Reviewing ensemble methods in Particle Swarm Optimization (PSO)

[1]Pranjali Dewangan, [2]Dr. Neelam Sahu

Abstract

The Particle Swarm Optimization algorithm (Particle Swarm Optimization or PSO, [Kennedy and Eberhart, 1995] is an algorithm bioinspired, which is based on an analogy with the social behavior of certain species. Two basic influences of inspiration are recognized in PSO:

- *The movement of flocks of birds, in which each individual dis- square using simple rules for adjusting its speed depending on of the observations he makes about the close individuals in the flock.*

- *A social model, in which each particle represents a belief, and influences between individuals, the approach of individuals to others by diffusion of said beliefs.*

Keywords: *Particle Swarm Optimization (PSO), algorithm*

I. Introduction

1.1 Foundations of the PSO algorithm

In general lines, the algorithm seeks to find the global optimum of a function (fitness function) by moving a set of "Particles" (swarm) in a space defined by the number of parameters of the function. Each particle is a possible solution, that is, a set of parameters, which are "evaluated" by calculating the fitness value that corresponds to them. In this movement, the particles use historical information (the result of your past scan), as well as information about your neighbors (other particles in the swarm).

On the other hand, to complete the specification of the algorithm, you have to define certain parameters of the algorithm (stop criteria, type of neighborhood, values of some constants, form of initialization). The swarm starts out scattered throughout the search space, but over time the particles converge them towards a reduced area of space in which they intensify the search for the solution.

Each particle is characterized by a position vector x i , a vector of velocity v i , and a memory containing the vector with the best position of the particle so far p i . The position vector encodes a solution to the optimization problem, so that each coordinate corresponds to the value of one of the parameters of the function to be optimized.

PSO models cooperation between individuals by defining a neighborhood for each particle, so that each iteration selects a candidate from neighborhood to guide the search for the particle.According to the authors, the algorithm was initially designed with a neighborhood dynamic, given the analogy with the flock of birds. However soonit was

[1] Scholar, Dr.C.V.Raman University,Bilaspur, Chhattisgarh
[2] Associate Professor,Dr.C.V.Raman University,Bilaspur, Chhattisgarh

determined that, for the optimization problems considered, it was su- efficient and efficient for the neighborhood of each particle to be defined by static form: that is, each particle has a priori defined neighbors.In this it approximates cellular genetic algorithms, cellular automata, and various neural computation models.

Research carried out in this field confirms that the topology thatgbest is only suitable in cases of uniform optimization problems.give them, that is, when there is no risk of premature convergence, since there is only one optimum and it is global.

At the opposite end (low interconnection), the Local topology is defined Best ("pbest"), with a cardinality that is usually reduced with respect to number of particles in the swarm. In many cases the number of neighbors remains at 3 or 5, including the particle in.

In Figure 1 the two mentioned topologies are represented. AtReferenced work([KennedyandMendes,2002])are analyzed in a empirical other possible models, which, however, have not reached a important dissemination in the PSO community.

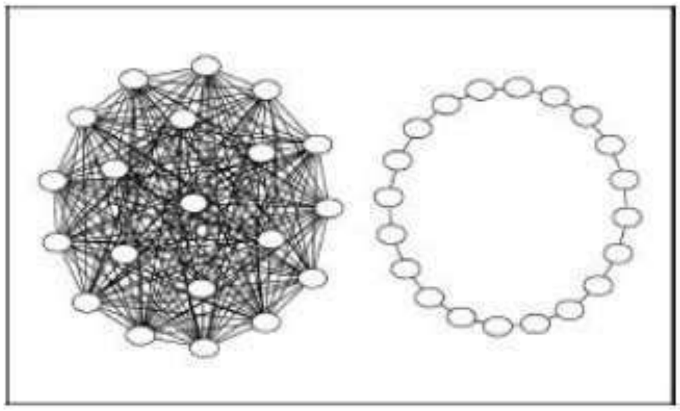

Figure 1 Neighborhood topology in PSO of type gbest (left) and lbest (right), with the particles represented in a two-dimensional space (figure reproduced from [Kennedy and Mendes, 2002])

There are also versions of PSO with dynamic neighborhoods, in which that the relationships between the particles evolve over time; although the The use of dynamic neighborhoods has a higher computational cost, it can be interesting for certain types of problem.

