Chemometrics and Intelligent Laboratory Systems 149 (2015) 153–165

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Particle swarm optimization (PSO). A tutorial

Federico Marini [a,*], Beata Walczak [b]

[a] Dept. of Chemistry, University of Rome "La Sapienza", P.le Aldo Moro 5, I-00185 Rome, Italy
[b] Dept. of Analytical Chemistry, University of Silesia, 9 Szkolna St., 40006 Katowice, Poland

ARTICLE INFO

Article history:
Received 28 April 2015
Received in revised form 25 August 2015
Accepted 26 August 2015
Available online 12 September 2015

Keywords:
Particle swarm optimization (PSO)
Variable selection
Warping algorithms
Continue and discrete optimization
Swarm intelligence

ABSTRACT

Swarm-based algorithms emerged as a powerful family of optimization techniques, inspired by the collective behavior of social animals. In particle swarm optimization (PSO) the set of candidate solutions to the optimization problem is defined as a swarm of particles which may flow through the parameter space defining trajectories which are driven by their own and neighbors' best performances. In the present paper, the potential of particle swarm optimization for solving various kinds of optimization problems in chemometrics is shown through an extensive description of the algorithm (highlighting the importance of the proper choice of its metaparameters) and by means of selected worked examples in the fields of signal warping, estimation robust PCA solutions and variable selection.

1. Introduction: the concept of swarm intelligence and its artificial implementations

Inspired by studies in neurosciences, cognitive psychology, social ethology and behavioral sciences, the concept of swarm intelligence (SI) [1] was introduced in the domain of computing and artificial intelligence in 1989 [2] as an innovative collective and distributed intelligent paradigm for solving problems, mostly in the domain of optimization, without centralized control or the provision of a global model. In this context, the starting point was the realization that many of the limitations and drawbacks observed in the framework of artificial intelligence may be ascribed to the assumption that intelligence be based in individual minds, neglecting the fundamental contribution of sociality. As a consequence, the concept of swarm – inspired by the collective behavior of social animals, such as bird, fishes, ants, bees, and termites – was introduced to investigate multi-agent distributed systems of intelligence. A swarm is a population of homogenous, simple agents performing rather elementary tasks and interacting locally among themselves, and their environment, with no central control: collective behavior emerges as a consequence of self-organization and local (and partially stochastic) interactions. Indeed, although these agents (swarm individuals) are relatively unsophisticated with limited capabilities on their own, the ability of achieving the final goal of problem solving is provided by their behavioral and interaction patterns through information sharing. Accordingly, swarm-based algorithms have recently emerged as a family of nature-inspired, population-based algorithms that are capable of producing low cost, fast, and robust solutions to several complex problems [1,3–5]. The concept of nature-inspired population-based algorithms is not new to chemometricians, as it covers methods, which have been widely used for many years by the community, such as genetic algorithms [6] or simulated annealing [7].

In this context, taking inspiration from various natural systems, different algorithms have been proposed in the literature, the most widely used of which are ant colony optimization (ACO) and particle swarm optimization (PSO), which are also representative of the two main ways interactions among agents may be defined, indirectly and directly, respectively. Other algorithms belonging to this group are, for instance, the bee algorithm, cuckoo search, harmony search, the flower algorithm, the bat algorithm, and the firefly algorithm [8].

Ant colony optimization is a probabilistic optimization technique, which is applicable where the task may be expressed as that of finding the best path along a graph [9,10]. Its inspiration stems from the wandering behavior of ants seeking a path between their colony and a source of food. In ACO, the artificial ants iteratively build solutions to the problem at hand by moving from a candidate state to another and it selects the successive step, among all the possible ones based on the combination of two factors: the "attractiveness" of the move, which is a metaheuristic parameter determining the a priori desirability of the state transition (and usually it is inversely related to the distance to the destination point), and the "pheromone trail", which indirectly provides the social interaction among the agents. Indeed, analogously to what happens in the behavior of real ants, which, along their wander in search of food, deposit pheromones on the ground, so that future members of the colony will choose with higher probability paths that are marked by stronger concentrations of these substances, the a

* Corresponding author.
E-mail address: federico.marini@uniroma1.it (F. Marini).

http://dx.doi.org/10.1016/j.chemolab.2015.08.020
0169-7439/

posteriori optimality (fitness) of a solution found by an artificial ant will be accompanied by an increase of the pheromone trail associated to that direction.

On the other hand, particle swarm optimization [11] was inspired by the information circulation and social behavior observed in bird flocks and fish schools, and it is a global optimization algorithm which is particularly suited to solve problems where the optimal solution is a point in a multidimensional space of the parameter (real-valued optimization). Inspiration from the natural analogues, i.e. schooling or flocking, translates to the property that agents (the particles) are characterized not only by a position, but also a velocity and are allowed to move around in the search space. Here, differently than for ACO, the social interaction is direct, as the movement of each particle is not only influenced by its own memory (its best solution found so far) but it is also directed towards the best position found by other agents, be they a subset of particles or the whole swarm. Due to its metaheuristic nature, which allows obtaining solutions also for non-differentiable problems which may be irregular, noisy or dynamically changing with time, PSO algorithm has found a wide range of application in many domains of computer science and applied mathematics, such as for the calculation of neural network weights [12,13], time series analysis [14], business optimization [15] and many others. However, despite its promising characteristics, it has found so far only a limited popularity in chemometrics, where only a few examples of applications are reported [16–27].

