Bees Algorithm

(described in step 3b) if the interval of definition of the parameters is not the same for all variables, the i-th dimension of the box is reduced as $s_i^e = \alpha s_i^e$. The initialisation and site abandonment procedures are designed to keep constant at each generation the sampling rate of the solution space[1].

Local search aims to find the fitness optimum within a neighbourhood centred on a promising solution. Because the centre of the neighbourhood is updated as better solutions are found (step 3), the scope of local search dynamically changes, and eventually includes the local attractor point in the search space (i.e. a local optimum). It should be noted that, like any stochastic optimisation procedure, local search is not guaranteed to stop at the local optimum. In particular, local search may be prematurely abandoned when *(i)* it stagnates for *stlim* iterations (e.g. stops on a flat surface) or *(ii)* global search finds more promising regions (fitter solutions) elsewhere in the search space (step 2).

Global random search aims to find previously unexplored regions of high fitness in the search space. Global search can also be used to increase adaptation to changes in dynamic fitness landscapes. The solutions found via local (i.e. the centres of the *nb* neighbourhoods) and global search are ranked at the end of every optimisation cycle, and the fittest *nb* solutions are kept as seeds (centres) for the next optimisation cycle. As the local exploitation of one given site progresses, the probability that this site is abandoned because random search found a fitter solution decreases. For this reason, some authors do not use global search [5], or give randomly generated solutions (*young bees*) time to 'grow up' [11].

3. Main Variants of the Bees Algorithm

Besides the standard procedure described in section 2, many different variants of the Bees Algorithm exist. A recent survey [9] separated three main branches of the Bees Algorithm, namely the *Basic Bees Algorithm* (BBA), the *Shrinking-based Bees Algorithm* (ShBA), and the *Standard Bees Algorithm* (SBA).

The BBA refers to the basic form of the Bees Algorithm, which is mentioned in several articles [5, 12, 13, 14] and performs parallel local searches using *fixed* neighbourhoods (i.e. no neighbourhood shrinking).

The ShBA [1] includes the neighbourhood shrinking procedure. The heuristics behind the neighbourhood shrinking procedure is to intensify the exploitation effort as the local search progresses, focusing the sampling on increasingly smaller regions of the solution space.

Finally the SBA, so termed in numerous articles [5, 11, 15], includes the neighbourhood shrinking and site abandonment procedures. The heuristics behind site abandonment is to abandon a site once the local search stagnates, to avoid being trapped into local fitness peaks.

[1] This is particularly useful when the performance of the BA is compared with the performance of other techniques.

Many recruitment, neighbourhood alteration, and site abandonment heuristics were proposed in the literature.

Ghanbarzadeh [16] proposed two methods for setting the number of recruited foragers proportionally to a) the fitness or b) the location of the sites. Other authors proposed recruitment schemes where the number of foragers was proportional to the fitness of the site, and decreased it progressively by a fixed amount [13], or according to a *fuzzy logic* policy [17]. Pham et al [18] used Kalman filtering to allocate number of bees to the sites selected for local search. This strategy was used to train a Radial Basis Function neural network, and improved the learning accuracy and speed of the neural network. Finally, Imanguliyev [19] proposed a recruitment scheme where the number of foragers for a site was computed on the *efficiency rate* of the site, rather than its fitness score.

In its basic instance [16], the search scope of a site is changed (reduced) when local search fails to improve. Ahmad [20] proposed two different methods to change dynamically the neighbourhood of a site: a) BA-NE where the search scope is *increased* if a better solution is found and kept invariant otherwise, and b) BA-AN, where the neighbourhood is *asymmetrically* increased along the direction that led to the last improvement and decreased otherwise.

When a site is abandoned, the best-so-far local solution is usually kept in memory [5]. However, in some cases [13, 21] all the local solutions found before abandoning a site are retained for later use. In Hierarchical Site Abandonment [17] when a site s is abandoned, all the other sites with fitness lesser or equal to s are abandoned too.

4. Related Techniques

The Bees Algorithm is part of a large family of metaheuristics mimicking the collective intelligence of honey bees [22]. Despite the common inspiration, these algorithms often differ for the operators used and the solution sampling strategy. The main differences between the Bees Algorithm and a number of other algorithms [23] based on the foraging behaviour of honey bees have been discussed in the literature [5, 6, 7]. An exhaustive comparison would be beyond the scope of this paper. This section will focus on two algorithms that have important similarities with the local search strategy of the Bees Algorithm: Variable Neighbourhood Search (VNS) [24] and *LJ* Search [25].

4.1. Variable Neighbourhood Search

Variable Neighbourhood Search iterates cycles of local search around a seed solution using neighbourhoods of different size. This idea has been successfully used [26] in numerous applications, like the Traveling Salesman [24] and the Vehicle Routing [27, 28] problems. The main steps of the VNS algorithm are:

Variable Neighbourhood Search: Main Steps

1. Initialise k neighbourhoods $S_1, \ldots, S_k$ of variable size around a randomly generated centre x, initialise $i = 1$;
2. Take neighbourhood S_i:
   (a) sample a solution v uniformly inside the neighbourhood S_i;
   (b) apply a local search procedure using v as seed to find a new solution v';
      i. if v' is fitter that x, set $x = v'$ and $i = 1$;
      ii. else set $i = i + 1$;
3. If $i > k$, terminate the algorithm and return the best found solution, otherwise iterate from step 2;

VNS is akin to local neighbourhood search at a BA site. A variant of this approach, called Reduced VNS (RVNS) [29, 30], skips the local search at step 2b, to keep $v' = v$. RVNS is equivalent to a BBA where $nb = 1$, $nr = 1$, and $ns = 0$ (no global search). Parallel VNS [31, 32] resembles more closely the BA, since it performs a number of concomitant local VNS search procedures.

The main difference between VNS and the BA is that the former uses a randomly generated sequence of neighbourhood sizes for local search, whilst the latter uses a fixed (BBA) or deterministically shrunk (ShBA) neighbourhood size. In addition, VNS uses a fixed number of foragers per optimisation cycle, whilst the number of foragers is determined by the quality of the neighbourhood in the BA. That is, as mentioned in Section 2, the standard Bees Algorithm allocates more foragers to the most promising (*elite*) ne neighbourhoods, and less to the remaining neighbourhoods. The decision on the allocation of foragers per site is dynamically updated at every optimisation cycle

4.2. LJ Search

The *LJ* Search Method was successfully used to optimise feedback control in nonlinear systems [33], as well as time-optimal [34] and time-delay [35, 36] systems. Given an N-dimensional minimisation problem, the *LJ* Search Method pseudocode is:

LJ **Search: Main Steps**

1. Let s^e_1 be the N-dimensional vector of the spans ($max - min$ value) of the interval of definition of the N decision variables, s_1 the seed solution, and $t = 1$ an index;

2. Sample nr solutions $v_1, \ldots, v_{nr}$ as follows:

$$v_i = s_t + u \cdot s^e_t \qquad i \in \{1, \ldots, nr\}$$

   where u is a vector of N real values independently sampled with uniform distribution in $[-0.5, 0.5]$;

3. Compute the next solution as:

$$s_{t+1} = \arg\min_v \{\mathscr{F}(v) \mid v \in \{s_t, v_1, \ldots, v_{nr}\}\}$$

4. Reduce the ranges of a given factor $0 < \alpha < 1$:

$$s^e_{t+1} = \alpha s^e_t$$

   and increment the counter $t = t + 1$;

5. check if the stop criterion is met. If so return s_t, otherwise go back to step 2;

Except for the initialisation of the neighbourhood, the *LJ* Search algorithm closely resembles the BA local search procedure at one site with neighbourhood shrinking. That is, the ShBA can be described as a multi-*LJ* Search method. Surprisingly, to the best of the authors' knowledge, the similarity between *LJ* Search and the Bees Algorithm has so far been overlooked in the literature.

A mathematical analysis of the properties of the *LJ* Search method was published by Gopalakrishnan [37]. In particular, Gopalakrishnan [37] proved that the succession of solutions $s_1, \ldots, s_n$ converges to a local optimum. Unfortunately, the proof of convergence is only valid for an infinite number of iterations. Moreover, in his analysis Gopalakrishnan [37] took in consideration hyperspherical neighbourhoods instead of the hypercubic ones that are actually used in the *LJ* Search algorithm. As will be shown in section 7, the use of different neighbourhoods has important consequences on the properties of the search.

The next sections analyse the properties of the local search procedure in the Bees Algorithm.

5. Analysis of Local Search Properties

This section clarifies the properties of the local search procedure of the SBA, such as its average step size and speed of convergence. These properties are estimated from an *a-posteriori* analysis on the 'lifetime' of a generic site s, from its discovery by a scout to abandonment when local search *stalls* (i.e. it fails to progress for *stlim* iterations). The case that the site is replaced by a more promising site found via global search is not included. If needed, the results of the below analysis are applicable to describe the behaviour of local search from any point in time, not necessarily the discovery of the site, until abandonment. Importantly, the final solution may not be the local optimum, that is, local search may only provide an approximation of the local optimum.

5.1. Local Search: Introduction and Definitions

The following nomenclature will be used:

- s is the site, described at any iteration (*cycle*) t by a triple $s_t = \{s_t^g, s_t^e, s_t^{ttl}\}$: the centre s_t^g, the edge s_t^e and the time to live s_t^{ttl};
- n is the number of local search cycles from discovery to abandonment of the site;
- s_1^g is the starting point (site centre) of the local search procedure;
- s_t^g with $1 < t < n$ is the site centre after t local search cycles;
- s_n^g is the final result of the local search, namely, the neighbourhood centre at the last local search cycle, before the site is abandoned (i.e. $s_n^{ttl} = 0$);
- S is the series of solutions found by local search at site s:

$$S = \{s_1^g, \dots, s_n^g\} \tag{1}$$

The solution found in the t^{th} local search cycle $L_{nr}(s_t^g)$ can be formalised as the result of the following endomorphic function (maximisation problem):

$$L_{nr}(s_t^g) = \arg\max_v \{\mathscr{F}(v) \mid v \in \{s_t^g, v_1, \dots, v_{nr}\}\} \tag{2}$$

with $v_i \sim \mathcal{C}(s_t^g, s_t^e)$ as a uniform sampling of a solution v_i in the hypercube centred in s_t^g with edge s_t^e. This sampling will be further discussed in section 7.

Since this is an *a posteriori* analysis, it will be assumed that every set of candidate solutions sampled at each cycle is known.

A local maximum L_{opt} is defined as:

$$L_{opt} \text{ is a l.m.} \Leftrightarrow \exists \epsilon > 0 \mid \mathscr{F}(L_{opt}) \geq \mathscr{F}(v) \tag{3}$$

for all $v \in \mathcal{C}(L_{opt}, \epsilon)$. If L_{opt} is the optimum of the subregion $\mathcal{C}(s_t^g, s_t^e)$, the operator L_{nr} provides a stochastic approximation of the local optimum within $\mathcal{C}(s_t^g, s_t^e)$. The expected quality of this approximation increases monotonically with the number nr of candidate solutions sampled.

$$L_{opt} = \lim_{nr \to \infty} L_{nr}(s) = \arg\max_v \{\mathscr{F}(v) \mid v \in \mathcal{C}(s_t^g, s_t^e)\} \tag{4}$$

The series of solutions S defined in eq. (1) shares the same convergence properties of the LJ Search proved in [37]. Namely, without site abandonment, a number of steps n exists such that the series of solutions S will eventually converge to a local optimum.