Finally, there are applications of PSO in which the concept of neighborhood dario is defined in a more complex way. An important case is the algorithms PSO for multi-objective optimization. In these, the purpose of the swarm consists of optimizing several fitness functions simultaneously; therefore-In general, it is not possible to use an algebraic criterion to determine what is the best solution among a set of neighbors, but there may be several particles that represent equally good compromises between the various objectives (Pareto set).

In Multiobjective PSO (MOPSO, [Coello et al., 2004]), the neighborhood of a particle is determined as a function of the set of solutions already obtained by other particles in $\Re$ n . When selecting a neighborhood, choose a set of particles whose vectors f are in a region $\Re$ n specific: regions of the Pareto front in which they have found few solutions. Among said "leading" particles are randomly selects the one that attracts the particle being considering.

1.2 Modification to the original PSO algorithm

During the first decade of the use of the PSO algorithm, numerous works that, starting from the original swarm, proposed modifications in order to improve its characteristics, usually in the form of empirical ma. Likewise, some works appeared that carried out analyzes theoreticians of the swarm equations, in order to better understand the mechanism of its operation: stability, convergence, study of trajectories and influence of parameters.

In order to study the convergence of the algorithm, and limit the explosion of the swarm, several mechanisms were studied:

Speed limitation. It simply consists of setting a maximum limit greater than the absolute value of the velocity components. That is to say, a parameter V max was set and the limitation condition was imposed next: | v t + 1 i| = minimum (| v t + 1|, V max). This mechanism has the pro- The problem that the parameter V max depends on the size of the search (X max), since in large search spaces, speed-small numbers would mean that swarm would not travel all the space or it would take a large number of iterations to do it. In the Initial work used to propose the value Vmax = Xmax .

In [Shi and Eberhart, 1998] a parameter or factor of inertia, w (see Equation 2.14). It is argued that the term of inertness cia of the equation is important for the swarm to be able to perform global searches, beyond the boundaries of the space defined by the positions of the initial particles. This is justified by the fact that, with the initially proposed parameters (c 1 = c 2 = 2.0), inmedium the swarm tends to contract towards the best "inside" positions.

The authors propose to include an explicit parameter with the objective to allow calibrating the balance between exploration and exploitation, by add the possibility of varying the value of said parameter as a function of of the iteration or state of the algorithm. In their analysis they conclude that, when the value of w is small, the swarm tends to behave like an algorithm that performs local searches, and relies more strongly initialization. When w grows, it goes on to do one more search global, but finds the optimum with more difficulty and greater number of iterations. Through the experimental study carried out, it is proposed a commitment value 0.9 <w <1.2.

$$V_{id}^{t+1} = w.{}_{id}^{vt} + C_1.\psi_1.(P_{id}^{t} - X_{id}^{t}) + C_2.\psi_2.\left(P_{gd}^{t} - X_{id}^{t}\right) \quad \boxed{2.14}$$

In the cited work, the use of a value of w decreasing is also proposed. enough over time. Subsequently, this proposal was analyzed from detailed in [Shi and Eberhart, 1999]. Although in this study the decreasing inertia factor seems to improve the behavior of the swarm, a study is shown in [Zheng et al., 2003] in which PSO has better results when using a value of w that grows with time instead of decreasing. Therefore, the behavior results be dependent on the problems considered.

In [Clerc, 1999] an alternative to the inclusion of the factor of inertia, which is discussed in more detail in [Clerc and Kennedy, 2002]. In this work, a theoretical study of the dynamics of the swarm was carried out. bre and it was observed that the convergence of the algorithm could be guaranteed under certain conditions by including additional parameters They are called " constriction parameters" . These results are obtained by analyzing simplified versions. swarm falls, and specifically versions in which they are removed random factors. These properties are subsequently confirmed data empirically using the full version of the swarm. The Constriction parameters are an alternative to the inertia mechanism. cia and speed limitation.

In [Eberhart and Shi, 2000] the PSO version was compared with cons- triction with the inertia version. The bottom line is that the methodwith constriction is a particular case of the method with inertia, since some parameters can be calculated as a function of others. However the method with constriction still requires the use of the constraintVmax = Xmax .

In [Trelea, 2003] an analysis of convergence and a study of the trajectories of the particles as a function of the values of the parameters. Aspects such as the size of the population are also analyzed. At work the recommended values for the parameters under study are confirmed previous god. On the other hand, it is stated that the conclusions obtained by the swarm with random factors are similar to those obtained without them, but their inclusion slows down convergence and increases hence the exploration of the search space, thus preventing some situations of premature convergence.