In this framework, the present paper aims at presenting the potential of using particle swarm optimization in chemometrics, through providing a tutorial introduction to the subject, together with a series of examples taken from different domains.

2. Theory

2.1. PSO within the general background of optimization problems

To allow a better comprehension of the PSO algorithm and in order to provide a general framework, the basic concepts in optimization will be reviewed in this section. In science, the term optimization refers to the process of identifying the best element (according to one or more user-specified criteria) among a set of possible alternatives [28]. In mathematical language, this is usually accomplished by defining the goal in terms of a parameterized function f, so that optimization boils down to finding the values of the parameters, which minimize or maximize (depending on the specific task) f itself. Since minimization of f corresponds to maximization of $-f$, in the remainder of the section we will discuss only the maximization task, without any loss of generality. By assuming that the function f, which is usually referred to as the "objective function", depends on D parameters, the maximization task may be defined as:

$$\begin{array}{l}\text{Given} \quad f : \mathbb{R}^D \rightarrow \mathbb{R} \\ \text{Find} \quad \mathbf{x}_{opt} \,|\, f(\mathbf{x}_{opt}) \geq f(\mathbf{x}) \quad \forall \mathrm{x} \in \mathbb{R}^D.\end{array} \tag{1}$$

The D-dimensional domain of the function $\mathbb{R}^D$ is called the search (or parameter) space and each of its points, characterized by the vector of coordinates $\mathbf{x}$, represents the candidate solutions of the problems, $\mathbf{x}_{opt}$ being the optimal one, i.e. the one which maximizes f. The value of f, which quantifies the degree of optimality of a particular set of parameters for the specific task, is often referred to as the *fitness*.

While maximization of the objective function is relatively straightforward when f is known analytically and differentiable at all points, in most real-world applications the latter conditions are rarely met. Indeed, the objective function is often defined implicitly (and may not be differentiable at some points of the domain) so that it may only be possible to evaluate the fitness value corresponding to a particular set of parameters (the candidate solution). Accordingly, in such cases the optimization task corresponds to an extensive search in the parameter space, which involves the (usually iterative) exploration of various candidate solutions through the evaluation of their fitness until the one corresponding to the maximum value is found [29,30].

In this context, particle swarm optimization, originally developed by Kennedy and Eberhart [11], is a meta-heuristic global optimization method, which belongs to the family of algorithms based on the concept of swarm intelligence. In analogy to the behavior of bird flocks and fish schools, in PSO the set of candidate solutions to the optimization problem is defined as a swarm of particles which may flow through the parameter (search) space defining trajectories which are driven by their own and neighbors' best performances. Indeed, unlike what happens with other nature-inspired algorithms, evolution is based on cooperation and competition among individuals through generations (iterations): the flow of information among particles, which can be limited to a local neighborhood (partial PSO) or extended to the whole swarm (global PSO) is an essential characteristic of the algorithm.

2.2. The basic PSO algorithm

In PSO, each candidate solution is called a "particle" and represents a point in a D-dimensional space, if D is the number of parameters to be optimized. Accordingly, the position of the ith particle may be described by the vector $\mathbf{x}_i$:

$$\mathbf{x}_i = [x_{i1} x_{i2} x_{i3} \ldots x_{iD}] \tag{2}$$

and the population of N candidate solutions constitutes the swarm:

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}. \tag{3}$$

In searching for the optimal solution of the problem, the particles define trajectories in the parameter space (i.e., iteratively update their positions) based on the following equation of motion:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{4}$$

where t and $t+1$ indicate two successive iterations of the algorithm and $\mathbf{v}_i$ is the vector collecting the velocity-components of the ith particle along the D dimensions. The velocity vectors govern the way particles move across the search space and are made of the contribution of three terms: the first one, defined the *inertia* or *momentum* prevents the particle from drastically changing direction, by keeping track of the previous flow direction; the second term, called the *cognitive component*, accounts for the tendency of particles to return to their own previously found best positions; the last one, named the *social component*, identifies the propensity of a particle to move towards the best position of the whole swarm (or of a local neighborhood of the particle, depending on whether a global or partial PSO is implemented). Based on these considerations, the velocity of the ith particle is defined as:

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + c_1(\mathbf{p}_i - \mathbf{x}_i(t))\mathbf{R}_1 + c_2(\mathbf{g} - \mathbf{x}_i(t))\mathbf{R}_2 \tag{5}$$

where $\mathbf{p}_i$ is the so called "personal best" of the particle, i.e. the coordinates of the best solution obtained so far by that specific individual, while $\mathbf{g}$ is the "global best", i.e., the overall best solution obtained by the swarm. The acceleration constants c_1 and c_2, which are real-valued and usually in the range $0 \leq c_1, c_2 \leq 4$, are called "cognitive coefficient" and "social coefficient", and modulate the magnitude of the steps taken by the particle in the direction of its personal best and global best, respectively. On the other hand, $\mathbf{R}_1$ and $\mathbf{R}_2$ are two diagonal matrices of random numbers generated from a uniform distribution in [0,1], so that both the social and the cognitive components have a stochastic influence on the velocity update rule in Eq. (5). Accordingly, the trajectories drawn by the particles are semi-random in nature, as they derive from the contribution of systematic attraction towards the personal and global best solutions and stochastic weighting of these two acceleration terms.

