We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Gravitational Singularities in Particle Swarms. Application to the Discovery of Gene Expression Patterns in DNA Microarrays.

Gerard Ramstein
Polytech'Nantes, University of Nantes
France

1. Introduction

Particle Swarm Optimization (PSO) is an evolutionary computation technique that exploits social intelligence found in bird flocking or fish schooling (Kennedy & Eberhart, 1995). The PSO algorithm generates potential solutions expressed as a particle swarm exploring the search space. Like genetic algorithms, a fitness function indicates the quality of the candidate solutions. The fitness function may have more than one optimum and some local optima may also interest the user. Many variants of PSO are capable of dealing with multimodal optimization problems. As many real world applications present several equivalent or nearby solutions, tools must be provided for the identification of multiple optima. Algorithms have been designed that either explore sequentially the search space or introduce a kind of parallelism. For instance, by analogy with bird swarms, niching strategies have been implemented.

In our model, called WPSO, particles are considered as celestial bodies, subject to the effect of gravity. A high density of particles causes the creation of a gravitational singularity called a wormhole. A particle falls into a wormhole if it crosses a boundary called the event horizon. It is then rejected into another region of space. Wormholes possess two interesting features. Firstly, a wormhole indicates a potential solution: the aggregation of particles around a particular spot reveals the existence of a local or optimal fitness value. Secondly, the presence of a wormhole avoids the premature convergence of the whole swarm since particles that are close to a previously explored region are redirected into another space region. A wormhole can then be seen as a mutation operator that dynamically preserves the diversity of the swarm.

The WPSO algorithm has been applied to the discovery of gene expression patterns in DNA microarrays. This new technology enables the simultaneous monitoring of the expression levels of thousands of genes in particular conditions (the effect of pathologies, drugs or stress, …). Large-scale experiments can be carried out to observe the co-expression of genes in order to better understand the underlying biological processes. An expression pattern is an expression profile common to a set of genes, characterizing a cancer signature or a biological function such as immune response or programmed cell death. Since these patterns correspond generally to dense regions of the expression measurements, our method is particularly well adapted.

The rest of the paper is organized as follows. Section 2 discusses related work; Section 3 describes the principle of PSO; Section 4 is devoted to the presentation of WPSO and a study of its performances; Section 5 deals with the application of WPSO to microarray data analysis. Finally a conclusion and some perspectives are presented in Section 6.

2. Related work

Evolutionary algorithms have been successfully applied to solve multimodal optimization problems. The concept of speciation has been inspired from the observation of biological evolution. Different species are in competition with each other in territories containing finite resources. These species tend to occupy different environmental niches.

The Island model (Whitley et. al. 1998) is for instance a technique providing multiple subpopulations that helps to preserve genetic diversity. Niching methods have been incorporated in Genetic Algorithms to promote the formation and maintenance of subpopulations (Mahfoud 1995). Two kinds of approaches have been proposed: while parallel niching methods maintain several niches simultaneously, sequential niching methods apply iteratively a swarm on the search space to find multiple optima. Sequential Niching (Beasley et. al. 1993) is based on a derating function that operates on the fitness landscape at a position where a solution was found. Sharing (Goldberg 1987) is another mechanism for maintaining population diversity. The fitness of an individual is scaled by the number of similar individuals in order to penalize redundant solutions. This sharing function depends on the distance between individuals, which is a way to control the diversity of species. Deterministic Crowding (Mahfoud 1995) is a selection method that facilitates the formation of niches, these latter being maintained by an elitist scheme.

PSO algorithm can easily be extended to perform multimodal optimization. An intuitive adaptation consists in restricting the social cognition of a particle. In the SPSO algorithm (Li 2004), species are created dynamically at each iteration according to the location of the particles. A species is determined by a neighborhood, defined by a representative particle and a radius specified by the user. In the UPSO algorithm, introduced by (Parsopoulos and Vrahatis 2004), the evolution of the particle velocity is controlled by two terms related to a global and a local neighborhood. The PVPSO method proposed by (Schoeman & Engelbrecht 2005) uses vector operations to determine the particles belonging to a niche. More precisely, dot products indicate if a particle converges on the neighbourhood-best of a given niche. Another parallel niching technique has been proposed in (Brits.a et. al. 2002). The NbestPSO algorithm is based on neighborhood-best defined for each particle x_i as the center of the mass of the positions of all the particles in the topological neighborhood of x_i. Instead of a radius, the neighborhood is defined by the n closest particles of x_i. The same authors have implemented another niching algorithm, called NichePSO (Brits.b et. al. 2002). The initial swarm, called the main swarm, explores the search space using the cognition-only model (Kennedy 1997) allowing the particles to memorize their best position without exchanging any social information with their neighbors. Candidate solutions are identified by the convergence of the fitness of each particle. New sub-swarms are then created, characterized by a radius around the best particle in the sub-swarm. Sub-swarms can merge if they intersect and absorb particles from the main swarm when they cross the boundary of a sub-swarm. In (Özcan & Yılmaz 2007), the authors propose a modified version of NichePSO, called mNichePSO, that greatly improves the PSO performance by constraining the niche radius size to a predefined maximum value. The same work introduces a new

algorithm called CSPSO based on a random walk and a hill climbing components. The hill climbing strategy consists in moving a particle to the position that generates a better fitness and the craziness component introduces random positions to further explore the space.