Due to the monotonically increasing nature of the series $\mathscr{F}(s_1^g), \ldots, \mathscr{F}(s_n^g)$ (see eq. (2)), s_n^g is the best solution found in the n iterations of local search at site s.

The standard neighbourhood shrinking heuristic can be formally defined for a hypercube as follows ($0 < \alpha < 1$):

$$s_{t+1}^e = \begin{cases} s_t^e & L_{nr}(s_t^g) \neq s_t^g \\ \alpha s_t^e & L_{nr}(s_t^g) = s_t^g \end{cases} \tag{5}$$

Neighbourhood shrinking can be similarly defined in the more general case of *box* sampling, where the interval of definition is not the same for all the N variables. In this latter case, s_t^e is the N-dimensional vector of search scope edges, and each edge is reduced of the same fixed factor α. It also holds for isotropic (hyperspherical) sampling. The standard site abandonment heuristic is applied when:

$$s_n^g = s_{n-1}^g = \cdots = s_{n-stlim}^g \tag{6}$$

In other terms, the local search is terminated when *stlim* consecutive *fix points*[2] of L_{nr} are found. It is also true that:

$$s_t^g = s_{t+k}^g \;\Leftrightarrow\; s_t^g = s_{t+1}^g = \cdots = s_{t+k}^g \tag{7}$$

for any positive index $k \in \mathbb{N}$.

The above definitions and properties are shared by most BA techniques, or can be easily modified to include other BA variants.

Proposition 1 (Search scope Intersection: Successive Solutions)**.** *Given two consecutive solutions s_t^g and s_{t+1}^g, s_t^g is included in the search scope of s_{t+1}^g*

$$s_t^g \in \mathcal{C}(s_{t+1}^g, s_{t+1}^e) \tag{8}$$

Proof. The proof is trivial: if local search stagnates at cycle t, $s_{t+1}^g = s_t^g$ and $s_{t+1}^e < s_t^e$ (neighbourhood shrinking). Then $s_t^g \in \mathcal{C}(s_{t+1}^g, s_{t+1}^e) = \mathcal{C}(s_t^g, s_{t+1}^e)$. If local search progresses at cycle t, $s_{t+1}^g \neq s_t^g$ and $s_{t+1}^e = s_t^e$ (no neighbourhood shrinking). Remembering that $s_{t+1}^g \in \mathcal{C}(s_t^g, s_t^e)$, it follows that $s_t^g \in \mathcal{C}(s_{t+1}^g, s_t^e) = \mathcal{C}(s_{t+1}^g, s_{t+1}^e)$. □

This property also holds in case of box and hyperspherical sampling is used.

[2]A fix point x of a function f is a point such as $f(x) = x$

5.2. Bounds on Reach

Hereafter, the distance in the solution space that local search is able to cover at a given site in a given number of cycles will be indicated as the *reach* of local search. That is, the reach is the distance between the starting point of local search (s_1) and the best approximation of the local optimum after n cycles (s_n), namely $d(s_1, s_n)$. The upper and lower boundaries of the reach are defined as follows:

Proposition 2 (Reach)**.** *The reach of local search in n learning cycles at a given site s centred on solution s_1^g is bounded within the $\left[0, n\frac{s_1^e\sqrt{N}}{2}\right]$ interval, where s_1^e is the site edge at the start of the search.*

Proof. Minimum reach occurs when local search stalls since the very beginning, namely $s_t^g = s_{t+1}^g\ \forall t \in \{1, \ldots, n = stlim\}$, and thus $s_1^g = s_n^g$.

At cycle t, local search is bounded within the hypercube $\mathcal{C}$ centred in s_t^g, where the farthest solutions lie at the four vertexes of $\mathcal{C}$. To attain maximum reach, local search must progress at each cycle, so as the initial site edge s_1^e is not reduced (i.e. no neighbourhood shrinking). The maximum step size per cycle is bounded by the distance between the centre and a vertex of the N-dimensional hypercube $\mathcal{C}$, that is $d_v = {}^{s_t^e\sqrt{N}}/_2$. The upper bound of the reach at a given site is therefore n times d_v. □

Proposition 2 gives the boundaries of 'how far' local search can travel in n learning cycles. The maximum step size is achievable only when the segment that joins s_1^g to s_n^g is parallel to the diagonal of the hypercube $\mathcal{C}$, and every pair of subsequent solutions s_t^g and s_{t+1}^g $(1 \leq t < n)$ are distant $d(s_t^g, s_{t+1}^g) = {}^{s_t^e\sqrt{N}}/_2$. For example, this would be the case of a fitness landscape consisting of a sloped hyperplane aligned with the diagonal of the hypercube $\mathcal{C}$, or a hypersphere of centre c lying in the direction of one of the diagonals of the hypercube centred in s_1^g.

Considering unitary time steps per iterations, the reach can be considered as a measure of the 'travelling speed' of the local search in the solution space. Closely related to the reach is the *convergence time* of the local search (i.e. the number of iterations taken to reach the local attractor). If the distance between the centre of the site s_1 and the optimum L_{opt} is $d(s_1^g, L_{opt})$, according to Proposition 2 the minimum number of iterations n_{min} required to reach L_{opt} are:

$$n_{min} = \frac{2d(s_1^g, L_{opt})}{s_1^e\sqrt{N}} \tag{9}$$

This is the lower bound on the convergence time, and can be used to evaluate the efficiency of local search on different fitness landscapes.

In the more general case of asymmetric boundaries (i.e. box sampling), the maximum reach can be computed as follows:

$$max_{reach} = \frac{n}{2}\sqrt{\sum_{i=0}^{N-1} s_1^e[i]^2} \tag{10}$$

where $s_1^e[i]$ is the i-th component of vector s_1^e. Finally, it is worth mentioning that most - if not all - BA variants use a hypercube to define the scope of local search. Consequently, the foragers are sampled inside an anisotropic region, and the maximum reach depends on the orientation of the segment that joins s_1^g to L_{opt} respect to the diagonal of the hypercube.

5.3. Expected Progress

To understand the behaviour of local search, it is useful to calculate the expected step size of one iteration of the procedure under certain assumptions on the fitness surface.

Proposition 3 (Expected Step Size)**.** *Given a strict monotonic increasing one-dimensional fitness landscape in* $[0, \ell] \in \mathbb{R}$ *(e.g. a straight line), and a site centred in* $s_t^e = \ell/2$ *with edge* $s_t^g = \ell$*, the expected step size of one local search iteration (i.e. the average distance between* s_t^g *and* s_{t+1}^g*) is:*

$$d(s_t^g, s_{t+1}^g) = \ell \frac{0.5^{nr+1} + nr}{nr + 1} - \frac{\ell}{2} \tag{11}$$

Proof. See electronic appendix. □

The result of proposition 3 is valid for any locally monotonic fitness slope, as long as $\mathcal{C}$ is fully in the monotonic region. Its validity extends also to multi-dimensional surfaces such as regions of hyperplanes or hyperspheres. For this to happen, the search scope must be isotropic (i.e. a hypersphere), the fitness landscape must be strictly monotonic along the straight line joining the centre of $\mathcal{C}$ to the local fitness maximum, and the fitness landscape inside the search scope must be symmetric respect to said straight line. If these conditions are verified, the expected step size will be given by eq. (11) for the direction where the slope is monotonic, and zero (no bias) in the other directions. The above conditions apply in the common case where local search is climbing one side of a fairly regular hill or slope, but $\mathcal{C}$ does not include the fitness maximum yet.

5.3.1. Expected Step Size: Experimental Verification

The theoretical result of proposition 3 was verified numerically (figure 1) on three 2D cases: on an inclined plane (leftmost column), near the top of a spherical hill (the hill top is at the border of the search scope, rightmost column), and far from the top of the spherical hill (middle column). The neighbourhood is a circle of radius 0.5 centred in $\{0.5, 0.5\}$. In all cases, the fitness surface is monotonically increasing along the horizontal diameter line, and symmetric with respect to that line. The number of foragers was varied ($nr = 1, 10, 20$). The plots show that the average (red triangle) of 10^3 independent local search trials is always in good agreement with the theoretical expectation (at the bottom of each sub-figure) along the horizontal diameter line (where the fitness landscape is monotonic), and aligned to the centre of the search scope in the vertical direction (i.e. no bias in the vertical direction).

Far from the peak, where the curvature of the sphere is small, the spread of the solutions on the fitness landscape is large, and indistinguishable from the spread on the planar surface. Near the hill top, where the curvature is large, the solutions are tightly clustered near the fitness maximum. This behaviour suggests that local search becomes increasingly focused and exploitative as it approaches the local fitness maximum.

Figure 1 also shows little difference between the spread of the solutions obtained using 10 and 20 foragers. Indeed, the expected step size grows in sublinear fashion with the number of foragers (eq: 11). Figure 2 shows how the average step size of 10^4 independent local search trials varies with the number of foragers (nr). Also in this case, the search trials were performed in a circle of radius 0.5 centred in $\{0.5, 0.5\}$, and the plot shows the result of local search (s_{t+1}) along the direction of the slope of an inclined plane. The numerical averages (blue dot) are in good agreement with the theoretical expectations of eq. (11) (red line). The plot highlights how the result of local search quickly reaches the borders of the neighbourhood, that is the asymptotic value of 1. In general, it can be said that the size of the neighbourhood determines more than the number of foragers the ability of local search to quickly climb (descend) a fitness slope.

6. Site Abandonment: Local Search Stalling Probability

This section analyses the probability that a site may be abandoned due to lack of progress of local search.

6.1. Site Abandonment: Definitions and Properties

Let s_t^g be the centre of site s at the t^{th} local search cycle and $\mathcal{C}(s_t^g, s_t^e)$ the search scope. Let $\mathcal{LR}_{s_t}$ and $\mathcal{GR}_{s_t}$ be the subregions of $\mathcal{C}$ including solutions of respectively lower or equal, and higher fitness. Explicitly:

$$\begin{aligned} \mathcal{LR}_{s_t} \subseteq \mathcal{C}(s_t^g, s_t^e) \quad & v \in \mathcal{LR}_{s_t} \Leftrightarrow \mathscr{F}(v) \leq \mathscr{F}(s_t^g) \\ \mathcal{GR}_{s_t} \subset \mathcal{C}(s_t^g, s_t^e) \quad & v \in \mathcal{GR}_{s_t} \Leftrightarrow \mathscr{F}(v) > \mathscr{F}(s_t^g) \end{aligned} \tag{12}$$

where

$$\mathcal{LR}_{s_t} \cup \mathcal{GR}_{s_t} = \mathcal{C}(s_t^g, s_t^e) \tag{13}$$

According to the above definitions, it can be said that local search progresses if the output of the endomorphism L_{nr} belongs to $\mathcal{GR}_{s_t}$:

$$\left[L_{nr}(s_t^g) = s_{t+1}^g \neq s_t^g\right] \Leftrightarrow s_{t+1}^g \in \mathcal{GR}_{s_t} \tag{14}$$

In general, $\mathcal{LR}_{s_t}$ and $\mathcal{GR}_{s_t}$ may include non-contiguous subregions, since the region covered by $\mathcal{C}$ may contain several local optima.