1.3 Theoretical analysis of PSO

PSO is a population-based optimization algorithm, but not a evolutionary algorithm, whose conceptual framework would lack a process of lesson between solutions. However, there is a relationship between PSO equations and some crossover operators in algorithms evolutionary algorithms (Evolutionary Algorithms, EA [Baeck et al., 2000]) also use two for numerical optimization. The study of this relationship is interesting because it allows to analyze the characteristics of PSO using terminology of said field (evolutionary algorithms).

To visualize this relationship, I would consider a complete meta-population put by individuals in their current positions, as well as memory of all of them (the best previous positions). In terms of EA, for each individual in the "current" population, a "child" is generated in the population. tion of the next iteration, product of a crossover operator which includes three parents, the three vectors x i , p i , and p g . The new individual generated replaces the parent, always keeping the best of both in the "Memory" of the particle as parent for next generations.The concept of neighborhood, when it is of type Ibest and static, existsalso in the so-called cellular genetic algorithms.

In [Trelea, 2003] the “deterministic swarm” is defined with the Equations sections 2.15 and 2.16; the study therefore considers the inclusion of inertia factor (a). In Figure 2.6 some of the path forms that are obtained, based on the values of a, b, for, c = 1 and d = 1. It is observed that, in all cases, the particle is approximates the best position, located at position y = 0, but can do it smoothly, or with a damped oscillation of various characteristics, depending on the values of said parameters.

In [van den Bergh and Engelbrecht, 2006] the analysis included goings the inertia factor and introducing the effects of the terms random. It is determind that these introduce distubances on theStudied behavior, which in general tend to include some diverstility even in conditions where the swarm is close to convergence.

$$V_i^{t+1} = a.v_i^t + b.(P_{id}^t - X_{it}^t) \qquad \boxed{2.15}$$

$$X_i^{t+1} = c.x_i^t + d.v_i^{t+1} \tag{2.16}$$

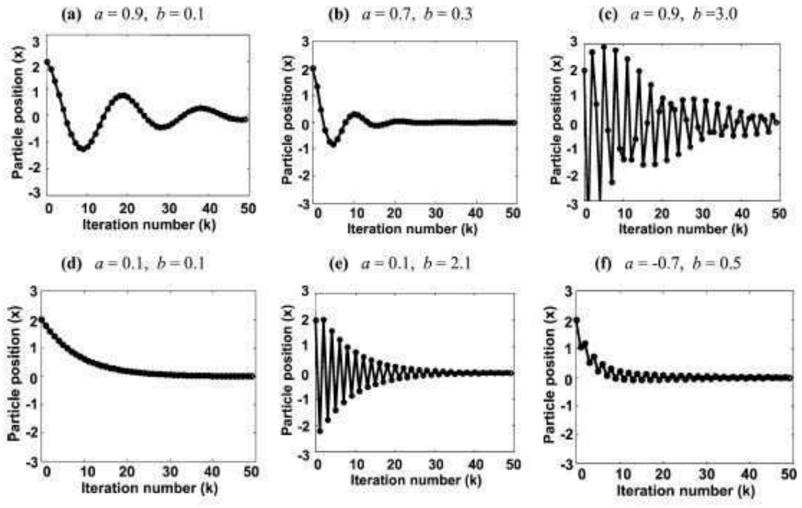

Figure 2 Representation of the trajectory of an isolated particle in a space one-dimensional, for different values of the PSO parameters (figure reproduced from [Trelea, 2003])

Complexity of the PSO algorithm. However, it is possible to approach the study PSO theory considering this randomness, as in [Clerc, 2006] and [Jiang et al., 2007]. In both analyzes, a swarm is considered in the "Stagnation '(stagnation), defined by the particles behave in an independent way. In other words, no new values are found that differentiate the points to which the particles are attracted.

In [Poli and Broomhead, 2007] and [Poli, 2008b] a study of PSO using Moment Analysis, under conditions similar to those above. three jobs. The authors characterize the statistical distribution of the samplingthat makes PSO of your search space. This type of study is carried out with the purpose of allowing subsequent analyzes that would allow characterizing systematic, not empirical way, the type of problems whose characteristics make them suitable for resolution through PSO.