3. Particle swarm optimization

PSO is an algorithm inspired by the model of social learning that simulates the flocking behavior of birds and fish. A PSO is based on a population of interacting elements (particles). A swarm of particles explores an n-dimensional space; the location of a particle determines a potential solution. Each particle knows the most promising position (*pbest*) encountered so far, as well as the best location (*gbest*) found by its neighbors (for our needs, the neighborhood corresponds to the whole swarm). Particles adjust their flight according to their own experience but are also influenced by their peers. The displacement of a particle is defined by its current position and a velocity vector that is updated at each iteration of the algorithm.

The algorithm starts with a population composed of random particles. The location of each particle can be arbitrary defined by using a uniform random sequence but it has been shown that randomized low-discrepancy sequences better cover the search space and then improve the performances of PSO (Nguyen et. al. 2007). In our implementation, particle locations have been initialized by using Sobol quasi random sequences. In the following step of the algorithm, the fitness value is computed for each particle and *pbest* revised if necessary, as well as *gbest*. The velocity vector v_i at iteration $t + 1$ is determined as follows:

$$v_i\,(t+1) = w.\,v_i\,(t) + g_1.R_1.(pbest\text{-}x_i) + g_2.R_2.(gbest\text{-}x_i) \tag{1}$$

where:

$$w = wmax\text{-}((wmax\text{-}wmin)/itermax).iter \tag{2}$$

In eq.1, g_1 and g_2 are weights associated to respectively the local (*pbest*) and global (*gbest*) terms, R_1 and R_2 are random values in $U(0,1)$ and x_i is the ith component of the location vector of the particle. The parameter w defined in eq.2 is a momentum coefficient that linearly decreases from *wmax* to *wmin*. The maximal velocity of a particle must be limited. Therefore, if $v_i\,(t)$ is greater than *vmax*, it is set to this maximal value.

The next step of the algorithm consists in updating the location vector as follows:

$$x_i\,(t+1) = x_i\,(t) + v_i\,(t+1) \tag{3}$$

The algorithm iterates until convergence is achieved or a maximal number of iterations is reached. The convergence of the swarm can be defined in different ways, a common approach being the convergence of the *gbest* particle. A common rule stipulates that its velocity must be quasi-null during a certain number of consecutive iterations.

4. Gravitational singularities and PSO

4.1 Description of the WPSO algorithm.
The WPSO algorithm proposed in this paper follows the same principles as those described previously, as long as the density of particles remains below a critical value. The convergence of many particles into a region causes the formation of a gravitational singularity, as it is the case when a massive star collapses. In our model, the agglomeration

of particles gives rise to the formation of a wormhole. An intra-universe wormhole is a hypothetical entity that connects two locations of the universe, offering a short cut through space and time. Particles approaching a wormhole are then instantaneously projected in another region (in our case a random location of the search space). This feature guarantees that a previously found solution is not reachable and it preserves the diversity of the swarm. The algorithm works as summarized in Fig. 1. In the beginning, particles move in a uniform space as in the classical model. After a certain time, a high density of particles gives birth to a wormhole. At this stage, the wormhole is not entirely formed. The group of particles (called W) that have created the wormhole are separated from the rest of the main swarm. This group goes across the space until convergence, as if its particles are the only individuals in the search space. When W converges, the wormhole is mature and remains in a constant location, determined by the position of the best particle in W at convergence. Particles belonging to W are then released and redirected in a random location and restored to the swarm; a new cycle can then be performed.
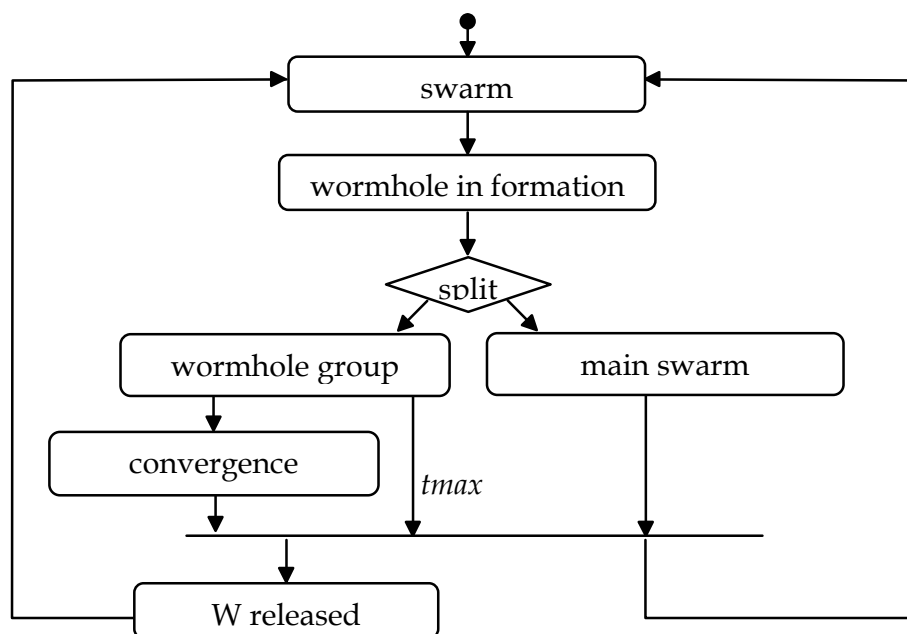

