Appl Intell (2010) 32: 279–291
DOI 10.1007/s10489-008-0145-8

Search intensity versus search diversity: a false trade off?

Alexandre Linhares · **Horacio Hideki Yanasse**

Published online: 13 September 2008

Abstract An implicit tenet of modern search heuristics is that there is a mutually exclusive balance between two desirable goals: search diversity (or distribution), i.e., search through a maximum number of distinct areas, and, search intensity, i.e., a maximum search exploitation within each specific area. We claim that the hypothesis that these goals are mutually exclusive is false in parallel systems. We argue that it is possible to devise methods that exhibit high search intensity and high search diversity during the whole algorithmic execution. It is considered how distance metrics, i.e., functions for measuring diversity (given by the minimum number of local search steps between two solutions) and coordination policies, i.e., mechanisms for directing and redirecting search processes based on the information acquired by the distance metrics, can be used together to integrate a framework for the development of advanced collective search methods that present such desiderata of search intensity and search diversity under simultaneous coexistence. The presented model also avoids the undesirable occurrence of a problem we refer to as the 'ergometric bike phenomenon'. Finally, this work is one of the very few analysis accomplished on a level of meta-meta-heuristics, because all arguments are independent of specific problems handled (such as scheduling, planning, etc.), of specific solution methods (such as genetic algorithms, simulated annealing, tabu search, etc.) and of specific neighborhood or genetic operators (2-opt, crossover, etc.).

Keywords Modern heuristics · Search methods · Combinatorial optimization · Distance metrics · Coordination policies

1 Introduction: intensity versus diversity

Combinatorial optimization problems have been of central concern in fields such as operations research (i.e., in production planning; or in logistics network design), computer science (i.e., in Very Large Scale of Integration design), and artificial intelligence (i.e., in planning or scheduling problems). One of the most fruitful approaches to industrial-scale massively multimodal combinatorial optimization problems consists of the use of local search heuristics. These methods must exhibit search intensity and search diversity to be effective. The goal of search intensity is to find the best solution contained within a relatively small region, while the goal of search diversity is to sample a large number of different regions, in order to make sure that the search space has been properly explored, and to locate the region(s) containing the global optimum. Without adequate search intensity, a method may pass close to the optimum solution and still be unable to find it. Without adequate search diversity, a method may become deeply absorbed in relatively poor regions of the search space, unable to find higher quality solutions that reside elsewhere.

The prevailing heuristic models do not exhibit these desirable goals co-existing simultaneously. Consider simulated annealing [1–3], a thermodynamically-inspired algorithm

A. Linhares (✉)
EBAPE/FGV, Praia de Botafogo 190, Rio de Janeiro 22257-970, Brazil
e-mail: alexandre.linhares@fgv.br

A. Linhares · H.H. Yanasse
Brazilian Institute of Space Research, LAC/INPE, Av Astronautas 1758, S. J. Campos, SP 12227-010, Brazil

A. Linhares
The Club of Rome, Rämistrasse 18, 8001 Zurich, Switzerland
url: www.clubofrome.org

based on the Gibbs distribution for obtaining configurations (solutions) at selected temperatures. At high temperatures, all configurations are equally probable; while at low (close to zero) temperatures, only the minimum energy (cost) configurations are to be found. Simulated annealing displays both search diversity and search intensity; *but not simultaneously*. The algorithm gradually advances from the pure search diversity found at high temperatures (when many regions are probed and few, if any, good solutions are found) to the pure search intensity found whenever the temperature reaches zero (at which point no unimproving solution can be accepted and thus diversity from then on will be absolute zero). In the process it gradually trades diversity for intensity.

With genetic algorithms [4–7], populations start with a diverse set of individuals taken from a random sampling of the 'population space', and gradually tend towards intensity, as the system converges in the direction of a 'clustered' set of individuals, which are akin to each other, losing, over the course of this process, great amounts of 'genetic diversity'. A well-known way around this problem is the proposal of fitness sharing [8], where two or more individuals that are closer than a predefined distance share their fitness, i.e., "downscaling the fitness of individuals with similar genes" [9, 10], and, therefore, creating a pressure for high population diversity. Unfortunately, fitness sharing may fundamentally downgrade intensity, to the point of, in extreme cases, being close to a random selection [9].

A third strategy, tabu search, uses either long-term memory structures to trigger a mode of search diversity or intermediate-term memory structures to trigger a mode of search intensity [11–13]. Thus, both diversity and intensity are to be expected in these algorithms, but *each one alternates at the expense of the other*. Similar alternations occur in scatter search [14, 15], in the initiation and sampling phases of microcanonical optimization [16, 17], and in the construction and improvement phases of GRASP [18–20]. (Note that even if solutions found in the construction phase of GRASP are very diverse, this does not automatically imply that the solutions found after the improvement phase will preserve such diversity, as solutions may cluster during the improvement phase.)

These classical local search models present subtle hints of a mutually exclusive tradeoff between search intensity and search diversity. A literature review, however, makes such a view explicit, as numerous studies have taken the, sometimes tacit assumption of such a mutually exclusive balance between search diversity and search intensity. In the context of genetic algorithms (applied to optimization), for example, one learns that "*a high crossover probability will encourage steady hill-climbing towards the optimal set of parameter values. On the other hand, a high rate of mutation will result in more searching of the less promising parts of the search space*" [21]. In another method, mechanisms "*designed to increase the pressure for improvement may be at the expense of population diversity. While such strategies can improve the results on small to moderate problems, in large problems they may not allow sufficient exploration of the solution space*" [22]. In other words, as one gets more intensity, one loses diversity.

Within tabu search: "*effective tabu search procedures keep a balance between intensification and diversification, that is, between reinforcing attributes associated with good solutions and driving the search into regions not yet visited*" [23]. Others mention the "*interplay between diversification and intensification*" [24], "*appropriate mixtures of intensification and diversification*" [25], and "*intensification/diversification tradeoffs*" [26].

Similar remarks have appeared in the context of scatter search, where "*…as indicated by its name, the method induces a real willingness for maintaining the collection points as scattered as possible, hence to have a good diversification. However, intensification can also be achieved…*" [15].

Our major claim can be consolidated after the following quote [27]:

> "Two main features *have to be balanced* in constructing heuristic algorithms:
>
> - The degree of *exploitation*, that is, the amount of effort directed to local search in the present region of the search space (if a region is promising, search more thoroughly);
> - The degree of *exploration*, that is, the amount of effort spent to search in distant regions of the space (sometimes choose a solution in a far region and/or accept a worsening one, to gain the possibility of discovering new better solutions).
>
> These *two possibilities are conflicting*: a *good tradeoff* between them is very important and must be carefully tuned in each algorithm." ([27]; emphasis ours)

Given such remarks, we have interest for the following hypothesis:

Claim 1 (The intensity versus diversity hypothesis) *Local search intensity and local search diversity are mutually exclusive*; *that is, metaheuristic algorithms cannot achieve both of these desirable attributes under perfect simultaneous co-existence.*

The main objective of this paper is to present the case that *search intensity and search diversity are not mutually exclusive in parallel systems*, and therefore to show that this hypothesis is remarkably false. Note that our claim is not that having intensity and diversity in simultaneous co-existence should *necessarily* be advantageous in terms of algorithmic performance. At this point it is not known whether such an advanced method would be practical to develop or would outperform existing local search models; however, this, while certainly a relevant (and interesting) empirical question, is beyond the scope of this paper.