Hereafter, the border (hypersurface) of an N-dimensional region A will be indicated as A^-, and the volume of A will be indicated as $\mathcal{V}(A)$. To analyse the

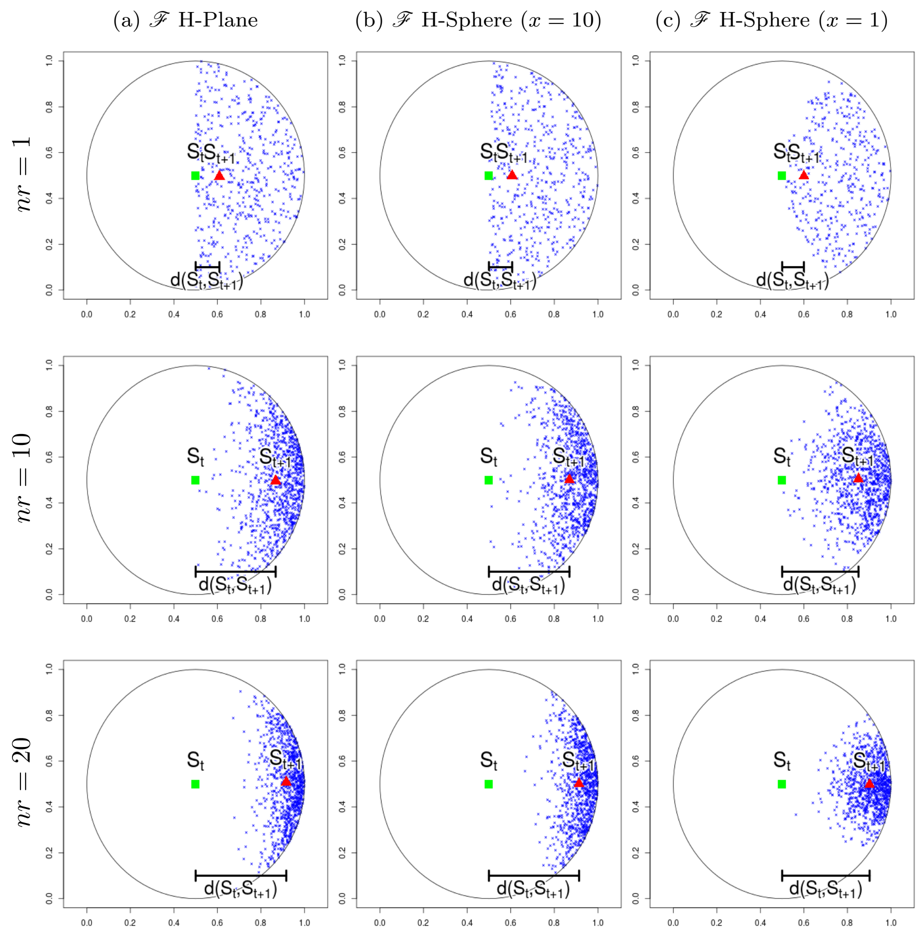

Figure 1: Search results (s_{t+1}) of 10^3 independent local search trials in three 2D fitness landscapes: a plane sloped in the horizontal direction (left column), a hypersphere with centre in $x = 10, y = 0.5$ (middle column), and a hypershpere centred in $x = 1, y = 0.5$ (right column). An isotropic circular search scope of centre $s_t^g = [0.5, 0.5]$ (green square) and radius $s_t^r = 0.5$ was used. The number of foragers nr was set to 1 (top row), 10 (middle row), and 20 bottom row. The blue dots represent the solutions found in the local search trials, and their arithmetic average is marked by the red triangle. The maximum is always on the border of the search scope, at the right-end extreme of the horizontal diameter line. At the bottom of each panel, the expected step size (11) in the direction of the maximum is shown.

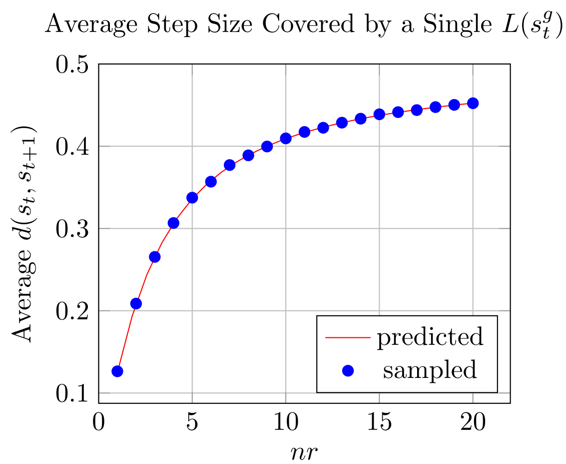

Figure 2: Result of local search using an isotropic search scope of radius $s_t^r = 0.5$ and centred in $s_t^g = \{0.5, 0.5\}$ on a sloped planar fitness surface. The predicted value (red line) along the direction of the slope was calcuated from eq. (11), and closely matches the average values of 10^4 independent local search runs (blue dots).

likelihood that local search stalls and the site is abandoned, it is useful to define the following two ratios:

$$|\mathcal{LR}_{s_t}| = \frac{\mathcal{V}(\mathcal{LR}_{s_t})}{\mathcal{V}\left(\mathcal{C}(s_t^g, s_t^e)\right)} \qquad |\mathcal{GR}_{s_t}| = \frac{\mathcal{V}(\mathcal{GR}_{s_t})}{\mathcal{V}\left(\mathcal{C}(s_t^g, s_t^e)\right)} \tag{15}$$

within the local search scope $\mathcal{C}(s_t^g, s_t^e)$, $|\mathcal{LR}_{s_t}|$ and $|\mathcal{GR}_{s_t}|$ represent the fraction of space where solutions of respectively lower and higher fitness lie. That is, they represent the *relative coverage* of $\mathcal{C}$ of the two regions $\mathcal{LR}_{s_t}$ and $\mathcal{GR}_{s_t}$. In particular, $|\mathcal{GR}_{s_t}|$ represents the probability that one random sample of the search scope yields a solution of higher fitness than s. From eq. (15), the following properties hold:

$$0 < |\mathcal{LR}_{s_t}| \leq 1 \quad 0 \leq |\mathcal{GR}_{s_t}| < 1 \quad |\mathcal{LR}_{s_t}| + |\mathcal{GR}_{s_t}| = 1 \tag{16}$$

Also, from eq. (15) it follows that a solution s_t^g is the optimum of the subregion $\mathcal{C}(s_t^g, s_t^e)$ iff:

$$\mathscr{F}(s_t^g) \geq \mathscr{F}(v) \quad \Leftrightarrow \quad |\mathcal{LR}_{s_t}| = 1 \wedge |\mathcal{GR}_{s_t}| = 0 \tag{17}$$

for all $v \in \mathcal{C}(s_t^g, s_t^e)$. The local exploitative search of the BA aims to locate L_{opt}, inside the search scope $\mathcal{C}(s_t^g, s_t^e)$. If the $\mathcal{GR}_{s_t}$ region is significantly smaller than the search scope, the probability of finding a better solution than s^g is small, and progress may be slow or stop. The neighbourhood shrinking procedure may mitigate this problem, progressively reducing the search scope and increasing the probability that a forager is generated inside $\mathcal{GR}_{s_t}$. For this to happen, neighbourhood shrinking needs to keep $\mathcal{GR}_{s_t}$ inside the search scope. Unless it is a local optimum, it can be shown that the site centre s^g is at least contiguous to $\mathcal{GR}_{s_t}$. That is:

Proposition 4. *A solution s^g is either a local optimum of the fitness function $\mathscr{F}$, or lies on the border $\mathcal{GR}^-_{s_t}$ of $\mathcal{GR}_{s_t}$.*

Proof. This can be proven by contradiction:

$$s^g \notin \mathcal{GR}^-_{s_t} \Leftrightarrow \exists \epsilon > 0 \mid v \in \mathcal{LR}_{s_t} \quad \forall v \in B(s^g, \epsilon) \tag{18}$$

where $B(s^g, \epsilon)$ is an N-dimensional ball of radius ϵ and centred in s^g. However:

$$v \in \mathcal{LR}_{s_t} \Leftrightarrow \mathscr{F}(v) \leq \mathscr{F}(s^g) \tag{19}$$

so s^g is either a local optimum or is inside $\mathcal{GR}^-_{s_t}$. □

This property holds for virtually any BA variant. An important consequence is that there is no neighbourhood reduction that completely excludes $\mathcal{GR}_{s_t}$ from the search scope. Accordingly, neighbourhood shrinking does not affect the ability of local search to progress to higher regions of fitness. If the $\mathcal{GR}_{s_t}$ region includes multiple peaks and s^g is not on the main peak, neighbourhood shrinking might make the main peak unreachable. If instead the fitness landscape is locally convex, neighbourhood shrinking will not affect the ability of local search to reach the main peak in $\mathcal{GR}_{s_t}$. However, neighbourhood shrinking affects the lower bound of the convergence time towards the optimum (eq. 9). Thus, it can be said that neighbourhood shrinking is most beneficial in the latest iterations of the local search at a site, when a good local optimum - or the global optimum - has been approximately located and the search focus is shifted from speed to the accuracy of the solution. In many cases, it can be argued that local search is indeed more likely to stall (and hence neighbourhood shrinking to be performed) once local search approaches the local peak, and $\mathcal{GR}_{s_t}$ becomes increasingly small.

6.2. Stalling Probability Without Neighbourhood Shrinking

Let us consider a site s centred on s^g_t at cycle t. The probability that local search without neighbourhood shrinking stalls at s will be henceforth indicated as $P(s^g_t = s^g_{t+s^{ttl}_t})$. It is computed as follows:

Proposition 5 (Stalling Probability Without Shrinking)**.** *Given site s centred on s^g_t at cycle t, the probability that local search without neighbourhood shrinking stalls is:*

$$P(s^g_t = s^g_{t+s^{ttl}_t}) = |\mathcal{LR}_{s_t}|^{nr \cdot s^{ttl}_t} \tag{20}$$

Proof. See electronic appendix. □

One important aspect of the stalling probability is that, since local search is random, it is not affected by the slope of the fitness surface. Proposition 5 is valid regardless whether $s_t^{ttl} = stlim$, that is, local search has not begun to stagnate yet, or $s_t^{ttl} < stlim$ and local search has already begun to stagnate.

Variants that use a dynamic assignment of foragers, like [13, 18, 17], yield a more complex behaviour that leads to a different stalling probability formulation. Some ideas on how to deal with these variants will be discussed later in this section. If neighbourhood shrinking is used, the progressive reduction of the search scope needs to be taken into account. In this case, it is possible that if local search is trapped in a secondary peak, the $\mathcal{GR}_{s_t}$ region may be lost as the search scope is reduced.

6.3. Stalling Probability With Neighbourhood Shrinking

Let us consider the case where after t cycles, local search stagnates for k cycles at s_t^g inside the basin of attraction of a local optimum L_{opt} ($s_t^g \neq L_{opt}$ and $\mathcal{GR}_{s_t}$ is one unique region). In this case, proposition 4 stipulates that s_t^g lies on the border of $\mathcal{GR}_{s_t}$. The most likely cause for repeated stalling of local search is that $\mathcal{GR}_{s_t}$ is small compared to $\mathcal{C}(s_t^g, s_t^e)$. However, if $\mathcal{GR}_{s_t}$ is small in comparison with $\mathcal{C}(s_t^g, s_t^e)$, and s_t^g is on the border of $\mathcal{GR}_{s_t}$, the distance d between s_t^g and L_{opt} is likely to be small compared to the search edge ($d(s_t^g, L_{opt}) \ll s_t^e$). That is, $\mathcal{GR}_{s_t}$ is likely to be far from the border of the search scope, and is not going to be changed by the neighbourhood shrinking procedure. One possible exception would be that $\mathcal{GR}_{s_t}$ was very long and thin (e.g. a narrow valley in a 2D space), and s_t^g and L_{opt} were on the two opposite sides of $\mathcal{GR}_{s_t}$ (in the 2D example, on the two sides of the valley). In this case. neighbourhood shrinking might reduce $\mathcal{GR}_{s_t}$, although of a very small amount (one extreme of the valley might be cut out).