Fig. 1. Gravitational PSO algorithm

The wormhole is defined by its event horizon, that is the gravity field where the space-time is so bent that nothing can escape from it. In our case, the event horizon is the region where particles vanish. Our implementation determines this boundary by a radius around the best particle belonging to W. This radius remains constant from the birth to the mature stage of the wormhole; only the center of the wormhole moves during its formation.

We need also to specify the critical point that causes the formation of the wormhole. As already stated, the birth of a wormhole is due to a high density of particles. One could compute the density of every particle by examining its neighborhood. A less computationally expensive strategy consists in only considering the best particle at a given time. The density is then defined as the number of particles nw around $gbest$, its neighbourhood being defined by a constant radius. The value of the radius is the same as the one characterizing the event horizon. Our algorithm then possesses two input parameters: r, the radius of the event horizon; nw, the number of particles for the formation of a wormhole.

The formation of a gravitational singularity induces two types of particles: those at the origin of the wormhole and the rest of the swarm. The former group searches the refined location of an optimum. As one expects this group to converge in a short period of time, it is necessary to decrease the inertia of the particles. For that purpose, we consider that the particles evolve in different times; the lifetime of the group W corresponds to the maximal period of time *tmax* authorized for the maturity of a wormhole. The inertia of these particles is given by :

$$w = wmax- (wmax-wmin)*(tc/tmax) \qquad (4)$$

where *tc* is the time elapsed since the formation of W (*tc* ≤ *tmax* by construction). This formula replaces Eq. 2 for the particles in W; the main swarm possesses a constant inertia fixed to *wmax*.

The convergence of W group is determined by the velocity of its best particle. When the velocity is less than a given threshold $\varepsilon=10^{-5}$, we consider that the location is stable and the wormhole mature. Note that any particle can be trapped by already-formed wormholes. A group W can then be dispersed and may be eventually unable to converge after a period of *tmax* trials. In that case, the wormhole is not confirmed and a new cycle of search is performed. Another possibility has to be considered: the W group converges at the edge of a wormhole already constituted. This phenomenon can be observed when the event horizon is too small, causing particles to be repeatedly attracted by the wormhole. A solution could be to increase the radius *r* of the event horizon as long as this situation is encountered, but this has several drawbacks, it is notably time consuming and based on a false assumption: the basin of attraction is not necessary spherical. A better approach consists in merging a wormhole adjacent to a previously found wormhole. All one has to do is to keep the rejection power of the merged wormhole without considering that it corresponds to an actual optimum. A new mature wormhole is merged to a previously found wormhole if the distance between their centers is less than 1.20x*r*.

4.2 Experimental results.

Six well-known benchmark functions have been used for comparison sake. These two-dimensional functions presenting multiple global and local optima are described in Table 1.

Label	Function	Range	Global	Local
f1	$z =(x^2+y-11)^2-(x+y^2-7)^2$	-10,+10	4	0
f2	$z =sin(2.2\pi x+\pi/2)(2-\lvert y\rvert/2)\ (3-\lvert x\rvert/2)$ $+sin(2.2\pi\ y^2+\pi/2)\ (2-\lvert y\rvert/2)\ (2-\lvert x\rvert/2)$	-2,+2	4	8
f3	$z =cos(x)^2+sin(y)^2$	-4,+4	6	0
f4	$z =((cos(2x+1)+2cos(3x+2)+3cos(4x+3)+$ $4cos(5x+4)+5cos(6x+5))$ $(cos(2y+1)+2cos(3y+2)+3cos(4y+3)+$ $4cos(5y+4)+5cos(6y+5))$	-2.0,+2.5	2	38
f5	$z =(y^2-4.5\ y^2)xy-4.7cos(3x-\ y^2(2+x))\ sin(2.5\pi x)$ $+(0.3x)^2$	-1.2,+1.2	1	9
f6	$z=4x^2-2.1x^4+(1/3)x^6+xy-4y^2+4y^4$	-1.9,+1.9	2	4

Table 1. Bidimensional functions. The range limits the values of *x* and *y* coordinates. The two last columns respectively represent the number of global and local optima.