In the following sections, we analyze what search diversity is intended to mean, and present a framework for designing new local search models that comprise high search intensity and high search diversity in simultaneous co-existence. These new models are based on the application of novel mechanisms referred to as *coordination policies*, i.e., guiding principles for coordinating the collective search among processes; and also on the use of *distance metrics* between solutions.

2 Distance and diversity metrics

A combinatorial optimization landscape is defined by the triple (Ω, N, Z), where Ω is a discrete and finite set of solutions, $N : \Omega \rightarrow 2^{\Omega}$ is a neighborhood operator, and $Z : \Omega \rightarrow \Re$ is the objective function to be minimized. Reachability under N can be assumed, i.e., for all $x, y \in \Omega$, there exists a sequence $x = s_1, s_2, \ldots, s_d = y$, where $s_{i+1} \in N(s_i)$ for all $1 \leq i < d$. If d is the minimum number of elements in any possible sequence from x to y, then we may say that $d_N(x, y) = d$, that is, the distance from x to y under operator N is d. Unfortunately, terms such as 'areas' or 'regions' of the search space are rather fuzzy notions, as there are no definite boundaries or demarcations in the search space. An area around p may be formally defined as $A(p, S) : \Omega x\{1, 2, \ldots, n\} \rightarrow 2^{\Omega}$, where S is the 'radius' of the area, hence $x \in A(p, S)$ iff $x \in \Omega \wedge d_N(x, p) \leq S$ (for the selected neighborhood operator N). Thus, to refer to two processes sharing 'the same area' is a notion relative to S(the size of the area) and to $d_N(x, p)$.

Surprisingly, these 'distance metrics' have not been used in local search models (the outstanding exceptions being the *bionomic algorithm* [28, 29], the 'diversity-guided evolutionary algorithms', which, once again, use distance metrics to "alternate between phases of exploration and phases of exploitation" [10]), and also memetic algorithms, which may use the Hamming distance to "freeze some genes [...] in order to escape from local attractors" (see [6, 32] and references therein). Distance metrics can reveal important facts about a pair of solutions: they give an idea of the work (and of the time) required to move from one solution to the other; they are correlated with the probability that the second solution will be visited by a process that passed through the first; finally, they also give a precise measure of similarity between any two solutions. In fact, there is a lack of an accepted measure of diversity in most modern heuristic models, and these distance metrics could be one such measure (obviously considering that they are related to a specific landscape—see [30]).

Without a notion of distance, a search process may become shortsighted. For example, it is possible for a search process to gravitate towards a large 'attractor' in the search space, while always keeping a relatively small distance from a particular solution, thus remaining 'anchored' to the local minima of this region for a long time. In this 'ergometric bike' phenomenon, countless movements may be performed while the variation in distance remains negligible. Without a precise measure of distance, it is hard to perceive whether this phenomenon is occurring, or if the process is exploring a broader part of the search space. For example, let us consider a particularly pathological case. Suppose for instance the following configuration:

(i) a 'basic' tabu search algorithm with a recency-based tabu list size of 10 (or other generally used size, such as 7, or 15, etc.);
(ii) a scheduling problem represented by permutations; and
(iii) the neighborhood structure given by the insertion operator, in which a piece (job/object) is deleted from its current position and re-inserted in another position of the permutation.

Let us suppose the algorithm has reached a local minimum, without loss of generality, at the solution $(1, 2, 3, 4, \ldots, n-2, n-1, n)$. Suppose then the following (obviously pathological) move sequence

$(2, 3, 4, \ldots, n-2, n-1, n, 1)$,

$(2, 3, 4, \ldots, n-2, n-1, 1, n)$,

$(2, 3, 4, \ldots, n-2, 1, n-1, n)$,

$(2, 3, 4, \ldots, 1, n-2, n-1, n)$, $\ldots$,

$(2, 3, 4, 1, \ldots, n-2, n-1, n)$,

$(2, 3, 1, 4, \ldots, n-2, n-1, n)$,

$(2, 1, 3, 4, \ldots, n-2, n-1, n)$,

$(1, 3, 4, \ldots, n-2, n-1, n, 2)$,

$(1, 3, 4, \ldots, n-2, n-1, 2, n)$,

$(1, 3, 4, \ldots, n-2, 2, n-1, n)$,

$(1, 3, 4, \ldots, 2, n-2, n-1, n)$, $\ldots$,

$(1, 3, 4, 2, \ldots, n-2, n-1, n)$,

$(1, 3, 2, 4, \ldots, n-2, n-1, n)$, $\ldots$,

$(1, 2, 4, \ldots, n-2, n-1, n, 3)$,

$(1, 2, 4, \ldots, n-2, n-1, 3, n)$,

$(1, 2, 4, \ldots, n-2, 3, n-1, n)$,

$(1, 2, 4, \ldots, 3, n-2, n-1, n), \quad \ldots,$

$(1, 2, 4, 3, \ldots, n-2, n-1, n), \quad \ldots,$

$(1, 2, 3, 4, \ldots, n-2, n, n-1).$

In this example, $(n^2 - n)/2$ movements are made while still preserving a distance of only one movement to the original local minimum! While such a pathological trajectory of radius 1 is extremely unlikely, note that the number of solutions—hence the probability of an event of this nature—grows exponentially on the size of the radius, it clearly demonstrates the importance of using precise distance metrics in local search metaheuristics.

Another important function of distance metrics comes from the recent analysis of the landscape of combinatorial optimization problems. Many researchers have considered the relation involving cost structure and distance, that is, the minimum number of applications of a given neighborhood operator needed to transform a solution into another, between locally optimal solutions. Boese, Kahng and Muddu [31], in studying the traveling salesman problem (TSP) and the graph bisection problem, argue for a *big-valley* structure where locally optimum solutions tend to get increasingly closer to each other as they approach the global optimum. New studies followed with similar analysis for the $n/m/P/C_{\max}$ flowshop and other problems [32–34]. In a related study, Mak and Morton [35] have analyzed the relation between the k-opt and the 2-opt metrics for the TSP.

2.1 What is search diversity?

This "ergometric bike" trajectory demonstrates that distance metrics do not by themselves compute the diversity of subset E. The enumerated solutions, even if numbering on the order of billions, may still be stuck to a small-diameter attractor and exhibit a pathologically small diversity. This example points to the suggestion of the following diversity metric based on the explored trajectory diameter:

$$Diversity(E) = \max_{\substack{\forall y \in E \\ \forall x \in E}} d_N(x, y).$$

This metric has the property of growing as a function of the diameter of E. That is, the farthest that two explored solutions encounter themselves apart, the higher the diversity. The use of this metric can prevent the ergometric bike phenomenon from occurring, if one fixes the recently explored solutions and only accepts diversity-increasing movements. However, despite this favorable property, this initial proposal does not provide a good *diversity* measure for a number of reasons:

(i) first, the maximum distribution obtainable by this function equals the maximum distance between any two solutions in Ω, which makes diversity a function of the size of Ω. For example, in a 100-city TSP under the insertion operator, the maximum distance between a pair of solutions is 99 movements. In a 10 000-city TSP, the maximum distance would be 9999 movements. It is desirable to have a measure of diversity which is *independent of problem size*. Let us label this property as *Requirement* 1 for an adequate diversity metric.

(ii) Another problem with this metric is that it is possible to rapidly reach the maximum diversity measured, simply by creating a single, iterative, distance-increasing trajectory at each step. This is trivial to do with most neighborhood operators (see for instance [37, 38]). In the 10 000 city TSP, it would be possible to achieve the maximum diversity in merely 9999 movements. But of course 9999 solutions can not count for high diversity in a space containing on the order of 10 000! solutions. This leads us to *Requirement* 2 for an adequate metric, as it clearly shows that *merely computing the explored trajectory diameter does not lead to an adequate diversity measure*.