1.4 PSO standard version

In order to reunify the proposals for both equations of motion as values for the PSO parameters, in [Bratton and Kennedy, 2007]It is intended to define what should be considered a reference version of PSO or “standard PSO”. It is a compendium of best practices and proposals to evaluate future modifications of the algorithmput. The authors of the work propose to use this document as a basis comparison or evaluation for any subsequent modification proposal and for reference software implementations.

One of the proposals of [Clerc and Kennedy, 2002] has been adopted as mo standard expression for PSO. The parameter χ or factor ofconstriction, as indicated in Equation 2.17.

$$V_{id}^{t+1} = \chi.(V_{id}^t + c_1.\psi_1.(P_{id}^t - X_{id}^t) + c_2.\psi_2.(P_{gd}^t - X_{id}^t)) \tag{2.17}$$

The parameter χ is derived from the rest of the parameters by the Equation 2.18,where$\psi=c_1+c_2$.

$$V_{id}^{t+1} + \frac{2}{2-\psi-\sqrt{\psi^2-4_\psi}} \quad \boxed{2.18}$$

With this expression, with ψ <4, the behavior of the swarm is believes that it spirals to and around the best solution, withoutrate of convergence, while with ψ> 4 the convergence would be fastand guaranteed. For simplicity, a value c 1 = c 2 and ψ = 4.1 is chosen , which results in the recommended parameters for the standard PSO swarm:χ = 0.72984, c 1 = c 2 = 2.05.

The recommended number of particles is also analyzed, and it is proposed that this varies between 20 and 50 depending on the execution time or cost desired computational.

Regarding the neighborhood topology, it is admitted that, in practice, the lbest topology performs better than gbest when evaluating the swarmon problems of a certain complexity. The gbest topology will be preferred only-mind in case the most important factor of evaluation is speed convergence or number of convergence risk assessments early.

In the standard PSO swarm an implementation criterion is introduce edtion of restrictions. It is recommended that the swarm does not evaluate the function of fitness when the position violates some restriction. In this way,that position will never update the best position of the particle. With bliss modification, the social terms of the equation of motion will tendto return the particles that violate the space constraints valid solutions.

II. Classification by Particle Swarm Optimisation

There are several jobs in which PSO is used to generate rules of classification, of the various types identified in Section 2.1.1: rules based on predicates, fuzzy rules, or proximity classifiers.

One of the first works was proposed in [Sousa et al.,2004], where PSO is used to obtain classification rules defined by intervalues for problems with two classes. In this work, each particle encodes an interval for each of the parameters of the classification problem. An interval of "indifference" is also reserved, which represents the situation in which any value of the attribute is valid. When the values of the parameters of a pattern "fall" within the ranges defined by the particle, the rule assigns the predominant class in the data set.

With a form of coding similar to the previous work, there is a under [Holden and Freitas, 2007] in which a hybrid version is proposed of PSO and ant colony (Ant Colony Optimization, ACO). In saying work, PSO is used to select induction rules based on inter- values for the numerical attributes of the problem, and ACO for the attributes discreet. PSO is also used, simultaneously, to determine the numerical parameters of the ACO algorithm.

In this case, various quality measures of the rules are tested; Interestingly, the fitness function depends on the class of the ruler:

For the majority class, the authors use the quality function Q1

(Equation 2.20) to predict the majority class

For the other classes, a different measure is used, based on the Sensitividad × Precisio´n (Equation 2.21); then modify the latter to use only the corrected precision value (Equation 2.22). In this second version, the value k (number of classes) is used to modify the usual value of the precision measurement (T P /(T P + F P)). In this way it is intended, simultaneously, to consider solutions precise and "simple". They determine that the best option for the problems considered is one that uses corrected precision (Equation 2.22).

$$Q_i^{3a} = \frac{TP}{TP+FN} \times \frac{TP}{TP+FP} \qquad \boxed{2.21}$$

$$Q_i^{3b} = 1 + \frac{TP}{1+k+TP+FP} \qquad \boxed{2.22}$$

In more recent works such as [Wang et al., 2006] and [Wang et al., 2007], a similar approach is used.

PSO can also be used to optimize a deduction model blurred [Esmin, 2007]. First, an encoding is defined that allows displaying the complete classifier on one particle; in the case of a system based on fuzzy rules, this representation includes the definition of the Necessary membership functions and the expressed classifier rules based on the fuzzy terms defined for each attribute. In this work, the model is made up of m inputs and one output, the function membership of each entry includes three triangular fuzzy terms, and each particle encodes a complete set of rules (all combinations possible nations of fuzzy terms). In this case, the position of the particle contains the values of:

1. The limits of the triangular terms in each membership function(inputs and outputs)
2. The output that corresponds to each possible rule

The mean square error function is used to evaluate the classifier between the predicted and expected value.