In (Özcan & Yılmaz 2007), these functions have been used to test the performances of the PSO algorithms presented in section 2: SPSO, NichePSO, mNichePSO, NbestPSO, UPSO, PVPSO) and CPSO. The same parameters and evaluation metrics have been taken into consideration for comparison purpose with WPSO.

The input parameters common to all the algorithms are given in Table 2. The additional parameters of WPSO are given in Table 3.

swarm size	50
number of iterations	25,000
wmin	0.6
wmax	0.8
g1	1.5
g2	1.5
vmax	range_size/20

Table 2. Input parameters. The parameter range_size is the size of the search space (e.g. for function F1, range_size = 20).

r	range_size/8
nw	15
tmax	500

Table 3. Input parameters specific to WPSO.

The measure proposed in (Özcan and Yılmaz 2007) has been retained for the estimation of performances: the overall success rate *osr* is given by:

$$osr = (gpr+gscr+lpr+lscr)/4 \tag{4}$$

where *gpr* is the global peak ratio, i.e. the ratio of the average number of global optima found to the total number of global optima. The local peak ratio lpr is a comparable measure applied to local optima. The global success consistency ratio *gscr* is the proportion of the runs in which all global optima are discovered to the total number of runs. The version dedicated to local optima is called *lscr*.

Table 4 shows the performance results observed for the functions of Table 2. CPSO and mNichePSO efficiently locate global optima in absence of local optima (functions f1 and f3). These two algorithms also respectively achieve the second and third best average performances, preceded by WPSO. Our algorithm performs well in functions presenting many local optima (functions f2, f4 and f5). It seems surprising that it is not always capable of finding all the optima of simpler functions, such as f1, but this is due to the restriction of the performance test requiring to set the same input parameters for all the functions. The experiments reveal that the most sensible parameter of WPSO is the radius denoting the event horizon. This may explain the weakness of some results, even if the merging of wormholes tends to minimize this effect.

PSO algorithm	f1	f2	f3	f4	f5	f6	avr.*osr.*	stdev.
SPSO	0.95	0.48	0.72	0.27	0.64	0.82	0.65	0.24
NichePSO	0.19	0.21	0.45	0.13	0.66	0.45	0.38	0.21
NbestPSO	0.90	0.43	0.74	0.31	0.46	0.58	0.57	0.22
CPSO	**1.00**	0.66	**1.00**	0.52	0.63	0.62	0.74	0.21
UPSO	0.88	0.54	0.55	0.48	0.45	0.54	0.57	0.16
mNichePSO	**1.00**	0.66	0.99	0.14	0.47	**0.91**	0.70	0.34
PVPSO	0.21	0.38	0.46	0.20	0.35	0.71	0.42	0.20
WPSO	0.98	**0.92**	0.99	**0.58**	**0.84**	0.62	**0.82**	0.18

Table 4. Average and standard deviation of *oscr* over 50 runs. Bold entries correspond to the best performance of the corresponding PSO algorithms.

5. Application to the analysis of DNA microarrays.

5.1 PSO for the extraction of gene expression patterns.

DNA microarrays allow to monitor the expression level of thousands of genes in a single experiment. They provide a better understanding of how cells work and respond to different stimuli. For instance, this technology is being applied to early diagnosis of cancer (Ochs & Godwin 2003) or drug discovery (Debouck & Goodfellow 1999). A common microarray data mining application consists in the analysis of the co-expression of genes. The underlying idea is that genes respecting a certain common expression pattern may be co-regulated. The identification of these patterns or signatures can help biologists to infer regulatory modules or biological functions (Yeung 2004). Generally, as microarray experiments focus on a particular disease or on the effect of a particular drug, only a low fraction of genes reveals a significant expression profile, related to the observed phenomenon. Therefore, we need to extract the most relevant signatures among many sets of poorly co-expressed genes. This task can then be seen as a multimodal optimization problem where the fitness is associated to the quality of the extracted signatures. In our application, a pattern is defined by an expression profile (i.e. a set of expression levels) and its neighborhood, that is the n closest profiles to the pattern according to the Euclidian distance. The parameter n is an input-specified value which determines the minimal number of genes constituting a signature. The fitness is defined as the mean distance of these n genes to their pattern. The signatures may be useful for further analyses; from a practical point of view, the set of genes can be easily extended to a larger neighbourhood by specifying a radius around the pattern, or to the contrary, the pattern can be discarded if its fitness is below a specified minimal value.