2.2 Requirements for a measure of diversity

Let us look beyond the two initial requirements presented above and formalize the requirements of domain and co-domain of the function:

Requirement 3 The domain of the function should be the set of explored solutions, E. It is necessary, in order to compute the diversity of the explored space, to have as the domain of such computation the whole set of explored solutions.

Requirement 4 The co-domain of the function should be the closed interval [0, 1]. This should enable a precise numerical estimation of diversity which is independent of problem size. It should be also clear that diversity should equal 0 iff $|E| = 0$ and that diversity equals 1 iff $E = \Omega$.

In the next section, we review new metrics for diversity that meet these initial requirements and also point out some additional requirements.

2.3 The structure of diversity

As shown, diversity is not synonymous (or even proportional) to distance trajectory. Let us now look at a proposal that respects the four requirements above. Let us label the following metric as the *pragmatic diversity metric*.

$$Diversity(E) = 1 - \max_{\substack{\forall y \in \Omega / E \\ \forall x \in E}} \frac{d_N(x, y)}{\max_{x*, y* \in \Omega} d_N(x*, y*)}.$$

This formula yields the maximum distance obtainable from an unexplored solution to an explored solution (measuring the maximum diameter of "open unexplored space"), divided by the maximum possible distance between any two candidate solutions (diameter of the search space). The latter measure is of course a constant, and is usually trivial to compute. But the former measure may be NP-hard, as discussed below. Note that the limit cases where $E = \varnothing$ and $E = \Omega$ must be specifically addressed. That is, we should take it by definition that:

$$\text{If } E = \varnothing \quad \text{then} \max_{\substack{\forall y \in \Omega/E \\ \forall x \in E}} d_N(x, y) = \max_{x*, y* \in \Omega} d_N(x*, y*)$$

$$\text{If } E = \Omega \quad \text{then} \max_{\substack{\forall y \in \Omega/E \\ \forall x \in E}} d_N(x, y) = 0.$$

This measure has the four properties entailed above: At the start of execution of the algorithm, $|E| = 0$, and thus diversity $= 0$. If all solutions have been explored, that is, if $E = \Omega$, then diversity $= 1$. It is proportional to the size of the largest "empty space". This is the most notorious insight of the model: *a measure of diversity can only be obtained if we have an idea of the size of the largest unexplored areas.*

Let us consider the computational complexity of this diversity metric. If computation of d is NP-Hard, as it is in the case for the TSP under the widely used 2-opt operator [36], then the diversity problem is also NP-Hard. Fortunately, there are numerous neighboring functions that are efficiently computable. In this case it is uncertain if the diversity metric is in fact an NP-hard problem, which leads us to a first open problem:

Open Problem 1 *Is the proposed pragmatic diversity measure computable in polynomial-time for a polynomial-time neighborhood operator?*

The reader should notice that, though the number of potential pairs of solutions (x, y) can grow exponentially large on the size of the problem—and hence present a *de facto* intractable problem, this does not immediately mean that the measure is NP-hard, because NP-Hardness is a property emanating from the size of the set E of explored solutions, not from the size of a specific solution. For example, in the 100-city TSP, a solution is of size 100 indexes. The exponential growth of the TSP search space is derived from this fact. However, a subset E of explored solutions may be very large and still keep the diversity computation under polynomial time (as a function of the size of E). Thus this remains an open research problem for further study.

The measure presented above attends to the previous requirements. However, it would also be desirable to have another measure of diversity that should be independent of the size of E.

It would be especially exciting to have the following information: what is the maximum diversity that can be obtained if we explore M solutions? If we had such information it might be possible to develop algorithms that could iteratively increase the diversity of the explored solution space. The thought of maximizing diversity at each movement, given a set E, leads us to a *theoretically achievable diversity metric*:

$$Diversity(E) = 1 - \frac{\max_{\substack{\forall y \in \Omega/E \\ \forall x \in E}} d_N(x, y)}{\max_{x*, y* \in \Sigma} d_N(x*, y*), \forall \sum \in 2^{\Omega}, |\sum| = |E|}.$$

This formula yields the maximum distance to the farthest unexplored solution, divided by the maximum possible distance theoretically obtainable in any subset of Ω with size $|E|$.

(i) At start, $|E| = 0$, and thus diversity $= 0$;
(ii) If all solutions have been explored, diversity equals 1;
(iii) Diversity is still proportional to the diameter of the largest "empty space", but now it is also a function of the size of the explored space, such that:
(iv) Diversity is automatically decreased as $|E|$ increases to a new level of potential empty-space possibility. That is, if a new move was executed, or a new generation in a genetic algorithm was construed, and the "maximum empty space" remained constant, while it was possible to have had obtained a greater diversity in a space of such size, then this would be captured in this metric.

Therefore, the use of this metric would enable the development of new algorithms theoretically capable of iteratively maximizing search diversity. There are, however, additional computational complexity concerns. The reader may have noticed that it seems that this problem belongs (at least) to the class of NP-hard problems, given that it demands an exponential number of subsets of Ω to be explored in order to compute the diversity function. Therefore we present the second open problem and the conjecture that this problem is hard to compute.

Open Problem 2 *Is the proposed theoretically achievable diversity measure computable in polynomial-time for a polynomial-time neighborhood operator?*

We conjecture that this is not true. In the next section, we turn attention to a framework for local search models that exhibit simultaneous intensity and diversity, based on the introduction of new control mechanisms termed *coordination policies*.

3 A framework for the simultaneous coexistence of intensity and diversity

To defend the feasibility of employing search intensity and search diversity simultaneously, in the following sections let us consider two assumptions: First, assume that there is a set of concurrent search processes, that each of these processes tends to explore the search space with high intensity. The other postulation assumes that there is a distance metric between solutions that is computable in polynomial-time. Let us analyze the first assumption: as an intensive local search based process—such as those considered in [11] or in [17] approaches an area, it tries, during a predetermined time window, to find the best possible solution contained in that area. This is mostly a trivial assumption: for all practical purposes, these search processes could be simple instantiations of tabu search *designed to explore the search space intensely* [11, 17].

This basic assumption in explicit form is:

Claim 2 *The proposed model exhibits high search intensity during the course of its execution.*

The model may exhibit a remarkably intensive behavior, continuously searching for high quality solutions within restricted regions of the search space. And as it may be assumed that this can occur during the entire execution of the algorithm [11], intensity (under the proposed model) can be preserved during the run (instead of alternating with diversity, as we will see). But this does not provide an automatic guarantee against an ergometric bike trajectory. It is now necessary to show that when we consider collective search models, then search diversity can also be shown to exist *during the whole execution*. In order to show that, consider two proposals: first, let us have a set of processes executing in parallel. Second, let us implement a 'boundary' of access for each process. Thus, while multiple processes conduct the search, the distance between processes is computed, such that an artificial 'boundary' may protect each process's area (in order to guarantee the preservation of search diversity). Thus, in case a particular process attempts to cross another process's boundary, this attempted movement should be rejected. The size of this boundary can obviously be made to vary with the size of the problem (this is discussed in detail below).

The suggested *coordination policies*, i.e., guiding principles to coordinate the collective search (based on the distances measured between search processes), may be classified in two categories: distribution coordination policies and redistribution coordination policies. In the distribution coordination policies, certain points act as *detractors*, maintaining a specific minimum distance between processes, and thus enforcing a high systemwide distribution. These are the policies considered in this paper, but there are also redistribution coordination policies, which have some points acting as *attractors* to other processes, bringing these processes closer to potentially better regions of the search space [38]. Let us consider two distribution coordination policies.