The proposal of [de Almeida and Pozo, 2007] is of interest. In this- In this case, the problem is to generate induction rules for problems with discrete attributes. The way to do it is to use a multiobjective PSO algorithm, so that, for each rule, they represent measured by a particle, a pair of criteria is evaluated simultaneously. I know experiment with a different pair of objectives: precision on possible examples positive and negative in one case, and sensitivity and specificity in another. The Particles evolve so that a set of rules is obtained that represent the best compromise solutions between the two goals considered. The results are difficult to evaluate, since the authors trans- they form the data sets that they use to reduce them to two classes. The Key contributions of the work are the use of a Michigan approach (one rule per particle) and the multi-objective evaluation criterion of the rules. Also in [Ishida et al., 2008] a multi-objective approach is used in classification by rules, using to evolve solutions the MOPSO algorithm ([Coello et al., 2004]).

There are other works that do not use rules of induction, but rules of proximity. As discussed in Section 2.1.2, proxy classifiers have good results in many domains, especially in cases in which knowledge about the structure of the data is limited. These classifiers are closer to the classification strategy than I will use the PSC algorithm that I propose later, since all they are nearest neighbor sort apps with replacement of prototypes.

In [Falco et al., 2006] PSO is used to locate a centroid by each class of a classification problem. In this case, each particle it contains as many centroids as classes. As a measure of fitness,use Equation 2.23. The training set is divided into this E in as many subsets E i attending to the class of patterns; then for each class i, the distances from each pattern are added training of said class (p k) to the centroid that corresponds to ponde (centroid i). The total is divided by the number of patterns in the training set that are of said class (| E i |). This measure quality is interesting because it does not directly calculate quality of the classification. The resulting classifier is limited to a single centroid by class.

In [Iswandy and Koenig, 2008] PSO is used to refine a set to prototypes for use as a nearest neighbor classifier. In this case the set is generated by different algorithms, and PSO is only used as an optimization on the solution of the previous algorithms. Each particle encodes the complete set of prototypes.

In [Nanni and Lumini, 2009] a fixed set of prototypes is coded with a conventional approach, similar to the one I use in this work as a measure of comparison (Section 3.2). The authors of end up that the PSO algorithm by itself is not competitive, but In some domains, the result can be improved by training sequences tially a set of independent classifiers and employing then this set through a voting system.

$$Q_i^2 = \frac{1}{|E_i|} \times \sum_{P_k \in E_i} d\ (\text{centroide}_{i,}\, P_k) \qquad \boxed{2.23}$$

There is another work that is interesting for our approach, since the coding of the problem is very similar. It is about the setting of [O'Neill and Brabazon, 2008], which combines concepts extracted from PSO with Self-Organizing Maps (SOM, [Kohonen, 1995]). The result is determined nominates Self Organizing Swarm (SOS). While the resulting application is an application of clustering, I cite this work because the coding of a prototype (neuron) for each particle, and its movement following PSO equations, it approximates the work done by us.

In SOS, the particles are arranged topologically in a rectangular matrix. tangular that determines the neighborhood relationship between them. The algorithm pro- yields incrementally, as in SOM. Thus, it is considered sequentially the set of training patterns; for each pattern, it is determined a "winning" particle as the particle closest to it; final- Mind you, the winning particle updates its position using the position of the pattern as a guide and a version of the PSO equations (Equation 2.14). Furthermore, when a particle is displaced, the position is also influenced of neighboring particles.

Altogether, it is a SOM in which the positioning rules The neurons in the mapping layer use the concepts of inertia and memory of PSO. However, unlike what happens in PSO, the in- Teraction occurs between particles and patterns, and only the particle moves which turns out to be closer to the pattern that is considered at all times.

There are other works of application of PSO to the clustering problem, that contribute ideas that are of interest to the development of classifiers. In [Omran et al., 2006] the binary PSO swarm is used in combination with a clustering algorithm to determine the optimal number of clusters.