The WPSO algorithm described in the previous section has been adapted to the analysis of DNA microarrays. The rule leading to the merging of two wormholes has been refined. Experiments have revealed that the distance between two pattern profiles is not an optimal criterion to determinate if two signatures are similar. A more precise indicator consists in the estimation of the likelihood of this situation. The corresponding hypothesis assumes that the members of both signatures are actually issued from the same distribution. We use the Bayesian Information Criterion (BIC) to accept or reject a wormhole on this basis. The BIC (Schartz 1978) is associated to the maximization of a log likelihood function and is capable of estimating the best model. The BIC formula used in the PSO is described in (Pelleg & Moore 2000).
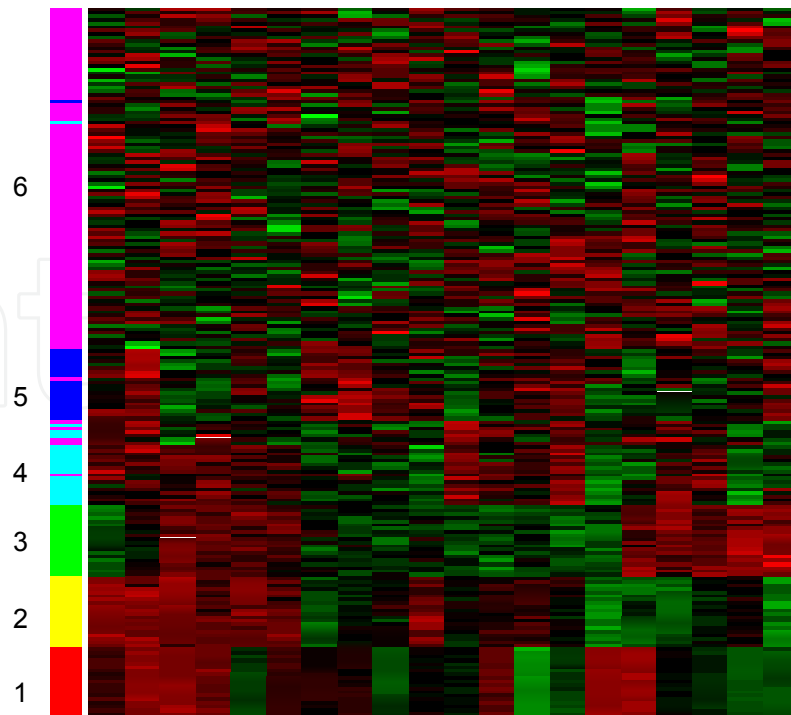
Fig. 2. Synthetic expression matrix

This figure shows a heat map graphical representation of the expression matrix. Data have been ordered in rows and columns using a hierarchical clustering, using the Euclidean distance and the ward method (dendrogram not showed). Rows correspond to genes and columns to samples (i.e. a microarray experiment). The conventional color palette shows under-expressed values in green, normal values in black and over-expressed value in red. The left bar indicates from which class each gene is issued ($\sigma(1)=0.1$, $\sigma(2)=0.2$, $\sigma(3)=0.3$, $\sigma(4)=0.4$, $\sigma(5)=0.5$, $\sigma(6)=1$). Note that the dense signatures ($\sigma=0.1$ to $\sigma=0.3$) are correctly aggregated in the figure, but the last two patterns have been aggregated with genes issued from the noise pattern.

5.2 Experiments on synthetic data.

To study the efficiency of the PSO, synthetic patterns have been generated. The creation of the expression matrix is based on the generation of five signatures. First, a pattern profile is randomly generated, using a uniform distribution in the range [-1,+1]. Second, 20 members of the pattern are generated: for each point of the profile, the gene expression value is expressed as the corresponding expression value of the pattern, perturbed by a Gaussian noise having a null mean and a standard deviation of σ. The homogeneity of the five gene sets are given by σ; the selected values are respectively (0.1, 0.2, 0.3, 0.4, 0.5). A random noise is added to the expression matrix, defined by a pattern having only null expression values and a strong standard deviation ($\sigma = 1$). A fraction f of the genes belonging to this noise pattern has been added to make the extraction process more complex and realistic. For instance, a fraction f of 5% corresponds to five patterns of 20 genes and a noise pattern of 1,900 genes (that is a matrix comprising 2,000 genes). Fig. 2 shows an example of synthetic expression matrix with $f=50\%$. One can observe that the five patterns cover a wide range of clusters, from the very homogeneous to the uncorrelated ones. For our experiments, four different fractions f of noise gene sets have been tested: 50%, 25%, 10%, 5%.
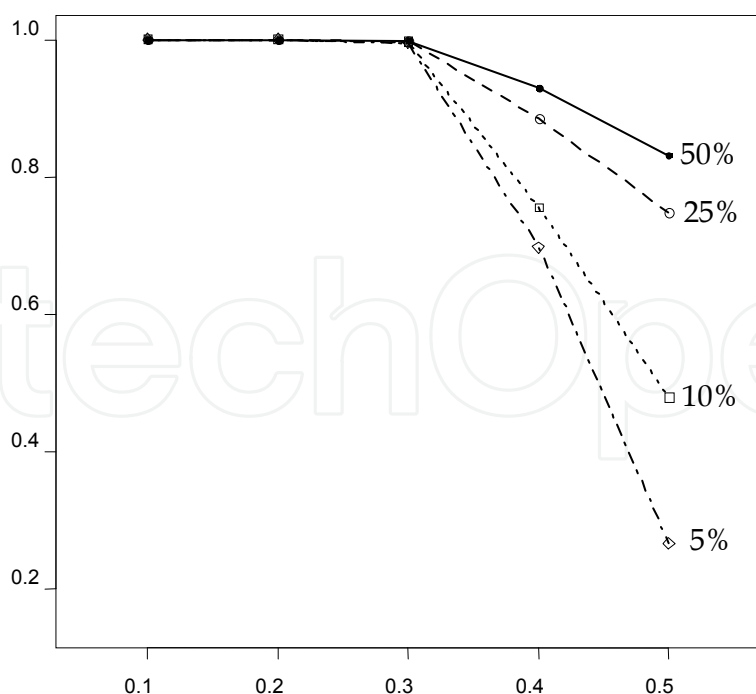
Fig. 3. Performance evaluation.