If the $\mathcal{GR}_{s_t}$ region at time t is unchanged after k successive applications of the neighbourhood shrinking procedure, it is possible to compute the relative coverages (eq. 15) of $\mathcal{LR}_{s_{t+k}}$ and $\mathcal{GR}_{s_{t+k}}$ as follows:

Lemma 1 (Coverage Reduction with constant $\mathcal{GR}_{s_t}$)**.** *Let s_t^g be the centre of site s at cycle t in the N-dimensional solution space. If local search stagnates for k cycles ($s_t^g = s_{t+k}^g$), and the region $\mathcal{GR}_{s_t} \neq \emptyset$ is not changed by neighbourhood shrinking, the relative coverages of $\mathcal{LR}_{s_{t+k}}$ and $\mathcal{GR}_{s_{t+k}}$ become:*

$$|\mathcal{LR}_{s_{t+k}}| = \frac{1}{\alpha^{kN}}(|\mathcal{LR}_{s_t}| - 1) + 1 = 1 - \frac{1}{\alpha^{kN}}|\mathcal{GR}_{s_t}| \tag{21}$$

$$|\mathcal{GR}_{s_{t+k}}| = \frac{1}{\alpha^{kN}}|\mathcal{GR}_{s_t}| \tag{22}$$

Proof. See electronic appendix. □

In the above analysis it is important to remember that $\frac{1}{\alpha^{kN}}|\mathcal{GR}_{s_t}| < 1$, otherwise $\mathcal{GR}_{s_t}$ would be larger than $\mathcal{C}(s_t^g, s_t^e)$, which is impossible by definition.

Lemma 1 is of quite general validity, as long as $\mathcal{GR}_{s_t}$ is small respect to $\mathcal{C}(s_t^g, s_t^e)$, and located relatively far from the borders of $\mathcal{C}(s_t^g, s_t^e)$. Even when a portion of $\mathcal{GR}_{s_t}$ is close to the border of $\mathcal{C}(s_t^g, s_t^e)$, neighbourhood shrinking reduces mostly the largest area ($\mathcal{LR}_{s_t}$), and $\mathcal{GR}_{s_{t+1}} \simeq \mathcal{GR}_{s_t}$. Equation (22) shows that the relative coverage of $\mathcal{GR}_{s_t}$, and hence the likelihood of sampling a fitter solution than s_t^g, grows ($1/\alpha > 1$) exponentially. Neighbourhood shrinking is therefore a powerful heuristics to foster progress in the the local search procedure. Neighbourhood shrinking introduces also a trade-off between reducing the reach of local search, and hence slowing down the convergence to the local optimum (see eq. 9), and making local search progress more likely, thus avoiding several cycles of stalling. The probability of a complete stalling of local search (i.e. site abandonment) can be calculated from eq. (22) as follows:

Proposition 6 (Stalling Probability With Neighbourhood Shrinking and Constant $\mathcal{GR}_{s_t}$)**.** *Let s_t^g be the centre of site s at cycle t in the N-dimensional solution space. The probability that local search stagnates for k cycles if $\mathcal{GR}_{s_t}$ is not changed by neighbourhood shrinking is:*

$$P(s_t^g = s_{t+k}^g) = \prod_{h=1}^{k} \left(1 - \frac{1}{\alpha^{hN}} |\mathcal{GR}_{s_t}|\right)^{nr} \tag{23}$$

Proof. After h cycles of stalling, the probability $P_{nr=1}(s_h^g = s_{h+1}^g)$ of not sampling a single solution fitter than s_t^g in $\mathcal{C}(s_t^g, s_{t+h}^e)$ is determined by the relative coverage of $\mathcal{LR}_{s_{t+h}}$, which is defined in eq. (21):

$$P_{nr=1}(s_h^g = s_{h+1}^g) = |\mathcal{LR}_{s_{t+h}}| = 1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}| \tag{24}$$

The probability of stalling at any cycle h is equal to the probability of not picking a fitter solution than s_t^g in nr independent samples of $\mathcal{C}(s_t^g, s_{t+h}^e)$:

$$P(s_h^g = s_{h+1}^g) = P_{nr=1}(s_h^g = s_{h+1}^g)^{nr} \tag{25}$$

The probability of k consecutive cycles of stalling is calculated from eqs. (24) and (25):

$$\begin{aligned} P(s_t^g = s_{t+k}^g) &= \prod_{h=1}^{k-1} P(s_h^g = s_{h+1}^g) = \\ &= \prod_{h=1}^{k-1} \left(1 - \frac{1}{\alpha^{hN}} |\mathcal{GR}_{s_t}|\right)^{nr} \end{aligned} \tag{26}$$

□

This result is valid as long as the number of recruited bees is constant for the k cycles monitored. If the number of bees changes at every iteration, for example as in Packianather et al [13], nr in eq. (23) should be replaced by a variable number nr_k.

The stalling probability can never be 0, since $\mathcal{LR}_{s_t} \neq \emptyset$ for any s_t. It should also be noted that the results of propositions 5 and 6 are independent of the neighbourhood shape. The implications of using hyperspherical instead of hypercubic neighbourhoods will be discussed in Section 7.

6.4. A large stlim or nr?

Proposition 6 is important to understand the behaviour of the Bees Algorithm when neighbourhood shrinking does not change $\mathcal{GR}_{s_t}$, or at least does not change it significantly. As discussed in Section 6.3, this occurrence is most likely when neighbourhood search is near the local optimum, that is, $\mathcal{GR}_{s_t}$ is small and near the centre of $\mathcal{C}$. In this case, the probability that local search stagnates is large ($|\mathcal{GR}_{s_t}|$ is small), and the site may be abandoned after *stlim* cycles of stalling before the local optimum is found (local search stalls).

The probability that local search stalls depends on the number *nr* of solutions that are sampled in one local search cycle, the stalling limit *stlim*, and the size of the search scope. The larger *nr* and *stlim* are, the more likely is to pick at least one solution within $\mathcal{GR}_{s_t}$, and thus the smaller is the likelihood that local search stalls. However, the effect of *nr* and *stlim* on the stalling probability is not the same, due to the nonlinear reduction of the search scope by neighbourhood shrinking. Given a fixed number of sampling opportunities (equal to $nr \cdot stlim$), the question is whether it is more beneficial to sample thoroughly $\mathcal{C}$ for lesser iterations (large *nr*), or sample less intensely $\mathcal{C}$ for longer times (large *stlim*).

In this section, it is assumed that *nr* and *stlim* can be increased by an integer factor $q > 1$, and the local search stalling probability will be indicated as $P_{nr}(s_t = s_{t+stlim})$, where the index *nr* accounts for the number of candidate solutions sampled in $\mathcal{C}$ in one local search cycle.

Lemma 2. *Let s_t^g be the centre of site s at cycle t in the N-dimensional solution space. Assuming that $\mathcal{GR}_{s_t}$ is not changed by neighbourhood shrinking, an increase in the stalling limit by an integer factor $q > 1$ modifies the stalling probability of local search as follows:*

$$P_{nr}(s_t^g = s_{t+q\cdot stlim}^g) = P_{nr}(s_t^g = s_{t+stlim}^g) \cdot T \tag{27}$$

where T as:

$$T = \prod_{k=stlim+1}^{q\cdot stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} \tag{28}$$

Proof. From Proposition 6, $P_{nr}(s_t^g = s_{t+q\cdot stlim}^g)$ is equals to:

$$\begin{aligned} P_{nr}(s_t^g = s_{t+q\cdot stlim}^g) = \prod_{k=1}^{q\cdot stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} = \\ \prod_{k=1}^{stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} \cdot \prod_{k=stlim+1}^{q\cdot stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} = \\ = P_{nr}(s_t = s_{t+stlim}) \prod_{k=stlim+1}^{q\cdot stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} \end{aligned} \tag{29}$$

□

Lemma 3. *Let s_t be the centre of site s at cycle t in the N-dimensional solution space. Assuming that $\mathcal{GR}_{s_t}$ is not changed by neighbourhood shrinking, an increase in the number of foragers by an integer factor $q > 1$ modifies the stalling probability of local search as follows:*

$$P_{q \cdot nr}(s_t^g = s_{t+stlim}^g) = P_{nr}(s_t^g = s_{t+stlim}^g)^q \tag{30}$$

Proof. The proof is straightforward:

$$\begin{aligned} P_{q \cdot nr}(s_t^g = s_{t+stlim}^g) &= \prod_{k=1}^{stlim} \left(1 - \beta^k |\mathcal{GR}_{s_t}|\right)^{q \cdot nr} = \\ = \left(\prod_{k=1}^{stlim} \left(1 - \beta^k |\mathcal{GR}_{s_t}|\right)^{nr} \right)^q &= P_{nr}(s_t^g = s_{t+stlim}^g)^q \end{aligned} \tag{31}$$

□

The next remark will prove that if $\mathcal{GR}_{s_t}$ is not changed by neighbourhood shrinking, increasing the stalling limit by an integer factor $q > 1$ has more effect on decreasing the stalling probability than increasing the number of foragers by the same factor.

Proposition 7 (*stlim* vs. *nr*)**.** *Let s_t be the centre of site s at cycle t in the N-dimensional solution space. Assuming that $\mathcal{GR}_{s_t}$ is not changed by neighbourhood shrinking, an increase in the stalling limit of an integer factor $q > 1$ reduces the stalling probability more than an equal increase in the number of foragers.*

$$P_{nr}(s_t = s_{t+q \cdot stlim}) < P_{q \cdot nr}(s_t = s_{t+stlim}) \tag{32}$$

Proof. See electronic appendix. □

Proposition 7 can also be proven considering a fixed number of available sampling opportunities $T = (q \cdot nr) \cdot stlim = nr \cdot (q \cdot stlim)$ of the search scope. If the choice is to increase the number of foragers, $\mathcal{C}$ will be sampled $q \cdot nr$ times for at most *stlim* cycles of stalling before being abandoned. If $\mathcal{GR}_{s_t}$ is unchanged by neighbourhood shrinking, all candidate solutions will be sampled with a stalling probability $\pi_{nr} \geq A = \left(1 - \frac{1}{\alpha^{stlim \cdot N}} |\mathcal{GR}_{s_t}|\right)$ (see eq. (A.32) in the electronic appendix). If instead the choice is to increase the stalling limit, $\mathcal{C}$ will be sampled nr times for $q \cdot stlim$ cycles, and $(q \cdot stlim - stlim) \cdot nr$ of these samples will have a stalling probability $\pi_{stlim} \leq A = \left(1 - \frac{1}{\alpha^{stlim \cdot N}} |\mathcal{GR}_{s_t}|\right)$ (see eq. (A.33) in the electronic appendix).

As long as $\mathcal{GR}_{s_t}$ is not disjoint multimodal, proposition 7 gives the practitioner a useful guideline to parameterize the Bees Algorithm. This case is not uncommon as the scope of the local search has narrowed down on the attraction basin of one peak of performance. If $\mathcal{GR}_{s_t}$ contain several peaks, there is the risk that repeated applications of the neighbourhood shrinking procedure may cut the main peak out of $\mathcal{GR}_{s_t}$. In this latter case, a high *nr* ensures that many sampling attempts are made before the main peak is lost. Unfortunately, the actual fitness landscape is not known, and trial-and-error is usually needed to address the *nr* vs. *stlim* trade-off. However, several empirical studies [5, 6, 7] obtained the best performances over a large set of varied benchmarks using large *stlim* values, suggesting a wide applicability of proposition 7.