III. Particle swarms for classifier optimization

There are other works where PSO is used in combination with other classifiers, so that it contributes more or less closely to improving them. In this case PSO is not used to find the classifier rules. I describe some examples below.Already in the first works with PSO this was proposed as a technique of training of neural networks [Kennedy et al., 2001]. In this sense It is possible to propose a network of neurons for classification that uses PSO to determine well the weights of the connections, well the architecture of the network,well the combination of both. In [Settles and Rylander, 2002] it is used swarm to optimize only the weights of a network that is used za in classification. A multi-swarm version is proposed in which several swarms cooperate in resolution, to improve initial results.

There are several jobs that perform this attribute selection using a binary PSO. It is a normal application of the algorithm described in the Ap- Given 2.2.5, but the fitness of the particle is calculated by applying of another “underlying” classifier (not based on PSO), which considers only the attributes selected by the particle. Some of the work in this field are the following:

In [Liu et al., 2006], PSO is used to determine the set of attributes butos and the centers and number of hidden neurons of a Base Network Radial (RBFNN) used in classification.

In [Correa et al., 2006] a specific version of PSO dis- creto for attribute selection, in which the particles encodeattribute index combinations without repetition; a key is used Bayesian sifier to determine the optimal classification. In [Marinakis et al., 2008], the binary version of PSO is used as as introduced in [Kennedy and Eberhart, 1997] for selection of attributes, combined with a K-NN classifier for classification. In [Chuang et al., 2008] a new binary version of PSO is proposed called Improved Binary PSO, (IBPSO), in which the best position the swarm is reset if it does not vary by a certain numberof iterations. This algorithm is applied for selection ofattributes and subsequent classification using the 1-NN rule.In [Melgani and Bazi, 2008] or [Huang and Dun, 2008], a Conventional PSO to simultaneously optimize the set of attributes butos and parameters of a vector support machine (SVM).

IV. Particle swarms for multimodal optimization

Multimodal functions are defined as those that have more than a local optimum, and may also have one or more global optimum.If there is only one global optimum, the problem is only to avoid premature convergence to a local optimum; this can be achieved by the introduction of diversification mechanisms.

There are two ways to eliminate these drawbacks:

The first is to transform fitness function whenlocates an optimum (possibly local). This is done through a transformation of “flattened” that makes the optimum disappear, of so that the swarm can continue to locate the rest. The solution is the sequence of optimal found.

The second is more relevant to our study, since it consists in trying to find all the optimum ones simultaneously; for it The swarm needs to be partitioned so that someof the particles are centered in some areas of the search space, and others in different areas. The techniques used are usually known with the name of "niche techniques" (niching). There are some examples of use of these techniques in PSO:

• In some works the so-called Niching PSO has been developed [Brits et al., 2002], [Nickabadi et al., 2008]. In this, theswarm into "sub-swarms", which follow different leader particlesdiffferent; in this way, they seek optimal positions around different point. As the division of the swarm occurs,there is the possibility of sub-swarm fusion when detecting They are searching around the same goal.The division and merger processes are carried out dynamically, based on measurements performed during the iteration of the algorithm.

On the other hand, in the absence of prior information on the location of the optimal settings for the fitness function, the partition should be adjusted dynamics, either as a function of some condition imposed on the particlesthat form a class, either simply by recalculating it iteration by iteration. This obviously adds a computational cost to the standard PSO algorithm.

Third, the way the space is distributed obeys criteria of location; especially when conceiving the idea of "sub-swarms," measure of the “size” of each sub-swarm to determine whether to cause division or merger.

These three concepts are applicable to the algorithm that I propose, since they share the common goal of finding good positions by simultaneously in scattered regions of the search space.