3.1 Uniform non-overlap

The first policy is referred to as the *uniform non-overlap* coordination policy (Fig. 1). This policy designates a specific area to each search process; these areas, surrounding the solutions explored by the processes, are set to a uniform size, and may not be entered by other processes (hence its name). Under this policy, there can be no overlapping between these areas, and thus, there is a *guaranteed distribution* of the processes along the search space—this formal

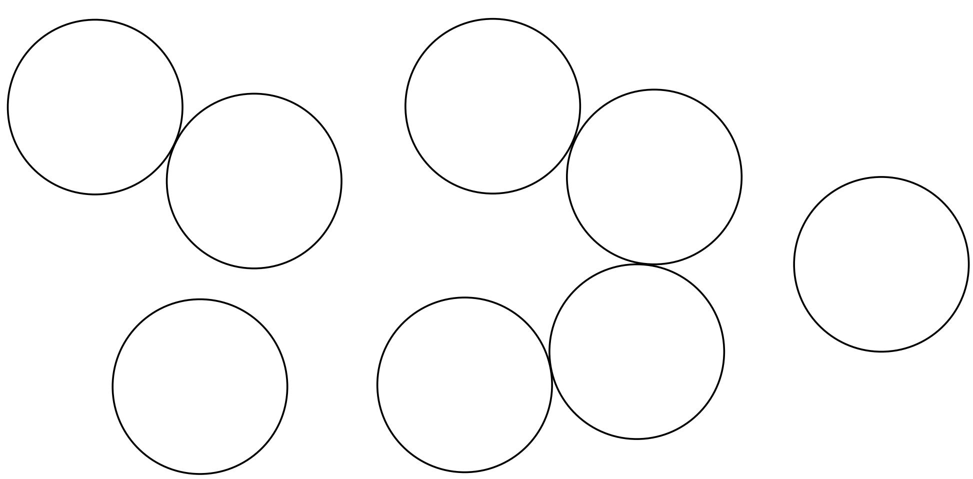

Fig. 1 A schematic look at the proposed framework (under the uniform non-overlap coordination policy): By employing coordination policies and distance metrics, we may enforce processes to maintain a minimum distance between each other (the boundaries of each process are represented by *circles*). These mechanisms not only prevent the wasteful duplication of search effort, but also guarantee high levels of search diversity throughout the execution, as the pressure for maintaining large distances will make the largest empty regions tend to be explored in a relatively small amount of time

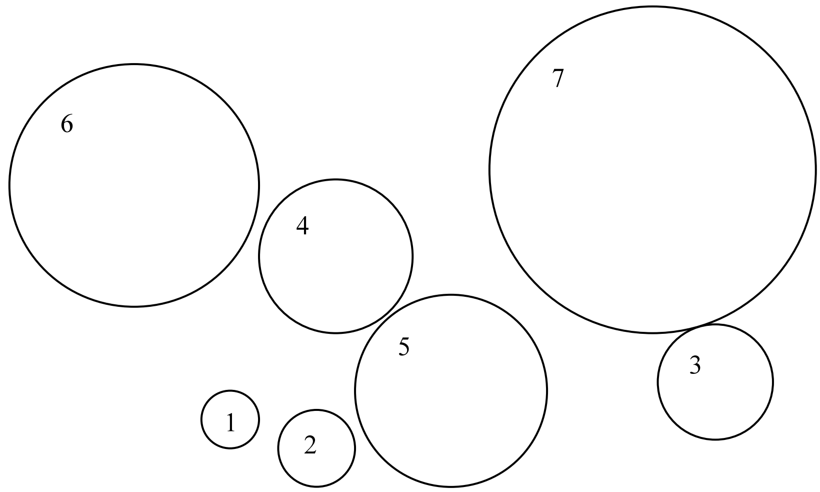

Fig. 2 Under the non-uniform ordered non-overlap with closest-cost distribution policy, the best search processes have increasingly smaller areas, which concentrate the exploration of the search space in proportion to their perceived quality, potentially taking advantage of spaces with big-valley structure

guarantee, it should be mentioned, is not achieved in the classic models of either genetic algorithms—or "genetic algorithm species", as in [39], simulated annealing, or even of tabu search (though the latter may incorporate long-term memory structures devised specifically for search diversification, it cannot provide a formal, mathematical, *guarantee* that the search will be suitably distributed along the search space, as it cannot precisely measure such distribution, for there is simply no function analogous to $D_N(x, p)$—or any measure of diversity for that matter).

Suppose, for instance, that process P_2 attempts a movement that will bring it into the area demarcated by the radius of process P_1. In order to prevent this, the movement can be rejected, or, alternatively, process P_1 can be moved outside of the area corresponding to P_2. In the first case, P_1 acts as detractor, while in the second case P_2 is the detractor. The 'winning' process can be decided either by the quality of the best solutions found by each process, or by the quality of the current solution under exploration. The number of interdependent search processes can obviously be made to scale as the number of available processors grows, enabling thus a greater distribution of the search.

There is, however, a high computational cost associated with computing the distance between all pairs of processes—especially if the distance metrics are not computable in linear time. One way around this is to devise incremental distance metrics, avoiding the entire computation of the distances at each movement. Calculating the distances using only the incremental information associated with each operation could bring down the complexity considerably, and additional gains can even be achieved: to further reduce the computational complexity (for any neighborhood operator selected), one may opt to reduce the number of distance computations from $O(p^2)$ to $O(p)$, where p is the number of processes. This can be done either by sampling pairs of processes in a random manner, or by adopting one of the 'faster' coordination policies presented in [38] which carefully select which pairs of processes will be evaluated.

3.2 Non-uniform distribution policies

The same policy may also be used in a Non-Uniform region size (Fig. 2). That is, the size of the areas allocated to each process may vary according to some specific variable. A candidate for such variable is the cost function obtained by the processes: one may use the order of the processes as a guide for establishing the total size of their areas.

As we have seen, the proposal of non-overlapping coordination policies is intended to obtain a search process that is simultaneously intensive and diverse (therefore showing that these desiderata are not mutually exclusive). A second proposal for coordination policies, whose goal would be to explore the search space more intelligently, may come from the previously mentioned 'big-valley' hypothesis: a correlation between solution quality and distance to the optimum that has been observed for some combinatorial optimization problems [31]. This correlation suggests that areas in which solutions of high quality have been found may be explored with more intensity than those areas in which no solution of high quality has been found. This idea should be explored in landscapes that exhibit a big-valley structure.

The non-uniform ordered non-overlap with closest-cost policy is thus a prime candidate for distributing search processes over the search space. There are a number of reasons for this. The policy holds all the advantages of the uniform ordered non-overlapping policy, with the possibility of increased adaptability if the best processes are allowed smaller areas than the worst processes: The regions of the search space which are perceived to contain higher quality solutions are explored in proportion to this higher perceived quality: a process that has found highest quality solutions holds a small area, and thus enables other processes to explore the vicinity of that area, while the processes occupying the perceived worst areas have correspondingly larger sizes, disabling access to those supposedly inferior quality regions. This policy seems appropriate for landscapes displaying big-valley structure, because the obvious tendency is for processes to cluster around the higher quality areas of the search space, while still preventing the wasted duplication of search effort of having two processes visit the same solution.