7. Local Search Scope Shape

Among the numerous variants of the BA, the shape of the search scope is one of the least researched features in the literature. In the standard formulation of the Bees Algorithm (Section 2), the search scope $\mathcal{C}(s_t^g, s_t^e)$ of a site s at the cycle t, is defined as a hypercube of side s_t^e centred in s_t^g. A new candidate solution $v \in \mathcal{C}(s_t^g, s_t^e)$ is generated uniformly sampling the hypercube $\mathcal{C}(s_t^g, s_t^e)$.

The main limitation of this *hypercubic sampling* is the anisotropic character of the search, which has the shortest extent in the direction of the coordinate axes, and the longest aligned with the diagonals of the $\mathcal{C}(s_t^g, s_t^e)$ hypercube. This anisotropy introduces a bias in the local search.

Moreover, as the dimensionality of the solution space increases, the volume of the $\mathcal{C}(s_t^g, s_t^e)$ hypercube exponentially increases, making the sampling more sparse (*curse of dimensionality*, [38]).

7.1. Isotropic Local Search

An *isotropic* search scope can be implemented using a *hypersphere* (*ball*) $\mathcal{B}$ centred in s_t^g of radius s_t^r. The reader is referred to [39] and [40] for the algorithmic details of how to achieve a uniformly distributed spherical sampling.

Cubic sampling can be replaced by spherical sampling keeping the rest of the Bees Algorithm unchanged. Neighbourhood shrinking in this case shrinks the hypersphere radius instead of the hypercube edge.

However, replacing cubic with spherical sampling does change the properties of the search. For instance, the maximum reach of local search (proposition 2) in one cycle changes from the diagonal of the hypercube $\frac{s_t^e\sqrt{N}}{2}$ to the radius of the hypersphere s_t^r, and does not scale any more with the dimensionality of the search space. Moreover, if cubic sampling is used, the volume of the search scope grows with the number of dimensions:

$$\mathcal{V}(\mathcal{C}(s_t^g, s_t^e)) = (s_t^e)^N \tag{33}$$

whilst the volume of the hypersphere initially grows and then decreases with the number of dimensions [41]:

$$\mathcal{V}(\mathcal{C}(s_t^g, s_t^r)) = V_N \cdot (s_t^r)^N = \frac{\pi^{N/2}}{\Gamma\left(N/2 + 1\right)}(s_t^r)^N \tag{34}$$

where Γ is gamma function. More precisely, keeping the radius s_t^r fixed, the volume increases for the first N^* dimensions, where

$$N^* = \{N \mid D_{N-1} < s_t^r \leq D_N\} \tag{35}$$

Parameter	N	nr	s_t^e	s_t^{ttl}	s_t^g	$\mathcal{GR}_{s_t}$ center	$\mathcal{GR}_{s_t}$ radius
Value	4	15	10	8	$[1, 0, 0, 0]$	$[0, 0, 0, 0]$	1

Table 1: Parameters setting used in the tests.

with

$$D_N = \frac{\Gamma\left(N/2 + 3/2\right)}{\sqrt{\pi}\Gamma\left(N/2 + 1\right)} \tag{36}$$

and sharply decreases afterwards, approaching zero for large N values. As mentioned in Section 6, replacing cubic with spherical sampling does not alter the validity of propositions 5 and 6.

Proposition 8 (Scope Variation Invariance)**.** *Let $\mathcal{C}(s_t^g, s_t^e)$ and $\mathcal{B}(s_t^g, s^r)$ be the local search scope using respectively cubic and spherical sampling, and $|\mathcal{GR}_{s_t}|^C$ and $|\mathcal{GR}_{s_t}|^S$ be the relative coverage of the $\mathcal{GR}_{s_t}$ region using respectively cubic and spherical sampling. If neighbourhood shrinking does not change the $\mathcal{GR}_{s_t}$ region, shrinking the edge/radius of the search scope of a factor α leads to the same change in the respective coverages:*

$$\begin{aligned} \mathcal{C}(s_t^g, \alpha s_t^e) &\Rightarrow \frac{1}{\alpha^N}|\mathcal{GR}_{s_t}|^C \\ \mathcal{B}(s_t^g, \alpha s_t^r) &\Rightarrow \frac{1}{\alpha^N}|\mathcal{GR}_{s_t}|^S \end{aligned} \tag{37}$$

Proof. See supplementary material. □

The consequence of proposition 8 is that the stagnation probability is computed in the same way (proposition 6) regardless of the kind of sampling used. However, the different behaviour of the search scope volume in the two cases has important implications for high dimensional spaces.

A possible enhancement of the current algorithm would be to switch the shape of the search scope opportunistically to foster the exploratory (cubic sampling) or exploitative (spherical sampling) goal of local search.

7.2. Stalling Probability: Experimental Verification

To verify and visualise the theoretical predictions of section 6, and how the stalling probability varies with the shape of the neighbourhood, the following experimental tests were carried out. The situation where local search has converged inside a large basin of attraction was mimicked. A four-dimensional fitness landscape of hyperspherical shape was considered. The hyperspherical basin had unitary radius, and was centred in the origin of the Cartesian space. In this landscape, local search was performed using a hypecubic neighbourhood of edge $s_t^e = 10$, time to live $s_t^{ttl} = 8$, and initially centred in $s_t^g = [1, 0, 0, 0]$. Fifteen forager bees were used to search the neighbourhood. The parameters of the example are summarised in table 1.

In this case, the $V(\mathcal{GR}_{s_t}) \ll V(\mathcal{C}(s_t^g, s_t^e))$ region is a hypersphere centred in the origin with unitary radius[3]. As per proposition 4, s_t^g lies on the (open) surface of the hypersphere $\mathcal{GR}_{s_t}$. It can be shown that he following variables take the values:

$$\begin{aligned} |\mathcal{GR}_{s_t}| &\approx 4.9348 \cdot 10^{-4} \\ \mathcal{V}(\mathcal{GR}_{s_t}) &\approx 4.9348 \end{aligned} \qquad \mathcal{V}(\mathcal{C}(s_t^g, s_t^e)) = 10^4 \tag{38}$$

and, by complementarity:

$$\mathcal{V}(\mathcal{LR}_{s_t}) \approx 9995.0652 \qquad |\mathcal{LR}_{s_t}| \approx 0.9995 \tag{39}$$

When no neighbourhood shrinking is used (case 1), the stalling probability is given by proposition 5:

$$P(s_t^g = s_{t+s^{ttl}}^g) = 0.9995^{15\cdot 8} \approx 0.9425 \tag{40}$$

When neighbourhood shrinking is used (case 2, shrinking factor $\alpha = 0.9$), the $\mathcal{GR}_{s_t}$ region is unchanged, therefore its volume is the same. Local search stalls after $s_t^{ttl} = 8$ consecutive cycles of stagnation. For each of these cycles of stagnation, neighbourhood shrinking is applied. The volumes of the initial and final search scope are:

$$\begin{aligned} \mathcal{V}(\mathcal{C}(s_t^g, s_t^e)) &= 10^4 \\ \mathcal{V}(\mathcal{C}(s_{t+s_t^{ttl}}^g, s_{t+s_t^{ttl}}^e)) &= 343.3684 \end{aligned} \tag{41}$$

where $s_{t+s_t^{ttl}}^e \approx 4.3047$. The initial and final relative coverage of the $\mathcal{GR}$ regions are:

$$|\mathcal{GR}_{s_t}| \approx 4.9348 \cdot 10^{-4} \qquad |\mathcal{GR}_{s_{t+s_t^{ttl}}}| \approx 1.4372 \cdot 10^{-2} \tag{42}$$

According to proposition 6 the stalling probability is now equal to:

$$\begin{aligned} P(s_t^g = s_{t+s^{ttl}}^g) &= \prod_{k=1}^{s_t^{ttl}} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} = \\ &= \prod_{k=1}^{8} \left(1 - \frac{1}{0.9^{k\cdot 4}} (4.9348 \cdot 10^{-4})\right)^{15} \approx 0.67144 \end{aligned} \tag{43}$$

When a hyperspherical (isotropic) neighbourhood of radius $s^r = 5$ (equivalent to a cubic sampling with edge $s_t^e = 10$) is used (case 3), the search scope volume $\mathcal{V}(\mathcal{B}(s_t^g, s_t^r))$ and the relative coverage of $\mathcal{GR}_s$ are:

$$\mathcal{V}(\mathcal{B}(s_t^g, s_t^r)) \approx 3084.2514 \qquad \begin{aligned} |\mathcal{GR}_{s_t}| &\approx 1.6 \cdot 10^{-3} \\ |\mathcal{LR}_{s_t}| &\approx 0.9984 \end{aligned} \tag{44}$$

[3]The actual radius of $\mathcal{GR}_{s_t}$ is 1^- since $s_t^g \notin \mathcal{GR}_{s_t}$.

	Sampling Type			
	Cubic		Spheric	
	Predicted	Experimental	Predicted	Experimental
Without NS	0.9425	0.9429	0.8252	0.8249
With NS	0.6714	0.67137	0.2725	0.2716

Table 2: Predicted stalling probability and experimental frequency of stalling events for the four cases described in section 7.2.

If neighbourhood shrinking is not used, the predicted stalling probability is (proposition 5):

$$P(s_t^g = s_{t+s_t^{ttl}}^g) = 0.9984^{15 \cdot 8} \approx 0.8252 \tag{45}$$

If neighbourhood shrinking is performed (case 4), the volumes of the initial and final (after $s_t^{ttl} = 8$ consecutive cycles of stagnation) search scope are:

$$\begin{aligned} \mathcal{V}(\mathcal{B}(s_t^g, s_t^r)) &\approx 3084.2514 \\ \mathcal{V}(\mathcal{B}(s_{t+s_t^{ttl}}^g, s_{t+s_t^{ttl}}^r)) &\approx 105.9034 \end{aligned} \tag{46}$$

and the initial and final $\mathcal{GR}_s$ relative coverages are:

$$|\mathcal{GR}_{s_t}| \approx 1.6 \cdot 10^{-3} \quad |\mathcal{GR}_{s_{t+s^{ttl}}}| \approx 4.6597 \cdot 10^{-2} \tag{47}$$

The predicted stalling probability is (proposition 6):

$$P(s_t = s_{t+s^{ttl}}) = \prod_{k=1}^{8} \left(1 - \frac{1}{0.9^{k \cdot 4}}(1.6 \cdot 10^{-3})\right)^{15} \approx 0.2725 \tag{48}$$

The theoretical predictions were numerically tested, performing 10^6 independent optimisation runs for each of the above four cases. Table 2 compares the predicted and experimental stalling frequency (number of times local search stalled divided by the total number of runs) for the four cases.

The empirical results prove the validity of the theoretical predictions. In particular, it is apparent that neighbourhood shrinking increases the probability of progress in local search, thus reducing the stalling probability. At the same time, the empirical examples show the significance of the consequences associated to the choice of neighbourhood shape. In general, the analysis of this section points out that the standard practice of using cubic sampling should be reevaluated in term of the search bias introduced, and the evolution of the stalling probabilities with repeated iterations of neighbourhood shrinking. In detail, a hypersphere of diameter $2s_t^r$ has a smaller hypervolume than a hypercube of edge $s_t^e = 2s_t^r$, and determines a more exploitative search with higher probability of finding solutions in $\mathcal{GR}_{s_t}$.

The tests for the four cases were also repeated varying the dimensionality of the fitness landscape from 2 to 12. The experimental results are shown in fig. 3 and confirm neighbourhood shrinking and hyperspherical sampling are effective policies against premature stalling.