1. Initialization. For each of the N particles:
   a. Initialize the position $\mathbf{x}_i(0)$ $\forall i \in 1{:}N$
   b. Initialize the particle's best position to its initial position $\mathbf{p}_i(0)=\mathbf{x}_i(0)$
   c. Calculate the fitness of each particle and if $f(\mathbf{x}_j(0)) \geq f(\mathbf{x}_i(0))$ $\forall i \neq j$ initialize the global best as $\mathbf{g}=\mathbf{x}_j(0)$
2. Until a stopping criterion is met, repeat the following steps:
   a. Update the particle velocity according to equation (5):
      $$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + c_1\left(\mathbf{p}_i - \mathbf{x}_i(t)\right)\mathbf{R}_1 + c_2\left(\mathbf{g} - \mathbf{x}_i(t)\right)\mathbf{R}_2$$
   b. Update the particle position according to equation (4):
      $$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1)$$
   c. Evaluate the fitness of the particle $f(\mathbf{x}_i(t+1))$.
   d. If $f(\mathbf{x}_i(t+1)) \geq f(\mathbf{p}_i)$, update personal best: $\mathbf{p}_i=\mathbf{x}_i(t+1)$
   e. If $f(\mathbf{x}_i(t+1)) \geq f(\mathbf{g})$, update global best: $\mathbf{g}=\mathbf{x}_i(t+1)$
3. At the end of the iterative process, the best solution is represented by $\mathbf{g}$.

Fig. 1. Scheme of the basic PSO algorithm.

Altogether, the iterative process described by Eqs. (4) and (5) is repeated until a stopping criterion – which may be, e.g., a prespecified total number of iterations, a maximum number of iterations since the last update of global best or a predefined target value of the fitness – is met.

A schematic description of the basic PSO algorithm is reported in Fig. 1.

2.2. A deeper look into the algorithm and its meta-parameters

Particle swarm optimization, as many other metaheuristic approaches, has several metaparameters that govern its behavior and efficiency in optimizing a given problem, specifically as the search behavior of particles, the influence of control parameters on the performances, and the convergence properties of the algorithm are concerned. Here it must be stressed that, in PSO, as in all stochastic evolutionary algorithms, the term convergence may refer to two different scenarios: convergence as the limit of a series of solution (where, for instance, all or most of the particles, reach the same point in the search space, which may not necessarily be the optimum), indicating the algorithm stability, and convergence to a local (or the global) optimum of the problem, which may be achieved by one or more particles (through personal bests or global best), irrespective of the overall behavior of the swarm. Accordingly, the understanding of the role of these metaparameters (such as the acceleration constants, the initialization strategies or the neighborhood definition) and of their impact on the final results is of critical importance for the design of an efficient optimization algorithm [1,31].

2.2.1. Position and velocity initialization

As shown in Fig. 1, being an iterative algorithm, PSO requires an initial estimate of the particles' positions and velocities and the choice of the way these two entities are initialized plays a determinant role in defining what is the probability that particles travel outside the boundaries of the search space and, as a consequence, in affecting the convergence properties of the solution. In particular, there is a general agreement in the literature – and it is also the authors' opinion – that initializing the particles' positions, so that they cover as uniformly as possible the search space, is the best option:

$$x_{ij}(0) \sim U\left(x_{j,\min}, x_{j,\max}\right) \tag{6}$$

where $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper boundaries of the jth dimension of the search space, respectively and U indicates a uniform distribution. This choice ensures good initial coverage of the parameter space and fosters the exploration ability of the algorithm, resulting in a relatively high convergence speed [32].

On the other hand, the selection of the best initialization strategy for the velocities is still debated. Indeed, many authors suggest using the same random initialization as for the particles, i.e. to draw the velocity component from a uniform distribution covering the entire domain of the search space, analogously to what was described in Eq. (6). However, while this choice results in a higher initial diversity of the swarm (due to the dominance of the inertia term and, as a consequence, to larger initial step sizes), it also may increase the tendency of the particles to violate the boundaries of the search space, causing a divergence of the algorithm ("velocity explosion") [33]. Although the latter is a general risk and different modifications to the basic PSO algorithm have been proposed in the literature to overcome that drawback (see also Section 2.2.3), it may still be a good option to set the initial velocities to zero or to very small random numbers, as the exploration ability is still guaranteed by the choice of the initial positions in Eq. (6) [34].

2.2.2. Choice of the acceleration constants c_1 and c_2

As evident from Eq. (5), the values of the acceleration constants c_1 and c_2 govern the extent to which the particles move towards the individual and global best particle, modulating the relative contributions of the social and cognitive terms. Different authors (see, e.g. [35] or [1]) have investigated the effect of these coefficients on the particles' trajectories and on the convergence properties of the algorithm, showing that as the acceleration constants are increased, the frequency of oscillation of the particle around the optimum increases while smaller values result in sinusoidal patterns. In general, it has been shown that the conditions:

$$c_1 = c_2 = 2 \tag{7}$$

work well for most of the applications [11].