The four lines correspond to the mean value of the performance measure, expressed for five values of σ (indicated in abscise). Each line is relative to a different fraction *f*.

The same input parameters as for the previous experiments have been used, except for the event horizon radius (set to 10) and the dimension of the search space. Gene profiles have been normalized (mean set to zero and variance to one). This pre-processing reduces the variation of measurements between genes, and limits the search space to a uniform range set to [-3,+3].

The performance test identifies the patterns given by the WPSO: the 20th closest genes to each pattern of the matrix are extracted and compared to the actual list of genes generated for the creation of the matrix. The performance measure is the similarity of these two sets, expressed as the ratio of two cardinals: the intersection over the union. 50 random matrices have been produced for each of the four matrix sizes. The results are summarized in Fig. 3. One can observe that the PSO algorithm perfectly identifies strong signatures (σ=0.1 to σ=0.3), independently of the matrix size. The most heterogeneous patterns (σ=0.4 to σ=0.5) are more problematic and sensible to the matrix size. These are limit cases where the probability of finding a neighbor issued from the noise pattern is not negligible. As shown in the example of Fig. 2, the corresponding clusters are indeed not correctly discriminated by the hierarchical clustering algorithm. It is thus not surprising that the increase of the number of noisy genes decreases the performance measure.

5.2 Experiments on real data.

The WPSO algorithm has been applied to a well-known study (Ross et. al. 2000), which particularly illustrates the power of microarray analysis. This remarkable work explores the variation of expression of 8,000 genes among 60 cell lines derived from human tumors. The dataset then presents a vast potential of expression patterns, due to the variety of the

observed phenotypes. In a similar manner as that proposed in the original paper, the set of genes has been filtered to remove the genes that do not show a significant variation of expression among the cell lines. After this pre-processing, a set of 1,200 genes has been retained and normalized like in the previous section. The same algorithm has also been applied.
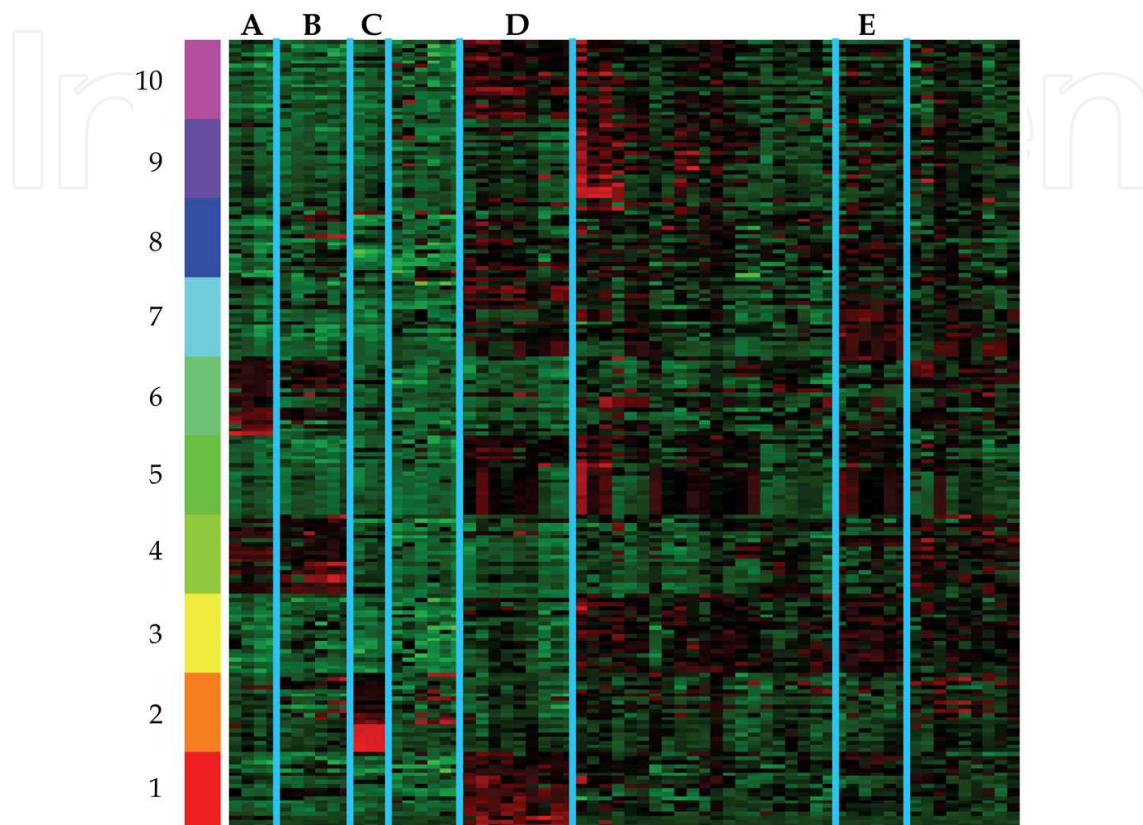