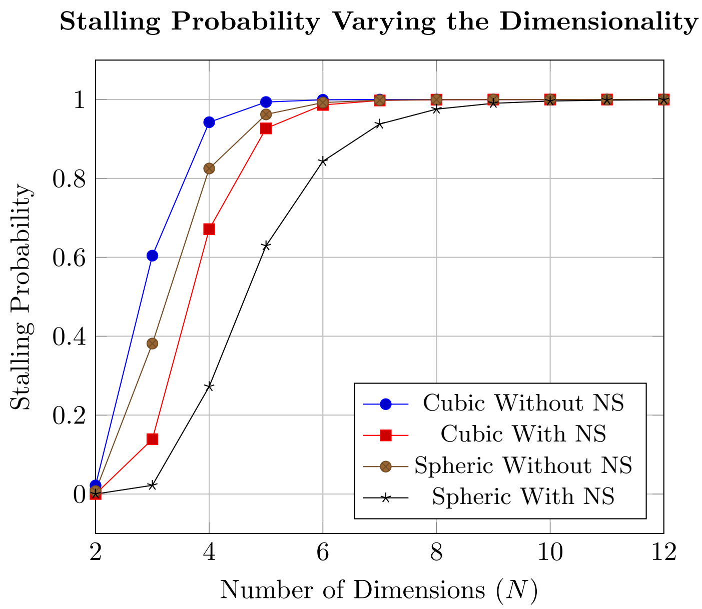

Figure 3: Stalling probability using different sampling methods, with and without the neighbourhood shrinking. All the parameters are kept fixed except the number of dimensions of the problem.

8. Discussion

There is a marked imbalance in the Swarm Intelligence literature, with a prevalence of experimental over theoretical studies. Despite the large success in applications, mathematical analysis of the algorithms is still limited, leaving several open questions on the behaviour, parameterisation, complexity, and nature of the various algorithms. Explanation of the metaheuristics is often limited to the biological metaphor, which may prevent the reader from gaining a full understanding of the search mechanisms [42]. The scarcity of analytical foundations in Swarm Intelligence research has been pointed out by several authors [43, 44, 45].

In this paper, the properties and main features of the Bees Algorithm were formalised and analysed. Despite a number of experimental studies [5, 6, 7, 9] benchmarked the capabilities of the Bees Algorithm, an analytical investigation of its behaviour and operators had never been carried out. The results of the proposed study clarify and support the previous experimental findings, as well as reveal so far overlooked properties. The main findings are summarised below.

The similarities and differences between the Bees Algorithm and standard optimisation methods were discussed. In particular, the Bees Algorithm can be regarded as a parallel version of the LJ Search and VNS methods, where the sampling of the neighbourhood is adaptively allocated (*waggle dance*) according to the fitness of the seed solution. In terms of local search, the main difference with the two aforementioned methods is in the way the neighbourhood is varied: the Bees Algorithm uses neighbourhood shrinking, whilst VNS tries a number of randomly generated shapes, and standard LJ shrinks the neighbourhood regardless of the progress of local search. Also, the Bees Algorithm terminates the local search after $stlim$ stagnation cycles, whilst LJ Search customarily terminates the search after a fixed number of iterations regardless of the progress. In terms of overall metaheuristic, the Bees Algorithm performs several local searches in parallel, adaptively shifting the sampling effort at each generation according to the progress of the search. Neighbourhoods can be abandoned due to lack of progress, or replaced with more promising ones found via global search. For a comparison between the Bees Algorithm and akin swarm optimisation techniques [46, 23] the reader is referred to [5].

The theoretical analysis of the properties of local search showed that the expected step size quickly approaches the maximum value as the number of forager bees is increased. If local search is desired to quickly climb (descend) the fitness slope, a large neighbourhood size is more beneficial than a large number of foragers. Analysis of the stalling probability also found limited benefits in increasing the number of local foragers. That is, neighbourhood shrinking and a large stagnation limit are the most effective policies against premature stagnation of local search. This latter result is in good agreement with the indications of several experimental studies [5, 6, 7], where best performances were obtained using the largest allowed value for the stagnation limit $stlim$.

One of the main contributions of this theoretical analysis regards the shape of the local neighbourhood. For ease of implementation, nearly all versions of the Bees Algorithm used hypercubic local neighbourhoods. As demonstrated, hypercubic sampling biases the search along the directions of the diagonal, and has poor exploitation capabilities in high dimensional spaces due to the curse of dimensionality. That is, the volume of hypercubic neighbourhoods is a power function of the search scope edge s_e. As suggested in section 7, the neighbourhood shape might be varied during the search to switch from explorative (cubic sampling) to exploitative (spherical sampling) search strategies.

9. Conclusions

The Bees Algorithm is a popular optimisation method inspired by the foraging behaviour of honey bees. Despite several experimental investigations, the properties of the Bees Algorithm have never been formally analysed. This paper covers this gap, focusing particularly on the properties of local search.

The main indications are that the local search capabilities of the Bees Algorithm are mainly determined by the size and shape of the neighbourhood, and the number of allowed stagnation cycles. A large neighbourhood enables a quicker progress on the fitness landscape. Conversely, reducing the neighbourhood size helps avoiding premature stagnation of local search. The effect of increasing forager recruitment on the expected search step size (section 5) and stagnation probability (section 6) grows sublinearly with the number of bees.

The shape of the neighbourhood function has been so far largely overlooked in the Bees Algorithm literature. However, it was shown in section 7 that the customary choice of hypercubic sampling creates large neighbourhoods in high-dimensional spaces due to the curse of dimensionality. On the other hand, hyperspherical sampling creates neighbourhoods of sizes that vary according to the gamma function, and tend to become small in high-dimensional spaces (zero for infinitely high-dimensional spaces). Thus, the exploitation capability of local search is highly influenced by the choice of neighbourhood shape.

Overall, the Bees Algorithm can be seen as a parallel adaptive version of the LJ Search and VNS algorithms (section 4), in which the modification of the neighbourhood size and allocation of sampling opportunities are dynamically adjusted according to the fitness of the neighbourhood centres and the local progress of the search. Differently from LJ Search and VNS, the Bees Algorithm also keeps on searching the fitness landscape for new promising neighbourhoods via the global search procedure.

Throughout the paper, the Bees Algorithm was presented in a rigorously mathematical and algorithmic format, beyond the customary qualitative description based on the biological metaphor. It is hoped that this new formalism improves the understanding of the Bees Algorithm, and spurs new analytical studies on its properties, and its similarities to and differences with other Swarm Intelligence metaheuristics.

[1] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, M. Zaidi, The bees algorithm — A novel tool for complex optimisation problems, in: Intelligent Production Machines and Systems, Elsevier, 454–459, 2006.

[2] D. T. Pham, M. Castellani, H. A. Le Thi, Nature-inspired intelligent optimisation using the bees algorithm, in: Transactions on Computational Intelligence XIII, Springer, 38–69, 2014.

[3] D. T. Pham, L. Baronti, B. Zhang, M. Castellani, Optimisation of Engineering Systems With the Bees Algorithm, International Journal of Artificial Life Research (IJALR) 8 (1) (2018) 1–15.

[4] K. Frisch, The role of dances in recruiting bees to familiar sites, Animal Behaviour 16 (4) (1968) 531–533.

[5] D. T. Pham, M. Castellani, The Bees Algorithm: Modelling Foraging Behaviour to Solve Continuous Optimization Problems, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 223 (12) (2009) 2919–2938.

[6] D. T. Pham, M. Castellani, Benchmarking and comparison of nature-inspired population-based continuous optimisation algorithms, Soft Computing 18 (5) (2014) 871–903.

[7] D. T. Pham, M. Castellani, A comparative study of the Bees Algorithm as a tool for function optimisation, Cogent Engineering 2 (1).

[8] A. Auger, B. Doerr, Theory of randomized search heuristics: Foundations and recent developments, vol. 1, World Scientific, 2011.

[9] W. A. Hussein, S. Sahran, S. N. H. S. Abdullah, The variants of the Bees Algorithm (BA): a survey, Artificial Intelligence Review 47 (1) (2017) 67–121.

[10] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE transactions on evolutionary computation 1 (1) (1997) 67–82.

[11] M. Castellani, Q. T. Pham, D. T. Pham, Dynamic optimisation by a modified bees algorithm, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 226 (7) (2012) 956–971.

[12] D. T. Pham, M. Castellani, A. Fahmy, Learning the inverse kinematics of a robot manipulator using the bees algorithm, in: Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on, IEEE, 493–498, 2008.

[13] M. Packianather, M. Landy, D. T. Pham, Enhancing the speed of the Bees Algorithm using Pheromone-based Recruitment, in: Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on, IEEE, 789–794, 2009.

[14] B. Yuce, M. S. Packianather, E. Mastrocinque, D. T. Pham, A. Lambiase, Honey bees inspired optimization method: the bees algorithm, Insects 4 (4) (2013) 646–662.

[15] Q. T. Pham, D. T. Pham, M. Castellani, A modified bees algorithm and a statistics-based method for tuning its parameters, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 226 (3) (2012) 287–301.

[16] A. Ghanbarzadeh, Bees Algorithm: a Novel Optimisation Tool, Ph.D. thesis, Engineering, 2007.

[17] D. Pham, A. H. Darwish, Fuzzy selection of local search sites in the Bees Algorithm, in: Proceedings of the 4th Virtual International Conference on Intelligent Production Machines and Systems, 1–14, 2008.

[18] D. T. Pham, H. A. Darwish, Using the bees algorithm with Kalman filtering to train an artificial neural network for pattern classification, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 224 (7) (2010) 885–892.

[19] A. Imanguliyev, Enhancements for the Bees Algorithm, Ph.D. thesis, Cardiff University, 2013.

[20] S. Ahmad, A study of search neighbourhood in the bees algorithm, Ph.D. thesis, Cardiff University, 2012.

[21] D. T. Pham, E. Koç, Design of a two-dimensional recursive filter using the bees algorithm, International Journal of Automation and Computing 7 (3) (2010) 399–402.

[22] A. Rajasekhar, N. Lynn, S. Das, P. N. Suganthan, Computing with the collective intelligence of honey bees–a survey, Swarm and Evolutionary Computation 32 (2017) 25–48.

[23] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of global optimization 39 (3) (2007) 459–471.

[24] N. Mladenović, P. Hansen, Variable neighborhood search, Computers & operations research 24 (11) (1997) 1097–1100.

[25] R. Luus, T. Jaakola, Optimization by direct search and systematic reduction of the size of search region, AIChE Journal 19 (4) (1973) 760–766.

[26] P. Hansen, N. Mladenović, Variable neighborhood search, in: Handbook of metaheuristics, Springer, 145–184, 2003.

[27] L.-M. Rousseau, M. Gendreau, G. Pesant, Using constraint-based operators with variable neighborhood search to solve the vehicle routing problem with time windows, in: Proceedings of the 1st Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 43–58, 1999.

[28] O. Bräysy, Local search and variable neighborhood search algorithms for the vehicle routing problem with time windows, Vaasan yliopisto, 2001.

[29] R. Whitaker, A fast algorithm for the greedy interchange for large-scale clustering and median location problems, INFOR: Information Systems and Operational Research 21 (2) (1983) 95–108.

[30] N. Mladenović, J. Petrović, V. Kovačević-Vujčić, M. Čangalović, Solving spread spectrum radar polyphase code design problem by tabu search and variable neighbourhood search, European Journal of Operational Research 151 (2) (2003) 389–399.