From these ideas, the following claim holds:

Claim 3 *The proposed model exhibits high search diversity during the course of its execution.*

By using these coordination policies (and the underlying distance metrics), one is able to stipulate precisely the minimum distance to be kept among processes. This leads to two conclusions: First, diversity would be guaranteed by the calculations of distance between search processes. In this way, we can indirectly control the size of the unexplored search areas, which will maximize diversity. Second, these policies would have the added advantage of preventing the undesirable duplication of search effort (should the areas of distinct processes overlap). Claim 3, together with Claim 2, leads to the following conclusion:

Claim 4 *There are reasons to believe that the intensity versus diversity hypothesis is false: high local search intensity is not mutually exclusive with high local search diversity, and both may be obtained simultaneously.*

In most local search models, we have that, in some moments, intensity is achieved at the expense of diversity; while at other moments, diversity is achieved at the expense of intensity: but that may be a false tradeoff. Though some thinkers have argued that intensity and diversity are "conflicting alternatives", high levels of both may indeed be achieved simultaneously, and such a fact opens a largely unexplored territory for innovations in heuristics research. In the next section we prove that there are polynomial-time metrics readily available for use.

4 Polynomially computable metrics

In this section we provide polynomial time metrics that can be used with permutation operators.

4.1 The 2-exchange operator

While the 2-opt operator is one of the most used for the TSP, other operators are even more popular for different sequencing problems, such as the 2-exchange and the insertion operators used in many scheduling applications. In this section we present a linear time algorithm that transforms π into σ using the minimum number of 2-exchange operations.

The reader should note that, contrary to what its name suggests, this problem is different from the *optimum exchange sorting* problem studied previously [41, p. 198], in which one is concerned with a comparison-exchange tree (which is a data structure used to study the performance of distinct exchange sorting algorithms, such as quicksort, bubblesort, or mergesort). We refer the reader to [41] for details.

The 2-exchange operator switches two positions i and j in a permutation, that is, it transforms permutation $(\pi_1, \ldots, \pi_i, \ldots, \pi_j, \ldots, \pi_n)$ to $(\pi_1, \ldots, \pi_j, \ldots, \pi_i, \ldots, \pi_n)$. Let I denote the identity permutation $(1, 2, \ldots, n)$. We present the following linear time algorithm for computing the minimum 2-exchange distance between π and $\sigma = I$.

```
d = 0;
for i = 1 to n do
    if π_i ≠ σ_i then begin
    2-EXCHANGE(i, σ^{-1}(π_i)); //σ^{-1}(i) denotes the
    position on permutation σ
    d = d + 1;                  // in which element i
     is found
    end;
return d;
```

This algorithm is trivial and it runs in linear time. What is not trivial to prove is that it indeed performs the minimum number of 2-exchange movements between π and $\sigma = I$. To prove the optimality of the algorithm, let us now state the problem in a graph theoretical form referred to as the *switch graph*. Let $G = (V, E)$ be a directed graph with vertex set $V = (v_1, v_2, \ldots, v_n)$ and edge set E. Let each vertex $v_i \in V$ be associated with a position i in the permutation π. The edges of G point from position i of π to the position in σ that element π_i appears. That is, directed edge $v_i v_j \in E$ if and only if $j = \sigma^{-1}(\pi_i)$. In Fig. 3 the switch graph for permutations $\pi = (2, 4, 3, 5, 1, 8, 7, 6)$ and $\sigma = I$ is displayed.

Note that, since each vertex sends and receives a directed edge, the switch graph consists of one or more independent cycles. The size of the cycles ranges from $\{1, 2, \ldots, n\}$, and the number of distinct cycles can be at most n, in the case of n self-cycles (i.e., vertices pointing to itself). Note that this is precisely the desired goal, i.e., to put each element in its correct position. Thus, we are concerned with increasing the number of cycles with the minimum number of operations possible. The problem can now be stated in terms of the switch graph.

Let B be the number of cycles in the switch graph between permutations π and σ, and consider Proposition 1.

Proposition 1 *Given $B < n$, it is always possible to increase the number of cycles by one unit by means of one 2-exchange movement.*

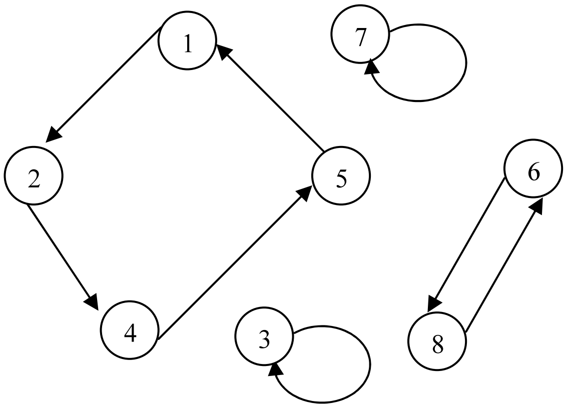

Fig. 3 Switch graph of permutations π and I

Proof Given that $B < n$, it is always possible to break a cycle C_{M+N} of size $M + N$ into two distinct cycles C_M and C_N, for any combination of $M \neq 0$ and $N \neq 0$, in a single exchange operation: select vertices $v_i, v_j \in C_{M+N}$; and, after the 2-exchange operation, the vertices that pointed to v_i and v_j now point to v_j and v_i, respectively. Thus, the cycle C_{M+N} is broken into two distinct cycles C_M and C_N, where M equals the number of edges on the path from v_i to v_j in C_{M+N}, and N equals the number of edges on the path from v_j to v_i in C_{M+N}. □

Proposition 2 *One 2-exchange movement can increase the number of cycles by at most one unit.*

Proof When two elements exchange positions in the permutation, 2 edges are deleted from the graph, and 2 edges are inserted. It is necessary to delete 2 edges in order to break a cycle; hence, each movement increases the number of cycles by at most one unit. □

Let $T(C_S)$ be the minimum number of applications of the 2-exchange operator to transform any S-sized cycle C_S into S self-cycles. Then, we have

Lemma 1 $T(C_S) = S - 1$.

Proof Follows trivially from Propositions 1 and 2. □

Let $D_{2\text{-exchange}}(\pi, \sigma)$ equal the desired 2-exchange distance metric between permutations π and σ. Then, we have

Theorem 2 $D_{2\text{-exchange}}(\pi, \sigma) = n - B$.

Proof It follows from lemma 1 that each S-sized cycle needs $(S - 1)$ movements to be transformed in S self-cycles. The desired distance metric can be directly obtained by the number of S-sized cycles in the switch graph, measured by

$$D_{2\text{-exchange}}(\pi, \sigma) = \sum_{S=1}^{n} (S-1)\alpha_S$$
$$= \sum_{S=1}^{n} S\alpha_S - \sum_{S=1}^{n} \alpha_S = n - B,$$

where α_S is the number of S-sized cycles between permutations π and σ. □

Since the presented algorithm always places one element in its correct position, it always breaks a cycle into two (with the new cycle having size exactly one). This leads to the proof of correctness of the algorithm.

4.2 The insertion operator

The insertion operator for permutations deletes one element from the permutation, and inserts it in a new position: either between two elements, or at the first or last positions, maintaining the order of the other elements intact. Once again, the metric devised here deals with fixed permutations. However, the extension to cyclic problems is readily attainable, as, in such case, it should suffice to maintain one element in a fixed position of the permutation (for example, by keeping element 1 fixed in the first position of the permutation). Note that, despite what the name suggests, the insertion distance metric considered here is not the previously studied number of moves of *sorting by insertion* [41, Sect. 5.2.1], in which the number of moves refers to the overall running time of the algorithms. We refer the reader to [41] for details.