Reference

1. Aha, D. W. (1992). Tolerating noisy, irrelevant and novel at- tributesininstance-basedlearningalgorithms.*Int.J.Man-Mach.Stud.*, 36(2):267-287.
2. Aha, D. W., Kibler, D., and Albert, M. K. (1991).
3. Instance-basedlearningalgorithms.*Mach.Learn.*,6(1):37-66.
4. Alcala-Fernández,J.,Garcia,S.,Berlanga, F.,Fernández,A.,Sanchez,L.,delJesus,M.,andHerrera,F.(2008).Keel: A data mining software tool integrating genetic fuzzy systems. *Genetic andEvolvingSystems,2008.GEFS2008.3rdInternationalWorkshopon*, pages83-88.
5. Blachnik, M. and Duch, W. (2008). *Prototype rulesfromSVM*,volume80/2008of*StudiesinComputationalIntelligence*, pages163-182.SpringerBerlin/Heidelberg.
6. Blackwell,T.(2007).*ParticleSwarmOptimizationinDy- namicEnvironments*,pages29-49.StudiesinComputationalIntelligence. Springer Berlin /Heidelberg.
7. Cano, J., Herrera, F., and Lozano, M. (2003). Using evolutionary algorithms as instance selection for data reduction in kdd: anexperimentalstudy. *IEEETransactionsonEvolutionaryComputation*, 7(6):561-575.
8. Prism: An algorithm for in- ducingmodularrules.*InternationalJournalofMan-MachineStudies*, 27(4):349-370.
9. Cervantes,A.,Isasi,P.,andGalván,I.(2005a). Binary Particle Swarm Optimization in classification. *Neural Network World*,15(3):229-241.
10. Duch, W. and Grudzinski, K. (2001). Pro- totype based rules-a new way to understand the data. In *Proceedings of International Joint Conference on Neural Networks, 2001 (IJCNN’01).*, volume3,pages1858-1863vol.3.
11. Eberhart, R. and Shi, Y. (2000). Comparing iner- tia weights and constriction factors in particle

swarm optimization. In *Proceedingsof2000CongressonEvolutionaryComputation,2000.*,volu- me1,pages84-88vol.1.

12. Holden, N. P. and Freitas, A. A. (2007). A hybrid pso/acoalgorithmforclassification.In*GECCO'07:Proceedingsof2007 GECCOconferencecompaniononGeneticandevolutionarycomputation*, pages2745-2750,NewYork,NY,USA.ACM.
13. [Holland,1976]Holland,J.(1976).Adaptation.*Progressintheoreticalbio- logy*,IV:263-293.
14. Jin, Y. (2006). *Multi-Objective Machine Learning*, volume 16/2006of*StudiesinComputationalIntelligence*.SpringerBerlin/Hei- delberg.
15. Jin, Y. (2007). Pareto-based multi-objective machine learning. In*7thInternationalConferenceonHybridIntelligentSystems,2007.HIS 2007.*,pages2-2.
16. Nickabadi, A., Ebadzadeh, M., and Safabakhsh, R. (2008). Dnpso: A dynamic niching particle swarm optimizer for multi- modaloptimization.In*ProceedingsofIEEECongressonEvolutionary Computation,2008(CEC2008).(IEEEWorldCongressonComputatio- nalIntelligence).*,pages26-32.
17. Omran, M. G., Salman, A. A., and Engelbrecht, A. P. (2006). Dynamic clustering using particle swarm optimization with ap- plicationinimagesegmentation.*PatternAnal.Appl.*,8(4):332-344.
18. O'Neill,M.andBrabazon,A.(2008).Self- organisingswarm(soswarm).*SoftComput.*,12(11):1073-1080.
19. [Poli and Broomhead, 2007] Poli, R. and Broomhead,D. (2007). Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation. In *GECCO '07: Procee- dingsof9thannualconferenceonGeneticandevolutionarycomputation*, pages134-141,NewYork,NY,USA.ACM.
20. Powell,M.J.D.(1987).*Radialbasisfunctionsformultiva- riableinterpolation:areview*,pages143-167.ClarendonPressInstitute of Mathematics and its Applications. Clarendon Press, New York, NY, USA.
21. Sousa, T., Silva, A., and Neves, A. (2004). Particle swarm based data mining algorithms for classification tasks. *Parallel Comput.*,30(5-6):767-783.
22. Suganthan,P.Ñ.(1999).Particleswarmoptimiserwith neighbourhoodoperator.In*ProceedingsofIEEECongressonEvolutio- naryComputation(CEC)*,pages1958-1962.
23. Zhang,L.,Zhou,C.,Liu,X.,Ma,Z.,Ma,M.,andLiang,
24. Y. (2003). Solving multi objective optimization problems using particle swarmoptimization.In*ProceedingsofIEEECongressonEvolutionary Computation2003(CEC2003)*,pages2400-2405.
25. Zhang, Q., Li, X., and Tran, Q. (2005). A modified par- ticleswarmoptimizationalgorithm.In*Proceedingsof2005International ConferenceonMachineLearningandCybernetics,2005.*,volume5,pages 2993-2995Vol.5.
26. Zhen, L., Wang, L., Wang, X., and Huang, Z. (2008). A novel pso-inspired probability-based binary optimization algorithm. In *ProceedingsofInternationalSymposiumonInformationScienceandEn- gieering,2008.ISISE'08.*,volume2,pages248-251.