[31] F. Garcia-López, B. Melián-Batista, J. A. Moreno-Pérez, J. M. Moreno-Vega, The parallel variable neighborhood search for the p-median problem, Journal of Heuristics 8 (3) (2002) 375–388.

[32] A. Djenić, N. Radojičić, M. Marić, M. Mladenović, Parallel VNS for bus terminal location problem, Applied Soft Computing 42 (2016) 448–458.

[33] R. Luus, Optimal control by direct search on feedback gain matrix, Chemical Engineering Science 29 (4) (1974) 1013–1017.

[34] R. Luus, A practical approach to time-optimal control of nonlinear systems, Industrial & Engineering Chemistry Process Design and Development 13 (4) (1974) 405–408.

[35] S. H. Oh, R. Luus, Optimal feedback control of time-delay systems, AIChE Journal 22 (1) (1976) 140–147.

[36] G. G. Nair, Suboptimal control of nonlinear time-delay systems, Journal of Optimization Theory and Applications 29 (1) (1979) 87–99.

[37] G. Gopalakrishnan Nair, On the convergence of the LJ search method, Journal of Optimization Theory and Applications 28 (3) (1979) 429–434.

[38] R. E. Bellman, Adaptive control processes: a guided tour, Princeton university press, 2015.

[39] J. Cook, Rational formulae for the production of a spherically symmetric probability distribution, Mathematics of Computation 11 (58) (1957) 81–82.

[40] M. E. Muller, A note on a method for generating points uniformly on n-dimensional spheres, Communications of the ACM 2 (4) (1959) 19–20.

[41] T. Stibor, J. Timmis, C. Eckert, On the use of hyperspheres in artificial immune systems as antibody recognition regions, in: International Conference on Artificial Immune Systems, Springer, 215–228, 2006.

[42] C. L. Camacho-Villalón, M. Dorigo, T. Stützle, Why the Intelligent Water Drops Cannot Be Considered as a Novel Algorithm, in: International Conference on Swarm Intelligence, Springer, 302–314, 2018.

[43] X.-S. Yang, Swarm-based metaheuristic algorithms and no-free-lunch theorems, in: Theory and New Applications of Swarm Intelligence, InTech, 1–16, 2012.

[44] J. Swan, S. Adriaensen, M. Bishr, E. K. Burke, J. A. Clark, P. De Causmaecker, J. Durillo, K. Hammond, E. Hart, C. G. Johnson, et al., A research agenda for metaheuristic standardization, in: Proceedings of the XI metaheuristics international conference, 7–10, 2015.

[45] A. P. Piotrowski, Across Neighborhood Search algorithm: A comprehensive analysis, Information Sciences 435 (2018) 334–381.

[46] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural networks, IEEE, 1942–1948, 1995.

[47] S. M. Ross, Introduction to probability models, Academic press, 2014.

An Analysis of the Search Mechanisms of the Bees Algorithm (Electronic Appendix)

Luca Baronti, Marco Castellani, and Duc Truong Pham
Department of Mechanical Engineering, University of Birmingham, United Kingdom

A. Theorems Proofs

A.1. Expected Step Size

Proposition 3 (Expected Step Size)**.** *Given a strict monotonic increasing one-dimensional fitness landscape in* $[0, \ell] \in \mathbb{R}$ *(e.g. a straight line), and a site centred in* $s^g_t = \ell/2$ *with edge* $s^e_t = \ell$*, the expected step size of one local search iteration (i.e. the average distance between* s^g_t *and* s^g_{t+1}*) is:*

$$d(s^g_t, s^g_{t+1}) = \ell \frac{0.5^{nr+1} + nr}{nr + 1} - \frac{\ell}{2} \tag{A.1}$$

Proof. The goal is to calculate the expected output of the stochastic local search operator $s^g_{t+1} = L_{nr}(s^g_t)$ defined in eq. (2), with the search scope within $[0, \ell]$. This output can be expressed as the following continuous random variable:

$$\begin{aligned} X &= \arg\max_{x_i} \{\mathscr{F}(s^g_t), \mathscr{F}(x_1), \dots, \mathscr{F}(x_{nr})\} = \\ &= \arg\max_{x_i} \{\phi(x_1), \dots, \phi(x_{nr})\} \end{aligned} \tag{A.2}$$

where:

$$x_i \sim U(0, \ell) \quad \text{and} \quad \phi(x_i) = \max\{\mathscr{F}(s^g_t), \mathscr{F}(x_i)\} \tag{A.3}$$

The expected value E of a continuous random variable Y defined in the interval $[a, b]$ is computed as [47]:

$$E[Y] = \int_a^b y \cdot PDF_Y(y) dy \tag{A.4}$$

where $PDF_Y(y)$ is the probability density function of Y. The probability density function of a random variable Y is equal to the derivative of the cumulative distribution function $CDF_Y(y)$. In the case of the variable X defined in eq. (A.2):

$$CDF_X(x) = \prod_{i=1}^{nr} P(\phi(x_i) \leq x) = CDF_{\phi(x)}(x)^{nr} \tag{A.5}$$

where $P(\phi(x_i) \leq x)$ is the probability that one random sample x_i of the search scope is less or equal to x. Differentiating $CDF_X(x)$ and plugging the derivative into eq. (A.4):

$$E[X] = \int_0^l \phi(x) \cdot nr \cdot CDF_{\phi(x)}(x)^{nr-1} PDF_{\phi(x)}(x) dx \tag{A.6}$$

Using the *Law of the Unconscious Statistician* [47] is possible to make the following substitution:

$$\begin{aligned} E[X] &= \int_0^\ell \phi(x) \cdot nr \cdot CDF_{\phi(x)}(x)^{nr-1} PDF_{\phi(x)}(x) dx \\ &= \int_0^\ell \phi(x) \cdot nr \cdot CDF_x(x)^{nr-1} PDF_x(x) dx \end{aligned} \tag{A.7}$$

The cumulative distribution function and the probability density function of a random variable X sampled with uniform probability in $U(0, \ell)$ are:

$$CDF_U(x) = \begin{cases} 0 & x < 0 \\ \frac{x}{\ell} & 0 \leq x < \ell \\ 1 & y \geq \ell \end{cases} \tag{A.8}$$

and

$$PDF_U(x) = \begin{cases} \frac{1}{\ell} & 0 \leq x \leq \ell \\ 0 & \text{elsewhere} \end{cases} \tag{A.9}$$

From eq. (A.2) it is known that $x \sim U(0, \ell)$, also $\mathscr{F}$ is assumed to be monotonic[4], therefore:

$$\begin{aligned} E[X] &= \int_0^\ell \max\{x, s_t^g\} \cdot nr \cdot CDF_U(x)^{nr-1} PDF_U(x) dx = \\ &= \int_0^{\ell/2} {\ell/2} \cdot nr \cdot \left(\frac{x}{\ell}\right)^{nr-1} \frac{1}{\ell} dx + \int_{\ell/2}^\ell x \cdot nr \cdot \left(\frac{x}{\ell}\right)^{nr-1} \frac{1}{\ell} dx = \\ &= \left. \frac{nr \cdot x^{nr}}{2nr(\ell^{nr-1})} \right|_0^{\ell/2} + \left. \frac{nr \cdot x^{nr+1}}{\ell^{nr}(nr+1)} \right|_{\ell/2}^{l} = \\ &= 0.5^{nr+1}\ell + \frac{nr \cdot \ell(1 - 0.5^{nr+1})}{nr+1} = \\ &= \frac{nr 0.5^{nr+1}\ell + 0.5^{nr+1}\ell + nr \cdot \ell - nr 0.5^{nr+1}\ell}{nr+1} = \\ &= \ell \frac{0.5^{nr+1} + nr}{nr+1} \end{aligned} \tag{A.10}$$

[4]In the proof the case of monotonic increasing fitness is considered. This is not a loss of generality since only the expected step size is considered, not the direction.

Therefore the average step size is:

$$d(s_t^g, s_{t+1}^g) = |s_{t+1}^g - s_t^g| = \ell \frac{0.5^{nr+1} + nr}{nr + 1} - \frac{\ell}{2} \tag{A.11}$$

□

A.2. Coverage Reduction with constant $\mathcal{GR}_{s_t}$

Lemma 1 (Coverage Reduction with constant $\mathcal{GR}_{s_t}$)**.** *Let s_t^g be the centre of site s at cycle t in the N-dimensional solution space. If local search stagnates for k cycles ($s_t^g = s_{t+k}^g$), and the region $\mathcal{GR}_{s_t} \neq \emptyset$ is not changed by neighbourhood shrinking, the relative coverages of $\mathcal{LR}_{s_{t+k}}$ and $\mathcal{GR}_{s_{t+k}}$ become:*

$$|\mathcal{LR}_{s_{t+k}}| = \frac{1}{\alpha^{kN}}(|\mathcal{LR}_{s_t}| - 1) + 1 = 1 - \frac{1}{\alpha^{kN}}|\mathcal{GR}_{s_t}| \tag{A.12}$$

$$|\mathcal{GR}_{s_{t+k}}| = \frac{1}{\alpha^{kN}}|\mathcal{GR}_{s_t}| \tag{A.13}$$

Proof. Since $\mathcal{GR}_{s_t}$ is not changed, $\mathcal{GR}_{s_t} = \mathcal{GR}_{s_{t+j}}$ and $\mathcal{V}(\mathcal{GR}_{s_t}) = \mathcal{V}(\mathcal{GR}_{s_{t+j}})$ $\forall j = 1, \ldots, k$. Also, remembering eq. (13):

$$\mathcal{LR}_{s_t} \cup \mathcal{GR}_{s_t} = \mathcal{C}(s_t^g, s_t^e) \Rightarrow \mathcal{V}(\mathcal{LR}_{s_t}) + \mathcal{V}(\mathcal{GR}_{s_t}) = \mathcal{V}(\mathcal{C}(s_t^g, s_t^e)) \tag{A.14}$$

However, neighbourhood shrinking reduces the local search edge of a factor α (Section 2 and eq. 5). That is, $s_{t+1}^e = \alpha s_t^e$ and after k successive repetitions of neighbourhood shrinking $s_{t+k}^e = \alpha^k s_t^e$. The volume of the search scope is reduced accordingly:

$$\mathcal{V}(\mathcal{C}(s_{t+k}^g, s_{t+k}^e)) = \mathcal{V}(\mathcal{C}(s_t^g, \alpha^k s_t^e)) = \alpha^{kN}\mathcal{V}(\mathcal{C}(s_t^g, s_t^e)) \tag{A.15}$$

Since $\mathcal{GR}_{s_t}$ is not reduced, the reduction $\mathcal{LR}_{s_t}$ will be equal to the reduction of $\mathcal{C}(s_t^g, s_t^e))$. That is:

$$\mathcal{V}(\mathcal{LR}_{s_{t+k}}) = \mathcal{V}(\mathcal{LR}_{s_t}) - \big(\mathcal{V}(\mathcal{C}(s_t^g, s_t^e)) - \mathcal{V}(\mathcal{C}(s_{t+k}^g, s_{t+k}^e))\big) \tag{A.16}$$

From the definition of the relative coverage (eqs. (15) and (A.15)):

$$|\mathcal{LR}_{s_{t+k}}| = \frac{\mathcal{V}(\mathcal{LR}_{s_{t+k}})}{\mathcal{V}(\mathcal{C}(s_{t+k}^g, s_{t+k}^e))} = \frac{\mathcal{V}(\mathcal{LR}_{s_{t+k}})}{\alpha^{kN}\mathcal{V}(\mathcal{C}(s_t^g, s_t^e))} \tag{A.17}$$