2.2.3. Avoiding the velocity explosion

One of the main characteristics of the PSO algorithm, as expressed in Eq. (5), is that the particle moves by a step which, along each direction of the search space, is different at any iteration. In particular, if the acceleration constants be fixed to the values reported in Eq. (7), careful inspection of Eq. (5) shows how – since both the terms $c_1\mathbf{R}_1$ and $c_2\mathbf{R}_2$ would be uniformly distributed in [0,2] with average value

equal to 1 – there would be a 50% probability that a particle overflies its momentary target, i.e., that one may expect that, on average, an individual would move beyond its temporary target in half of the iterations. Accordingly, since velocities are updated stochastically, it may happen that the trajectory of a particle expands into wider and wider cycles across the parameter space, crossing the boundaries of the search space and, eventually, approaching infinity (situation that, in Section 2.2.1, was already called "velocity explosion"). To overcome this problem, different strategies have been proposed in the literature to damp the oscillations of the particles across the search space and to facilitate convergence. In particular, the two approaches that are most commonly adopted are the so-called velocity clamping and the introduction of an inertia weight [36].

By expression velocity clamping, one indicates a technique by which a limit is posed to the maximum velocity of each particle, in order to prevent it from moving too far beyond the limits of the search space. Accordingly, a velocity threshold is introduced in the algorithm, so that

$$\begin{array}{l} \text{if } v_{ij}(t+1) > v_j^{\max} \quad \text{then } v_{ij}(t+1) = v_j^{\max} \\ \text{if } v_{ij}(t+1) < -v_j^{\max} \quad \text{then } v_{ij}(t+1) = -v_j^{\max} \end{array} \tag{8}$$

where $v_{ij}(t+1)$ is the velocity component of the ith particle along the jth direction at the $(t+1)$th iteration of the algorithm and $v_j^{\max}$ is the maximum absolute value of velocity allowed along the same jth direction in the parameter space. The velocity clamping strategy has proved to be rather effective in damping the oscillations by providing a good balance between global exploration and local exploitation [1]. However, its main drawback is that the choice of the best value of $v_j^{\max}$ is strongly problem-dependent and, moreover, finding the accurate value for the problem at hand is very critical and not simple, so that it is difficult to find a generally applicable criterion or even an empirical rule of thumb. What many authors agree on, anyway, is that the value of $v_j^{\max}$ should be chosen in the interval:

$$v_j^{\max} = k\frac{(x_{j,\max} - x_{j,\min})}{2} \quad k \in (0,1] \tag{9}$$

where $x_{j,\min}$ and $x_{j,\max}$ have the same meaning as in Eq. (6) and k is a constant.

Another way to avoid the swarm divergence due to the velocity explosion is the introduction of an inertia weight $\omega(t+1)$ to control the momentum of the particle by modulating the contribution of the velocity at the previous iteration to the definition of the particle's actual velocity [37]. Accordingly, the velocity update rule as described in Eq. (5) is modified to:

$$\mathbf{v}_i(t+1) = \omega(t+1)\mathbf{v}_i(t) + c_1(\mathbf{p}_i - \mathbf{x}_i(t))\mathbf{R}_1 + c_2(\mathbf{g} - \mathbf{x}_i(t))\mathbf{R}_2 \tag{10}$$

so that one may argue that a large value of the inertia weight (>1) favors global exploration, while with a small one (<1) local exploitation dominates. Although inertia weight was originally introduced as a constant by Shi and Eberhart [37], several studies have shown how a dynamical adjustment of its value may significantly improve the convergence properties of PSO [38–42]. A survey of the most commonly adopted dynamical adjustment strategies for inertia weight is reported in Table 1.

Implementation of a linearly decreasing inertia weight has shown to provide very good results in many real world applications and several authors suggest to adopt the combination $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$ to achieve the best performances. In general, a comparative study carried out by Bansal et al. [38] on a set of benchmark functions for optimization has shown how random inertia weight is better if faster convergence is desired, while chaotic decreasing inertia weight is the most consistent (leading to the lowest average error over a set of 30 repeated simulations). On the other hand, constant and linearly decreasing inertia weight strategies were the ones leading to overall lowest errors.

Table 1
Dynamic adjustment strategies for inertia weight.