It is not trivial to see that the permutation (8, 5, 6, 9, 4, 2, 1, 7, 3) requires 6 insertions while (1, 8, 5, 2, 6, 3, 9, 4, 7) requires 4. This problem is more general than that considered in [42] for the leading element insertion distance, which is referred to as *head* insertions, and considers only insertions of the very first element of the permutation. They have shown that:

Theorem 3 *The number of insertions required to sort a permutation σ by head insertions is $n - k$, where k is the largest integer such that the last k entries of σ form an increasing subsequence* [42].

In our more general case, to order by insertion the permutation (1, 8, 5, 2, 6, 3, 9, 4, 7), for example, the insight needed is that the subsequence (1, 5, 6, 9) already appears in the correct order, but the remaining elements are in incorrect positions in relation to this subsequence. We refer to such a subsequence as a *correct subsequence*. A *maximal correct subsequence* is one such that no element of the permutation can be added to it while the subsequence remains in correct order. A maximal correct subsequence with the largest size possible (i.e., maximum number of elements) is denoted as a *maximum correct subsequence*. In this example, the subsequence (1, 5, 6, 9) is a maximal correct subsequence, but is not maximum (interested readers should also consider the maximum partial order algorithm [44]). The following lemma enables us to establish a link between the insertion distance and the maximum correct subsequence.

Lemma 2 *If a permutation contains a maximal correct subsequence of size K, it is always possible to order it in $n-K$ insertions.*

Proof Relative to the maximal correct subsequence, there are $n-K$ incorrect positions in the permutation. Each one of these positions can be corrected by inserting it in the right position on the maximal correct subsequence by one application of the insertion operator. Thus, $n-K$ insertions are sufficient to order this permutation.

Thus, given a maximal correct subsequence of size K, $n-K$ insertions are sufficient to order the permutation. (This does not prove, however, that this is the number of necessary insertions to order the permutation.) □

It is important to maximize K, i.e., to find a maximum correct subsequence. A maximum correct subsequence can be obtained in polynomial time, as shown next. We will construct a graph (G, V) that may be traversed as a set of trees, where each tree represents one of the maximal correct subsequences. First, create a set of nodes where each node v_i corresponds to a position i in permutation π. Next, create a set E of edges by placing one directed edge $v_i v_j$ iff (1) $i<j$ and, (2) $\pi_i<\pi_j$ and, (3) $\neg\exists k : i<k<j$ and $\pi_i<\pi_k<\pi_j$. Conditions 1 and 2 enforce that positions i and j belong to a correct subsequence while condition 3 avoids transitivity. Note that G is directed and does not include cycles and may thus be traversed as a set of trees where the nodes that do not receive edges are the roots of the trees. In Fig. 4, the associated traversal-tree for the example permutation $\pi=(1,8,5,2,6,3,9,4,7)$ is shown.

Lemma 3 *The maximum correct subsequence equals the maximum height, H, of all the traversal-trees contained in G.*

Proof Each path from the nodes to the leaves represents a maximal correct subsequence. Therefore, the maximum correct subsequence will be given by the maximum height of the trees contained in G. □

Lemma 4 *It is necessary to perform at least $n-H$ insertions.*

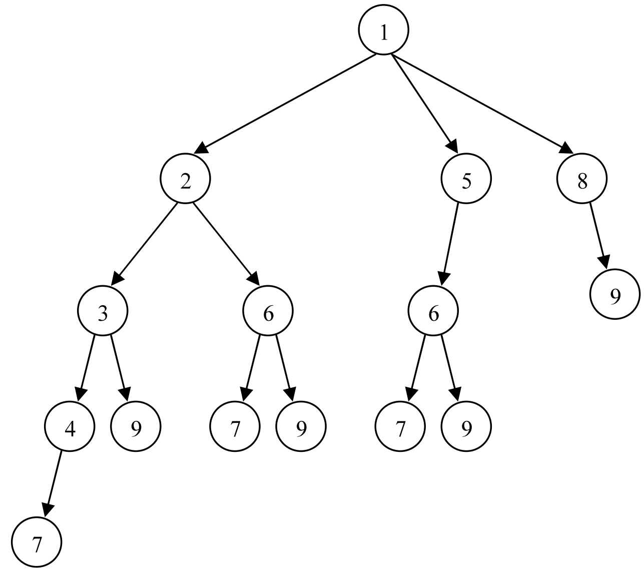

Fig. 4 The traversal-tree of G for $\pi=(1,8,5,2,6,3,9,4,7)$

Proof Consider a maximum correct subsequence. If one element of the permutation is deleted, the size of maximum correct subsequence cannot be larger than the original; in fact, it can only remain either equal to the original or equal to the original minus one. If an element is included in the permutation, the size of the maximum correct subsequence cannot be smaller than the original. It can be equal to the original or equal to the original plus one. An insertion movement is equivalent to these two steps being carried out successively. Hence, an insertion movement can augment the maximum correct subsequence in at most one element. As the maximum correct subsequence has size H, it is necessary to carry out at least $n-H$ insertions to order the permutation. □

Since we know that $n-H$ insertions are sufficient to order the permutation, this leads us to the desired distance metric:

Theorem 4 *The insertion distance of a permutation π of n integers is $n-H$.*

Proof Follows from Lemmas 2, 3 and 4. □

This algorithm enables us to prove the correctness of the desired metric, but it demands $O(n^2)$ time. Fortunately, there is a algorithm available for "determining the longest subsequence in a sequence" [43]—so our concern here is limited to the demonstration of correctness of the metric.

A final comment concerns the distribution of the insertion and the 2-exchange distance metrics. Solutions that are far from each other under one landscape may be close under another landscape. For example, the sequence $(2,3,4,5,6,\dots,n,1)$ requires $(n-1)$ 2-exchanges versus

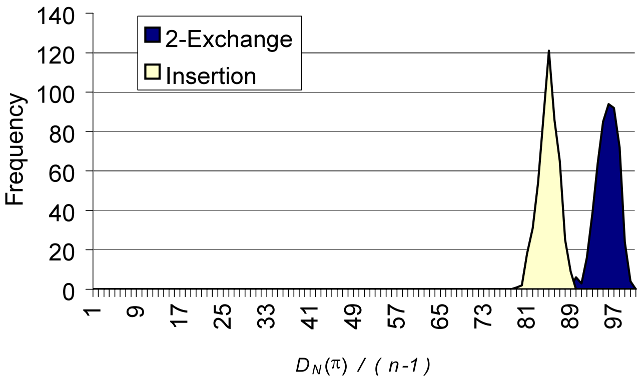

Fig. 5 Distributions of the insertion and the 2-exchange distance metrics

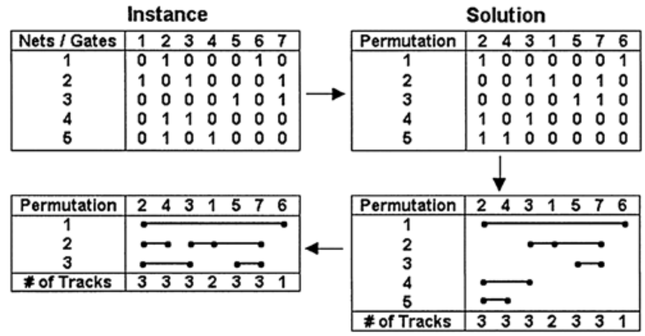

Fig. 6 The gate matrix layout problem, and an optimum solution, given by the permutation (2 4 3 1 5 7 6)

2 reversals versus 1 insertion move. On the other hand, the sequence $(n, \ldots, 6, 5, 4, 3, 2, 1)$ requires $(n-1)$ insertions versus $\lfloor n/2 \rfloor$ 2-exchanges versus a single reversal. Figure 5 displays the frequency of each distance metric, obtained from the distance of 500 random fixed asymmetric permutations to the identity permutation. To 'smoothen' the curves, the size of the permutations was taken uniformly from the interval [90, 110]. This figure clearly shows that there is a strong statistical tendency for two distinct solutions to be closer under insertion than under 2-exchange operations (since the reversal operator is NP-hard, we were unable to provide comparisons). We have found out that this relation also holds for cyclic and symmetric problems, and it might imply that search algorithms may be better served by the insertion operator than by the 2-exchange operator, a hypothesis that we leave as an open problem for further research.