And from eq. (A.16), $\frac{\mathcal{V}(\mathcal{LR}_{s_{t+k}})}{\alpha^{kN}\mathcal{V}(\mathcal{C}(s_t^g, s_t^e))}$ is equals to:

$$\begin{aligned}
&\frac{1}{\alpha^{kN}} \cdot \frac{\mathcal{V}(\mathcal{LR}_{s_t}) - (\mathcal{V}(\mathcal{C}(s_t^g, s_t^e)) - \mathcal{V}(\mathcal{C}(s_{t+k}^g, s_{t+k}^e)))}{\mathcal{V}(\mathcal{C}(s_t^g, s_t^e))} = \\
&= \frac{1}{\alpha^{kN}} \cdot \frac{\mathcal{V}(\mathcal{LR}_{s_t}) - (\mathcal{V}(\mathcal{C}(s_t^g, s_t^e)) - \alpha^{kN}\mathcal{V}(\mathcal{C}(s_t^g, s_t^e)))}{\mathcal{V}(\mathcal{C}(s_t^g, s_t^e))} = \\
&= \frac{1}{\alpha^{kN}} \cdot \frac{\mathcal{V}(\mathcal{LR}_{s_t})}{\mathcal{V}(\mathcal{C}(s_t^g, s_t^e))} - \frac{(1 - \alpha^{kN})}{\alpha^{kN}} \cdot \frac{\mathcal{V}(\mathcal{C}(s_t^g, s_t^e))}{\mathcal{V}(\mathcal{C}(s_t^g, s_t^e))} = \\
&= \frac{1}{\alpha^{kN}} \cdot |\mathcal{LR}_{s_t}| - \frac{(1 - \alpha^{kN})}{\alpha^{kN}}
\end{aligned} \tag{A.18}$$

Equation (21) is obtained rearranging the final line of eq. (A.18). Remembering eq. (16), it is also straightforward to show that:

$$\begin{aligned}|\mathcal{LR}_{s_{t+k}}| &= \frac{1}{\alpha^{kN}}(|\mathcal{LR}_{s_t}| - 1) + 1 = \\ &= \frac{1}{\alpha^{kN}}(1 - |\mathcal{GR}_{s_t}| - 1) + 1 = \\ &= 1 - \frac{1}{\alpha^{kN}}|\mathcal{GR}_{s_t}|\end{aligned} \tag{A.19}$$

Finally, eq. (22) is obtained from eq. (21) and eq. (16)

$$\begin{aligned}|\mathcal{GR}_{s_{t+k}}| &= 1 - |\mathcal{LR}_{s_{t+k}}| = \\ &= 1 - \left(1 - \frac{1}{\alpha^{kN}}|\mathcal{GR}_{s_t}|\right) = \frac{1}{\alpha^{kN}}|\mathcal{GR}_{s_t}|\end{aligned} \tag{A.20}$$

□

A.3. Stalling Probability Without Shrinking

Proposition 5 (Stalling Probability Without Shrinking)**.** *Given site s centred on s_t^g at cycle t, the probability that local search without neighbourhood shrinking stalls is:*

$$P(s_t^g = s^g_{t+s_t^{ttl}}) = |\mathcal{LR}_{s_t}|^{nr \cdot s_t^{ttl}} \tag{A.21}$$

Proof. The site stalls if the search stagnates for the next s_t^{ttl} cycles, that is, if all the nr candidate solutions generated during s_t^{ttl} local search cycles lie in $\mathcal{LR}_{s_k}$. When local search stagnates, the centre of the site is unchanged, and if the search scope is not changed (no neighbourhood shrinking), $|\mathcal{LR}_{s_t}|$ is constant:

$$|\mathcal{LR}_{s_t}| = |\mathcal{LR}_{s_k}| \qquad \forall k \in \mathbb{N} \mid t \leq k < t + s_t^{ttl} \tag{A.22}$$

The joint probability that all the solutions sampled during one given cycle k of local search belong to $\mathcal{LR}_{s_k}$ is indicated as:

$$P(v \in \mathcal{LR}_{s_k}) \qquad \forall v \in \{v_1, \ldots, v_{nr}\} \quad v_i \sim \mathcal{C}(s_k^g, s_k^e) \tag{A.23}$$

Due to the uniform sampling, the probability that one solution is picked from $\mathcal{LR}_{s_k}$ corresponds to $|\mathcal{LR}_{s_k}|$. Remembering eq. (A.22), it follows that:

$$P(v \in \mathcal{LR}_{s_k}) = |\mathcal{LR}_{s_k}|^{nr} = |\mathcal{LR}_{s_t}|^{nr} \tag{A.24}$$

The stalling probability can then be computed as the joint probability of s_t^{ttl} consecutive stagnations of local search cycles:

$$P(s_t = s_{t+s_t^{ttl}}) = \prod_{k=t}^{t+s_t^{ttl}-1} |\mathcal{LR}_{s_k}|^{nr} = |\mathcal{LR}_{s_t}|^{nr \cdot s_t^{ttl}} \tag{A.25}$$

□

A.4. stlim vs. nr

Proposition 7 (*stlim* vs. *nr*)**.** *Let s_t be the centre of site s at cycle t in the N-dimensional solution space. Assuming that $\mathcal{GR}_{s_t}$ is not changed by neighbourhood shrinking, an increase in the stalling limit of an integer factor $q > 1$ reduces the stalling probability more than an equal increase in the number of foragers.*

$$P_{nr}(s_t = s_{t+q \cdot stlim}) < P_{q \cdot nr}(s_t = s_{t+stlim}) \tag{A.26}$$

Proof. Remembering lemmas 2 and 3, eq. (32) can be re-written as:

$$A < P_{nr}(s_t = s_{t+stlim})^q \tag{A.27}$$

with

$$A = P_{nr}(s_t = s_{t+stlim}) \prod_{k=stlim+1}^{q \cdot stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} \tag{A.28}$$

with $P_{nr}(s_t = s_{t+stlim})$ a non-null probability, and hence a positive real number. Equation (A.27) can thus be rewritten as:

$$\prod_{k=stlim+1}^{q \cdot stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} < P_{nr}(s_t = s_{t+stlim})^{q-1} \tag{A.29}$$

Remembering proposition 6:

$$P_{nr}(s_t = s_{t+stlim})^{q-1} = \left(\prod_{k=1}^{stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr}\right)^{q-1} \tag{A.30}$$

Equation (A.29) becomes:

$$\prod_{k=stlim+1}^{q \cdot stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} < \prod_{k=1}^{stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{(q-1) \cdot nr} \tag{A.31}$$

The two terms inside the brackets on the right and left hand sides of A.31 express the relative coverage of $\mathcal{LR}_{s_t}$ at time k. That is, they represent the probability of picking a solution of lower fitness than s_t inside $\mathcal{C}$ at time k. They become smaller as k increases ($\alpha < 1$), and thus:

$$1 - \frac{1}{\alpha^{stlim \cdot N}} |\mathcal{GR}_{s_t}| \leq 1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}| \tag{A.32}$$

for all $k \in \{1, stlim\}$. Likewise:

$$1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}| \leq 1 - \frac{1}{\alpha^{stlim \cdot N}} |\mathcal{GR}_{s_t}| \tag{A.33}$$

for all $k \in \{stlim + 1, q \cdot stlim\}$. Accordingly:

$$X < Y \qquad \text{and} \qquad W < Z \tag{A.34}$$

with

$$
\begin{aligned}
X &= \prod_{k=1}^{stlim} \left(1 - \frac{1}{\alpha^{stlim \cdot N}} |\mathcal{GR}_{s_t}|\right)^{(q-1)\cdot nr} \\
Y &= \prod_{k=1}^{stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{(q-1)\cdot nr} \\
W &= \prod_{k=stlim+1}^{q\cdot stlim} \left(1 - \frac{1}{\alpha^{kN}} |\mathcal{GR}_{s_t}|\right)^{nr} \\
Z &= \prod_{k=stlim+1}^{q\cdot stlim} \left(1 - \frac{1}{\alpha^{stlim \cdot N}} |\mathcal{GR}_{s_t}|\right)^{nr}
\end{aligned}
\tag{A.35}
$$

Equation (A.31) ($W < Y$) is certainly true if $W < Z \leq X < Y$. Setting $A = \left(1 - \frac{1}{\alpha^{stlim \cdot N}} |\mathcal{GR}_{s_t}|\right)$, $Z \leq X$ can be rewritten as:

$$
\prod_{k=stlim+1}^{q\cdot stlim} (A)^{nr} \leq \prod_{k=1}^{stlim} (A)^{(q-1)\cdot nr} \tag{A.36}
$$

Developing the sequence of products:

$$
(A)^{nr \cdot q \cdot stlim} \leq (A)^{(q-1)\cdot nr \cdot stlim} \tag{A.37}
$$

Given that A is a stalling probability, and hence $0 < A \leq 1$, the inequality eq. (A.37) is true because the left hand side is raised to a higher power than the right hand side of the inequality. □

A.5. Scope Variation Invariance

Proposition 8 (Scope Variation Invariance)**.** *Let $\mathcal{C}(s_t^g, s_t^e)$ and $\mathcal{B}(s_t^g, s^r)$ be the local search scope using respectively cubic and spherical sampling, and $|\mathcal{GR}_{s_t}|^C$ and $|\mathcal{GR}_{s_t}|^S$ be the relative coverage of the $\mathcal{GR}_{s_t}$ region using respectively cubic and spherical sampling. If neighbourhood shrinking does not change the $\mathcal{GR}_{s_t}$ region, shrinking the edge/radius of the search scope of a factor α leads to the same change in the respective coverages:*

$$
\begin{aligned}
\mathcal{C}(s_t^g, \alpha s_t^e) &\Rightarrow \frac{1}{\alpha^N} |\mathcal{GR}_{s_t}|^C \\
\mathcal{B}(s_t^g, \alpha s_t^r) &\Rightarrow \frac{1}{\alpha^N} |\mathcal{GR}_{s_t}|^S
\end{aligned}
\tag{A.38}
$$

Proof. This can be directly proven as follows:

$$
\begin{aligned}
&\frac{\mathcal{V}(\mathcal{GR}_{s_t})}{\mathcal{V}(\mathcal{C}(s_t^g, \alpha s_t^e))} = \frac{\mathcal{V}(\mathcal{GR}_{s_t})}{(\alpha s_t^e)^N} = \\
&= \frac{1}{\alpha^N} \cdot \frac{\mathcal{V}(\mathcal{GR}_{s_t})}{(s_t^e)^N} = \frac{1}{\alpha^N} \cdot \frac{\mathcal{V}(\mathcal{GR}_{s_t})}{\mathcal{V}(\mathcal{C}(s_t^g, s_t^e))}
\end{aligned}
\tag{A.39}
$$

$$
\begin{aligned}
&\frac{\mathcal{V}(\mathcal{GR}_{s_t})}{\mathcal{V}(\mathcal{B}(s_t^g, \alpha s_t^r))} = \frac{\mathcal{V}(\mathcal{GR}_{s_t})}{V_N \cdot (\alpha s_t^r)^N} = \\
&= \frac{1}{\alpha^N} \cdot \frac{\mathcal{V}(\mathcal{GR}_{s_t})}{V_N \cdot (s_t^r)^N} = \frac{1}{\alpha^N} \cdot \frac{\mathcal{V}(\mathcal{GR}_{s_t})}{\mathcal{V}(\mathcal{B}(s_t^g, s_t^r))}
\end{aligned}
\tag{A.40}
$$

□