Strategy	Definition of inertia weight	Reference
Constant inertia weight	$\omega(t) = \omega = const$	[37]
Random inertia weight	$\omega(t) = 0.5 + \frac{r}{2} \quad r \sim U(0,1)$	[39]
Linearly decreasing inertia weight	$\omega(t) = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} t$	[40]
Global–local best inertia weight	$\omega_{ij}(t) = 1.1 - \frac{p_{ij}(t)}{g_j(t)}$	[41]
Chaotic descending inertia weight	$\omega(t) = (\omega(0) - \omega(t_{\max}))(\frac{t_{\max} - t}{t_{\max}}) + \omega(t_{\max})z$ $z = 4r(1-r)$ with $r \sim U(0,1)$	[42]
Chaotic random inertia weight	$\omega(t) = 0.5r_1 + 0.5z$ $z = 4r_2(1-r_2)$ with $r_1, r_2 \sim U(0,1)$	[42]

2.2.4. Network topology

The basic PSO algorithm, which defines the social term in the velocity update rule (Eq. (5)) as a function of the position of the best overall particle in the swarm (the global best $\mathbf{g}$), may be easily trapped in a local optimum. Indeed, fast convergence is often achieved as all the particles tend to be attracted simultaneously to the portion of the search space where the global best is. However, if the global optimum is not close to the best particle, this characteristic may hinder the possibility of the swarm to explore other areas [43].

One way of limiting the probability of a premature convergence to local optima is to define the social component of the velocity update equation not in terms of the global best $\mathbf{g}$ but just based on the best known position $\mathbf{l}$ (local best) of a sub-swarm "around" the particle that is moved (its neighborhood). Here it must be stressed that, even if the neighborhood could be defined also geometrically, e.g. as the k nearest particles, it is, more often, a social one, defined on the basis of a topology which depends only on the particle index i (in practice, the neighborhood of each individual is defined – at the very beginning – as a list of particles which does not change with the iterations). The advantage of a local best swarm (partial PSO) is that while neighbors are closely connected, the individuals that are topologically distant are also relatively independent of one another, so they may search different portions of the parameter space or explore different local optima without the overall swarm being trapped in any of them: as a consequence, the swarm usually converges at a slower rate but there is a higher probability that it eventually locates the global optimum.

2.2.5. Swarm population

The population of the swarm is another factor that may have an impact on the performances of the PSO algorithm. Indeed, on one hand a larger population increases the diversity of the swarm and its exploration ability but on the other hand it may also increase the probability of premature convergence and the computational efforts. However, in most cases it has been demonstrated that when the number of individuals is larger than 50, PSO is not sensitive to the size of the population [1].

2.3. Discrete PSO

Even if PSO was originally designed and developed for optimization problems involving real-valued parameter spaces, a number of variants have been proposed in the literature, which are able to deal also with discrete domains. One important example is represented by the binary PSO algorithm [44], which may be used, for instance, to operate variable selection (see Section 6). In binary PSO, the parameters that need to be optimized (and hence the candidate solutions, i.e. the particles' positions) may assume only the value 0 or 1 so, at each iteration, the vector $\mathbf{x}_i$ is binary-coded. On the other hand, the velocity vector – which is

always updated according to Eq. (5) or (10) – is still real-valued and, after a suitable scaling transformation to adjust its range to [0,1]:

$$v'_{ij} = \frac{1}{1+e^{-v_{ij}}} \tag{11}$$

its components are interpreted as the probabilities that the various bits corresponding to the parameters will change their status. Accordingly, if $v_{ij}' = 0.80$, there will be an 80% probability that the jth component of the ith particle's position will change its value. In mathematical terms, this concept translates to the following equation of motion, which, for binary PSO, substitutes Eq. (4):

$$\begin{aligned} x_{ij}(t+1) &= |x_{ij}(t)-1| \quad \text{if } v'_{ij}(t+1)>s \\ x_{ij}(t+1) &= x_{ij}(t) \quad \text{if } v'_{ij}(t+1)<s \end{aligned} \qquad s \sim U(0,1) \tag{12}$$

where s is a random number uniformly distributed in [0,1].

3. A first toy example using a standard benchmark function for optimization

In order to show the main characteristics of the PSO algorithms, especially in terms of how the swarm evolves over time, and of the effect of varying some of the meta-parameters, in the present section the use of PSO algorithm to look for the minimum of a classical benchmark function for optimization problems (here considered in two dimensions) is discussed. The function is Ackley's one [45,46]:

$$f(x_1,x_2) = -20e^{-0.2\sqrt{0.5\sum_{i=1}^{2} x_i^2}} - e^{\sqrt{0.5\sum_{i=1}^{2}\cos(2\pi x_i)}} + 20 + e \tag{13}$$

which is widely used for testing optimization algorithms and it is characterized by being multimodal with many local optima, regularly spaced along the search range, but a single global minimum localized in (0,0), as shown in Fig. 2.