Fig. 4. Signatures obtained from the WPSO algorithm (same convention as in Fig. 2).

Fig. 4 shows the resulting expression patterns. An interesting feature of these signatures is that they are related to sub-sets of cell lines specific to certain tissue origins. The blue lines indicate the most remarkable observations: for instance, columns labeled A (resp. B, C, D, E) are notably differentially expressed in the 6th (resp. 4th, 2nd, 1st, 5th) signature and specific to samples belonging to breast (resp. colon, leukaemia, melanoma, renal) tumors.

The overall average silhouette width (*oasw*) is a measure expressing the tightness and the separation of a clustering (Rousseeuw 1987). The *oasw* of the ten extracted signatures is equal to 0.25. A positive value denotes a correct partition, the maximal theoretical value being 1. To assess the relevance of this result, different sets of ten signatures have been generated. Firstly, a clustering of the data into k clusters has been performed using the kmeans algorithm. The value of k has been issued from a uniform random value between 10 and 40. Secondly, a subset of 10 clusters has been randomly selected and for each cluster, a signature has been completed using the 20 closest data around the centroid. For 10,000 random trials, the mean score of *oasw* was 0.15 (s.d. 0.04) and the best score was *oasw* =0.24 (with k=13). This result demonstrates that WPSO discovers signature sets with remarkable high cohesion.

6. Conclusion

An original PSO algorithm called WPSO has been proposed, inspired from the concept of gravitational singularity. The convergence of particles generates a wormhole whose action is twofold: during its formation, it searches the best location of an optima; in a mature phase, it rejects particles to other regions of the search space. Experiments reveal that this algorithm provides efficient results on benchmark functions. A bioinformatics application has been implemented to extract signatures of co-expressed genes from microarray data. On a real experiment, the discovered patterns are tight, well separated and related to known phenotypes. As the size of event horizon determines the efficiency of the WPSO algorithm, a major improvement would consist in an automatic adaptation of this parameter according to the shape of the fitness function. A potential indicator for this optimization could be the frequency of particles absorbed by a wormhole.

7. References

Beasley, D.; Bull, D.R.; Martin, R.R. (1993). A Sequential Niching Technique for Multimodal Function Optimization, *Evolutionary Computation,* 1(2), p 101-125,   MIT   Press. ISSN 0302-9743.

Brits, R.; Engelbrecht, A.P.; van den Bergh, F. (2002). Solving Systems of Unconstrained Equations using Particle Swarm Optimization, in *Proceedings of the Int. Conf. on Sys.,Man and Cyber.*, vol. 3, p. 6, Piscataway, NJ , ISBN 0-7803-7437-1, Hammamet, Tunisia.

Brits, R.; Engelbrecht, A.P.; van den Bergh, F. (2002). A niching particle swarm optimizer, in *Proceedings of the 4th Asia-Pacific Conf. on Simulated Evolution and Learning*, vol. 2, pp. 692-696, Singapur.

Debouck, C.; Goodfellow, P. N. (1999). DNA microarrays in drug discovery and development, *Nat. Genet.*, vol. 21, pp 48 − 50, January.

Goldberg, D.E.; Richardson, J. (1987) Genetic Algorithm with Sharing for Multimodal Function Optimization, in *Proceedings of the Second International Conference on Genetic Algorithms*, p 41-49, Cambridge, Massachusetts, United States .

Kennedy, J.; Eberhart, R. (1995). Particle swarm optimization, in *Proceedings of the IEEE Int. Conf. on Neural Networks*, pp. 1942–1948, Piscataway, NJ.