5 An example

Let us provide a simple example which may clear the exposition. Consider the *gate-matrix layout problem*, a problem that we have shown to be NP-Hard [45]. A gate matrix layout circuit consists of a set of gates (vertical wires) with transistors (dots) that are used to interconnect the gates. There must be horizontal wires known as tracks interconnecting all the gates that share transistors at the same position. If we model the problem as a matrix problem, we are given a binary $I \times J$ matrix $P = \{p_{ij}\}$, with $p_{ij} = 1$ if gate j holds a transistor at position i, otherwise $p_{ij} = 0$. The wiring between gates is represented by the ones interconnecting the leftmost and rightmost gates, hence, the resultant matrix holds the consecutive-ones property.

One basic feature of the problem is that the sequencing of the gates does not alter the underlying logic equation implemented by the circuit. The number of tracks is the only variable determining the overall circuit area, since the number of gates is constant. Therefore, it is essential to sequence the gates in order to minimize the number of tracks, i.e., the number of necessary physical rows to implement the circuit, given by

$$Z^{\pi}_{GMLP}(P) = \max_{J \in \{1,\ldots,J\}} \sum_{i=1}^{I} p^{\pi}_{ij}.$$

Therefore, GMLP is the problem of $\min_{\pi \in \Gamma} Z^{\pi}_{GMLP}(P)$, where P is an instance of GMLP. The problem is displayed in Fig. 6, from matrix form to the final optimized VLSI layout.

Consider, for instance, that local search methods are employing the insertion operator. In this case, we can generate new, non-minimum solutions, with a known distance from the optimum (as given by Sect. 4).

For example, the permutation (2 4 7 5 1 3 6) demands 4 insertions in order to be retransformed to the optimum. It generates 5 tracks, yielding almost twice the minimum integrated circuit size. The permutation (2 1 3 4 5 7 6) demands 3 insertions, and generates 4 tracks, also more than the minimum.

Notice, however, that a lower number of necessary insertions does not imply that less tracks will be generated: the permutation (2 7 4 3 1 5 6) yields 5 tracks, while only requiring one insertion to achieve the global minimum. However, since at each point there are n^2 possible insertions, *the probability that the optimum will be achieved rapidly increases as the distance to the optimum solution decreases*.

Consider, now, that the system is maximizing the diversity of the solutions being explored, as given by $D(E)$ in Sect. 2.3 above, by using one of the suggested coordination policies. Move selection will hence be done by attempting to improve the quality of each solution being explored, *as long as it does not violate the space allocated to another solution under exploration*. This leads us to our major proposal: *the higher the diversity between solutions becomes, the higher the probability that at least one of these solutions lies closer to the optimum (and, as a corollary, that the optimum may be found)*.

6 Discussion

This work criticizes the assumption that there may be a mutually exclusive tradeoff between search intensity and search diversity in parallel systems. There are three contributions in this work. First, it is argued that search intensity and search diversity are not mutually exclusive. Even leading researchers have argued for ‘finding *the right balance* between intensity and diversity’, and this idea of a ‘balance’, or of a supposed tradeoff, between these desiderata immediately presupposes their mutual exclusion. By introducing a framework for collective models in which intensity is achieved at the process level, while diversity is achieved at the collective, systemwide level, it is demonstrated that, for systems with multiple search processes, these desiderata can co-exist simultaneously throughout the whole execution of the system (instead of alternating, or of gradually moving from diversity towards intensity).

A second contribution consists of the control mechanisms referred to as *coordination policies*. These are general principles for guiding the distribution (and also re-distribution) of search processes along the search space. Some additional coordination policies for re-distribution of search processes are presented in [38].

Finally, we also discuss innovative uses of distance metrics: by employing such metrics, it is possible to compute, for example, the distance between the current solution being explored and the best solution found. It is also possible to more clearly see whether a search process is exploring many distinct regions of the solution space or if it is instead gravitating around some point. Still another possibility is the development of tabu distances: instead of using lists of previous steps, there could be a “trail” of previously visited solutions (acting as ‘detractors’) and a corresponding distance (‘radius’) to be kept from each of these solutions, thus guaranteeing a high search diversity over the course of the search (and obviously countering the ergometric bike phenomenon). The distance metrics also enable the calculation of how much the processes are dispersed over the search space, and also, in case two (or more) processes cluster around a small area, to drive one (or more) out of that area, in order to eliminate the wasteful duplication of a computationally valuable search effort.

As an anonymous referee pointed out, perhaps the most promising line of further research lies with another sense of ‘tradeoff’. We have been discussing the *theoretical* sense of a tradeoff between intensity versus diversity; but the computational costs involved in circumventing this tradeoff might, at least theoretically, turn out to be prohibitive, which leads us to a *practical* or *concrete* tradeoff: One question concerns whether it is *possible* to obtain diversity and intensity; another question concerns whether it is *beneficial* to do so. In fact, one may argue that the tradeoff may really reside in information theory: at each point one wants to ‘ask the question which will reduce uncertainty the most’; under this perspective intensity and diversity are secondary; and information—and all the computational costs of obtaining such valuable information—becomes the primary issue.

The skeptical reader could argue that this paper does not present a specific optimization problem and a specific local search strategy—it does not demonstrate that simultaneous intensity and diversity is indeed advantageous in terms of algorithmic performance. This is done in order to discuss these issues *in general*, to discuss issues that are independent from problems or solution methods. Whether or not greater performance is to be expected is a significant empirical question, but it does not alter the fact that *the very idea of a mutually exclusive balance between diversity and intensity needs revising*—and that is the central thesis here, which deserves to be truly scrutinized by the research community. Because the proposed framework is detached from the heuristics underlying the search processes, from the distance metrics employed (which vary from operator to operator), and even from the optimization problem being solved, it deserves to be discussed independently (successful experiments have been carried out in [40]). This work is hence one of the very few analysis accomplished on a level of *meta-meta-heuristics*, because all arguments are independent of specific problem handled (such as scheduling, planning, TSP, etc.), of specific solution methods (such as genetic algorithms, simulated annealing, tabu search, etc.) and of specific neighborhood or genetic operators (2-opt, BOX crossover, etc.). We hope some of the insights brought out here will eventually open exciting unexplored territory for further heuristics research.

Acknowledgements Part of this work was executed when Dr. Linhares was visiting the Centre de Recherche sur les Transports (CRT) of the Université de Montréal, Canada; a visit for which the authors are grateful to Dr. Gilbert Laporte. We are grateful for the comments of the anonymous referees, and we appreciate financial support from grant #97/12785-8 of the FAPESP foundation, from the PROPESQUISA program of the Getulio Vargas Foundation and also from grants from the CNPq Foundation. The authors would like to thank three anonymous referees for numerous improvements in the paper, and for the mention of the use of the Hamming distance in memetic algorithms.