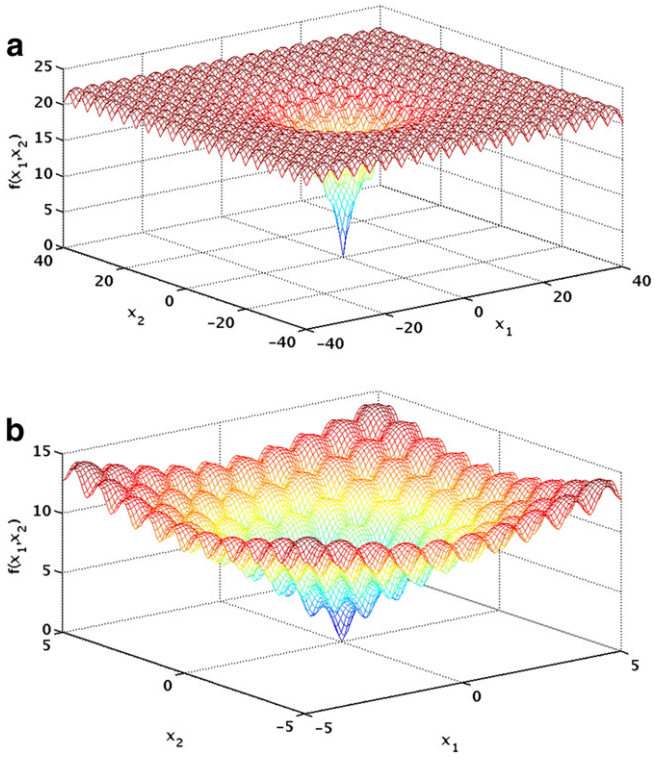

Fig. 2. Graphical representation of the Ackley function in two dimensions over the whole search space (a) and zoomed in the proximity of the global minimum.

Due to its multimodality, this function may pose a problem for some optimization algorithms, especially those based on gradient descent, as they can get stuck in one of the many local minima.

Particle swarm optimization was then used to look for the function's global minimum and accordingly, as described in Section 2, the fitness of each candidate solution was defined as minus the value of Ackley function in that point. A swarm of 50 particles was initially used and the trajectories were regulated by the laws of motion reported in Eqs. (4) and (10) with a constant inertia weight set to 0.7 and a stopping criterion of 500 iterations. Starting velocities were set to zero, while positions were initialized randomly in the search range of the function ([−32, +32] for both axes). The evolution of the swarm under these conditions is shown in Fig. 3, where the candidate solutions at selected number of iterations are reported together with the target function.

As shown in the figure, at the earlier stages of the evolution, exploration tendency prevails and the particles span a great portion of the parameter space, sometimes also crossing the borders of the search space; however, due to the inertia weight, no divergence of the algorithm is observed. With the increase in the number of iterations, more and more particles flow towards the global optimum, which is reached quite early. Already after 50 iterations, the global best of the swarm is very close to the global minimum of the function and some particles are located in a region near to it. After 100 iterations, almost all of the particles are concentrated in the proximity of the global optimum of the function, even if still some candidate solution lies far away, describing a trajectory, which keeps exploring different parts of the parameter space. At around 350 iterations, all the particles of the swarm have reached the global minimum of (0,0).

The same example may be used to show the effect of the number of particle population on the swarm evolution and on the final accuracy of the optimization. To this purpose, the results obtained with a population of 50 particles were compared to those corresponding to a swarm of 20 or 100 individuals. In particular, Fig. 4a–c shows the fitness values of the individual particles along the whole course of their evolution, while Fig. 4d describes the changes in the fitness of the global best of the swarm at each iteration.

It was observed that all the three populations converge to the same optimal solution, but the smaller one requires a higher number of iterations (370) than the others, which need 349 and 354 iterations, respectively. Moreover, it can be observed how, by increasing the number of particles in the swarm, its diversity and exploratory ability remain significantly higher for a longer number of iterations. In general, also according to what was already discussed in Section 2.2.5, it can be affirmed that 50 particles are a good compromise.

4. A first chemometric example: using PSO for signal alignment

In order to show the potential of PSO algorithms in chemometrics, some examples taken from different domains will be discussed in this and the following sections. As a first example, the possibility of using PSO in the framework of signal warping (alignment) is presented.

Instrumental signals may be vulnerable to different sources of instrumental variations, which may result in misalignment and peak shifts [47]. This is a major problem in the data analysis of chromatographic or electrophoretic signals, but it may also affect other techniques, such as nuclear magnetic resonance profiles (^{1}H NMR). Accordingly, to solve the problem of misalignment, approaches based on the synchronization of entire signals (or portions of it) by translation or warping have been presented in the literature, the most popular or which being Dynamic Time Warping (DTW) [48], Correlation Optimized Warping (COW) [49], Parametric Time Warping (PTW) [50], Local Warping [51], Automated Alignment (AA) [52], and iCoshift [53,54]. In the present paper, the possibility of adopting a parametric time warping approach [50] which relies on the use of the Particle Swarm Optimization (PSO) algorithm for the computation of the warping function is discussed. Parametric time warping operates by explicitly