Kennedy, J. (1997). The particle swarm: Social adaptation of knowledge, in *Proceedings of the International Conference on Evolutionary Computation*, p 303–308. IEEE Service Center, Piscataway, NJ.

Li, X. (2004). Adaptively Choosing Neighborhood Bests using Species in a Particle Swarm Optimizer for Multimodal Function Optimization, in *Proceedings of GECCO 2004*, pp. 105-116, LNCS 3102, eds. Deb, K. et al., Springer-Verlag, Seattle, USA.

Mahfoud, S. (1995). Niching Methods for Genetic Algorithms. *PhD thesis*, University of Illinois at Urbana Champaign, intern. report 95001.

Nguyen, Q. U.; Nguyen, X. H.; McKay, R. I.; Tuan, P. M. (2007). Initialising PSO with randomised low-discrepancy sequences: the comparative results, in *IEEE Congress on Evolutionary Computation*, pp 1985-1992, 25-28 September, Singapore.

Ochs, M. F.; Godwin, A. K. (2003). Microarrays in cancer: research and applications, BioTechniques,Vol. suppl., pp 4--15, March.

Özcan, E.; Yılmaz, M. (2007). Particle Swarms for Multimodal Optimization. *Proceedings of the 8th international Conference on Adaptive and Natural Computing Algorithms*, Eds. Lecture Notes In Computer Science, vol. 4431. Springer-Verlag, Berlin, Heidelberg, pp 366-375, Warsaw, Poland, April 11 - 14. ISSN 0302-9743.

Parsopoulos, K. E.; Vrahatis, M. N. (2004). UPSO: A Unified Particle Swarm Optimization Scheme, Lecture Series on Comp. and Computational Sci., Vol. 1, *Proceedings of the Int. Conf. of Computational Methods in Sci. and Eng.*, Vouliagmeni-Kavouri, Greece, pp. 868-873.

Pelleg, D. Moore, A. (2000). X-means: Extending K-means with efficient estimation of the number of clusters, in *Proceedings of the 17th International Conf. on Machine Learning*, pp 727--734. ISBN 1-55860-707-2.

Ross, D. T.; Scherf, U.; Eisen, M. B.; Perou, C. M.; Rees, C.; Spellman, P.; Iyer, V.; Jeffrey, S. S.; Van de Rijn, M.; Waltham, M.; Pergamenschikov, A.; Lee, J. C.; Lashkari, D.; Shalon, D.; Myers, T. G.; Weinstein, J. N.; Botstein, D.; Brown, P. O. (2000). Systematic variation in gene expression patterns in human cancer cell lines. Nat Genet ,march,vol. 24(3), pp 227--235, ISSN 1061-4036.

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* No 1, vol. 20, no 1, pp, 53-65. Elsevier Science Publishers B. ISSN 0377-0427.

Schoeman, I.L.; Engelbrecht, A.P. (2005). A Parallel Vector-Based Particle Swarm Optimizer, *Proceedings of the International Conf. on Neural Networks and Genetic Algorithms*, pp. 268-271. ISBN 978-3-211-24934-5

Schwarz, G. (1978). Estimating the Dimension of a Model, *The Annals of Statistics*, vol. 6, no 2, pp 461--464, ISSN 0090-5364.

Whitley, D.; Rana, S.; Heckendorn, R.B. (1998). The Island Model Genetic Algorithm: On Separability, Population Size and Convergence, *Journal of Computing and Information Technology*, vol. 7, pp. 33−47.

Yeung, K.; Medvedovic, M.; Bumgarner, R. (2004). From co-expression to co-regulation: how many microarray experiments do we need?, *Genome Biology*, vol. 5, no 7:R48, ISSN 1465-6906.

Evolutionary Computation

Edited by Wellington Pinheiro dos Santos

This book presents several recent advances on Evolutionary Computation, specially evolution-based optimization methods and hybrid algorithms for several applications, from optimization and learning to pattern recognition and bioinformatics. This book also presents new algorithms based on several analogies and metafores, where one of them is based on philosophy, specifically on the philosophy of praxis and dialectics. In this book it is also presented interesting applications on bioinformatics, specially the use of particle swarms to discover gene expression patterns in DNA microarrays. Therefore, this book features representative work on the field of evolutionary computation and applied sciences. The intended audience is graduate, undergraduate, researchers, and anyone who wishes to become familiar with the latest research work on this field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Gerard Ramstein (2009). Gravitational Singularities in Particle Swarms. Application to the Discovery of Gene Expression Patterns in DNA Microarrays., Evolutionary Computation, Wellington Pinheiro dos Santos (Ed.), ISBN: 978-953-307-008-7, InTech, Available from: http://www.intechopen.com/books/evolutionary-computation/gravitational-singularities-in-particle-swarms-application-to-the-discovery-of-gene-expression-patte

INTECH
open science | open minds