References

1. van Laarhoven PMJ, Aarts EHL (1988) Simulated annealing: theory and applications. Kluwer Academic, Norwell
2. Debuse JCW, Rayward-Smith VJ (1999) Discretisation of continuous database features for a simulated annealing data mining algorithm. Appl Intell 11:285–295
3. Ansari N, Sarasa R, Wang GS (1993) An efficient annealing algorithm for global optimization in Boltzmann machines. Appl Intell 3:177–192
4. Ombuki BM, Ventresca M (2004) Local search genetic algorithms for the job-shop scheduling problem. Appl Intell 21:99–109
5. Ghosh S, Ghosh A, Pal SK (2003) Incorporating ancestors’ influence in genetic algorithms. Appl Intell 18:7–25

6. Mendes A, Linhares A (2004) A multiple population evolutionary approach to gate matrix layout. Int J Syst Sci 35:13–23
7. Beasley JE (2000) Population heuristics. In: Pardalos PM, Resende MGC (eds) Handbook of applied optimization. Oxford University Press, London
8. Goldberg DE, Richardon J (1987) Genetic algorithms with sharing for multimodal function optimization. In: Grefenstette JJ (ed) Genetic algorithms and their applications (ICGA'87). Lawrence Erlbaum Associates, Hillsdale, pp 41–49
9. Ursem RK (2001) When sharing fails. In: Proc. of the third congress on evolutionary computation, pp 873–879
10. Ursem RK (2002) Diversity-guided evolutionary algorithms. In: Proc. of parallel problem solving from nature VII, pp 462–471
11. Glover F (1989) Tabu search—Part I. ORSA J Comput 1:190–206
12. Glover F, Laguna M (1997) Tabu search. Kluwer Academic, Dordrecht
13. Laguna M, Barnes JW, Glover F (1993) Intelligent scheduling with tabu search—an application to jobs with linear delay penalties and sequence-dependent setup costs and times. Appl Intell 3:159–172
14. Glover F (1977) Heuristics for integer programming using surrogate constraints. Decis Sci 8:156–166
15. Cung V-D, Mautor T, Michelon P, Tavares A (1997) A scatter search based approach for the quadratic assignment problem. In: Proc of the IEEE international conference on evolutionary computation, pp 165–170
16. Linhares A, Torreão JRA (1988) Microcanonical optimization applied to the traveling salesman problem. Int J Mod Phys C 9:133–146
17. Linhares A, Yanasse HH, Torreão JRA (1999) Linear gate assignment: a fast statistical mechanics approach. IEEE Trans Comput-Aided Des Integr Circuits Syst 18:1750–1758
18. Feo TA, Resende MGC (1995) Greedy randomized adaptive search procedures. J Glob Optim 6:109–133
19. Argüello MF, Bard JF, Yu G (1997) A GRASP for aircraft routing in response to groundings and delays. J Combin Optim 1:211–228
20. Resende MGC, Ribeiro CC (1997) A GRASP for graph planarization. Networks 29:173–189
21. Backhouse PG, Fotheringham AF, Allan G (1997) A comparison of a genetic algorithm with an experimental design technique in the optimization of a production process. J Oper Res Soc 48:247–254
22. Dowsland K (1996) Genetic algorithms: a tool for OR? J Oper Res Soc 47:550–561
23. Laguna M, Glover F (1993) Bandwidth packing: a tabu search approach. Manag Sci 39:492–500
24. Rochat Y, Taillard ED (1995) Probabilistic intensification and diversification in local search for vehicle routing. J Heuristics 1:147–167
25. Jain AS (1998) A multi-level hybrid framework for the deterministic job-shop scheduling problem, PhD thesis, University of Dundee, Dundee, Scotland, UK
26. Glover F (1990) Tabu search: a tutorial. INTERFACES 20:74–94
27. Colorni A, Dorigo M, Maffioli F, Manniezzo V, Righini G, Trubian M (1996) Heuristics from nature for hard combinatorial optimization problems. Int Trans Oper Res 3:1–21
28. Christofides N (1994) The bionomic algorithm. In: Proc of the associazione Italiana di ricerca operativa conference, Savona, Italy
29. Maniezzo V, Mingozzi A, Baldacci R (1998) A bionomic approach to the capacitated p-median problem. J Heuristics 4:263–280
30. Jones T (1995) One operator, one landscape. Santa Fe Institute Technical Report 95-02-025. Santa Fe Institute, Santa Fe, NM 87501
31. Boese KD, Kahng AB, Muddu S (1994) A new adaptive multistart technique for combinatorial global optimizations. Oper Res Lett 16:101–113
32. Merz P, Freisleben B (1999) Fitness landscapes and memetic algorithm design. In: Corne D, Glover F, Dorigo M (eds) New ideas in optimization. McGraw-Hill, New York
33. Reeves CR (1999) Landscapes, operators, and heuristic search. Ann Oper Res 86:473–490
34. Mendes A, França P, Moscato P (2002) Fitness landscapes for the total tardiness single machine scheduling problem. Neural Netw World 2:165–180
35. Mak K-T, Morton AJ (1995) Distances between traveling salesman tours. Discrete Appl Math 58:281–291
36. Caprara A (1999) Sorting permutations by reversals and Eulerian cycle decompositions. SIAM J Discrete Math 12:91–110
37. Linhares A (2004) The structure of local search diversity. WSEAS Trans Math 3:216–220
38. Linhares A (2001) Industrial pattern-sequencing problems: some complexity results and new local search models. PhD thesis, National Institute of Space Research, São José dos Campos, Brazil
39. Jelasitya M, Dombib J (1998) GAS, a concept on modeling species in genetic algorithms. Artif Intell 99:1–19
40. Linhares A (2004) Minimization of open orders: a re-distribution coordination policy. Int J Prod Res 42:1189–1205
41. Knuth D (1975) The art of computer programming, vol. 3. Sorting and searching. Addison-Wesley, New York
42. Aigner M, West DB (1987) Sorting by insertion of leading elements. J Combin Theory Ser A 45:306–309
43. Chen S, Smith SF (1996) Commonality and genetic algorithms. The Robotics Institute CMU-RI-TR-96-27, Carnegie Mellon University
44. Orlowski M, Pachter M (1989) An algorithm for the determination of a longest increasing subsequence in a sequence. Comput Math Appl 17:1073–1075
45. Linhares A, Yanasse HH (2002) Connections between cutting-pattern sequencing, VLSI desing, and flexible machines. Comput Oper Res 29:1759–1772

Alexandre Linhares is a Research Professor at the Getulio Vargas Foundation in Rio de Janeiro, Brazil. The Brazilian Computing Society gave him the honorable mention "for the great scientific contribution" in his Ph.D. Thesis. His work appears in international journals such as Artificial Intelligence, Biological Cybernetics, Behavioral and Brain Sciences, Cognitive Science, IEEE Transactions on Evolutionary Computation, Computers & Operations Research, Minds and Machines, International Journal of Systems Science, Journal of Air Transport Management, among others.

Horacio Hideki Yanasse is the head of the Special Technologies Center of the Brazilian Institute of Space Research. He holds a Ph.D. in Operations Research from the Massachusetts Institute of Technology. His work appears in the major OR outlets, such as Management Science, Operations Research, Discrete Applied Mathematics, Journal of the Operational Research Society, European Journal of Operational Research, among many others. He is the editor of the Brazilian Operations Research Journal, *Pesquisa Operacional*.