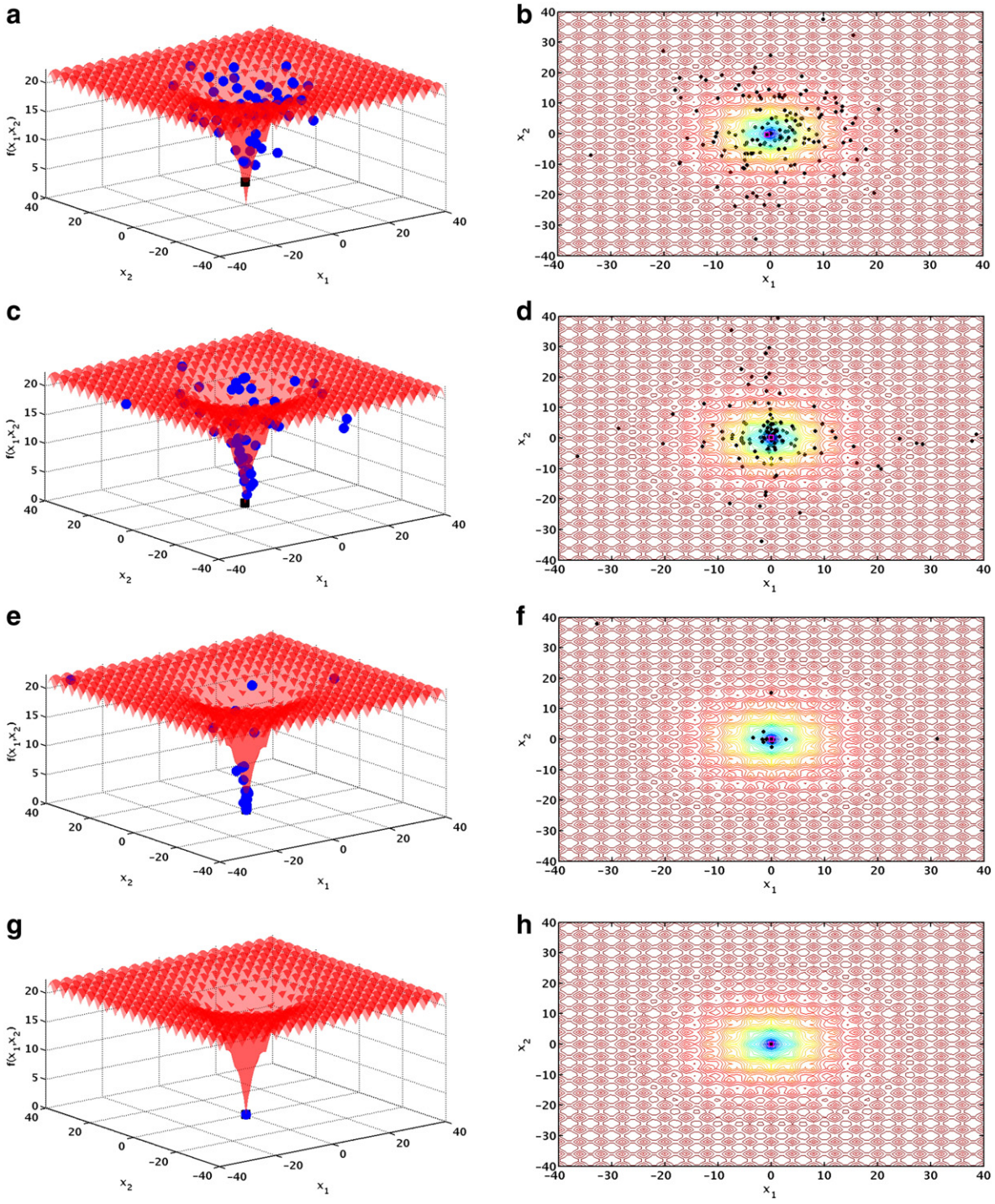

Fig. 3. Graphical representation of the evolution of the particle positions (circles) and global best (square) when a 50 particle swarm is applied to the search of the minimum of Ackley function. Results are displayed as surface and contour plots for: (a,b) 10 iterations, (c,d) 50 iterations, (e,f) 100 iterations, and (g,h) 350 iterations.

modeling the functional relation between the time scales of the original and the warped chromatograms according to:

$$\mathbf{time}_{new} = f(\mathbf{time}_{old}). \tag{14}$$

This functional relation is expressed in a parametric way and the optimal values of the parameters are chosen as those, which maximize the correlation between the *ith* signal and the selected standard. In particular, in the proposed method the transformed time axis '$\mathbf{time}_{\mathrm{new}}$' of the warped signal is expressed as a linear combination of an opportune number of radial basis functions (RBF) uniformly distributed along the time axis, **T**:

$$\mathbf{time}_{new} = [\varphi_1 \varphi_2 \ldots \varphi_k] \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} \tag{15}$$

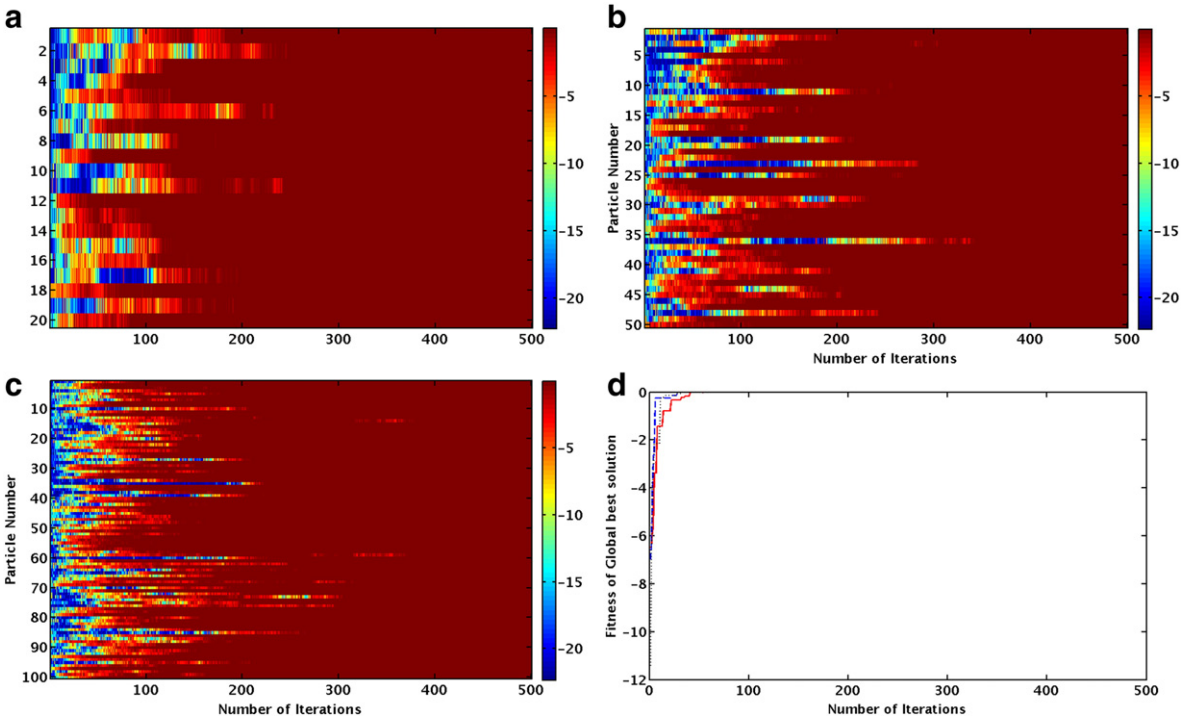

Fig. 4. Effect of swarm population on the convergence properties of PSO applied to the search of the minimum of Ackley function. Individual fitness for each particle at each iteration is displayed as heatmap for a swarm of 20 (a), 50 (b) and 100 particles (c). The evolution of the global best fitness is reported as line plot in panel (d). Legend: continuous red line — 20 particles, dotted black line — 50 particles, dashed blue line — 100 particles.

where column vector φ_k represents an output of the *kth* radial basis function for all n time points:

$$\varphi_k = e^{-\frac{(\mathbf{t}-\tau_k)^2}{\sigma^2}} \tag{16}$$

τ_k and σ being the center and the width of the RBF, respectively.

In this framework, particle swarm optimization (PSO) algorithm is proposed as an optimization method for the estimation of the best set of coefficients $c_1, c_2, \ldots, c_k$, i.e. of the set of parameters which maximizes correlation between the warped signals.

4.1. Data

In order to demonstrate the potential and the performances of the proposed alignment algorithm based on particle swarm optimization on 1- and 2-dimensional chromatographic signals, three different data sets were analyzed.

4.1.1. Data set 1

Data set 1 contains the pair of chromatograms, which was used to demonstrate the performance of Correlation Optimized Warping (COW) [49], Automated Alignment (AA) [52] and Fuzzy Warping (FW) [55]. These chromatograms have different lengths and contain different number of peaks (with different height and shape). In our study, the shorter profile is selected as a target, T. A detailed description of these signals can be found in Ref. [49].

4.1.2. Data set 2

Data set 2 contains the LC–MS signals of 6 urine samples [56]. The LC–MS samples were injected onto an Acquity™ UPLC BEH 2.1 × 100 mm, 1.7 μm column (Waters, Manchester, UK) and separated using a solvent gradient. Mass spectrometric determinations were made using an LCT Premier™ mass spectrometer (Waters). Most reagents used in the experiment were of the LC–MS purity grade.

4.1.3. Data set 3

Data set 3 contains 44 GC–MS signals, downloaded from [57], representing 44 red wine samples, produced from one and the same grape variety, harvested in different geographical areas, that have been purchased from the local supermarkets in the area of Copenhagen and Denmark. Geographical origins and the numbers of wine samples are, as follows: Argentina (6), Chile (15), Australia (12), and South Africa (11). The wine samples have been analyzed using the head space GC–MS. For each sample, the mass spectrum scan (m/z: 5–204) measured at 2700 elution time-points was obtained, providing the data cube of size 44 × 2700 × 200. Full description of the data and sample preparation is given in Ref. [58].

4.2. Results on data set 1

The principles of the proposed method based on particle swarm optimization will be discussed and demonstrated in detail for data set 1, containing the two signals presented in Fig. 5a and b. As already reported, this data set has already been used as a benchmark to test different alignment algorithms.

In order for the PSO approach to be used for alignment, it is necessary to define a parameterized warping function and a corresponding optimization criterion (fitness). As described in Eq. (15), in this study a linear combination of equally spaced radial basis functions is used to perform the mapping between the original and warped time scales, so the parameters to be optimized are the regression coefficients $c_1, c_2, \ldots c_k$. On the other hand, the correlation coefficient between the warped and the target profiles is chosen as the fitness criterion.

Before applying the alignment procedure, the two signals in Fig. 5a and b were background corrected and extrapolation to the same length was performed: the initial correlation of the two signals equals to 0.24. Successively, to proceed with PSO, user has to define the number of basis functions, *k*, and the population size. For $k = 5$ and population size = 50 with a constant inertia of 0.8 and the maximum number of

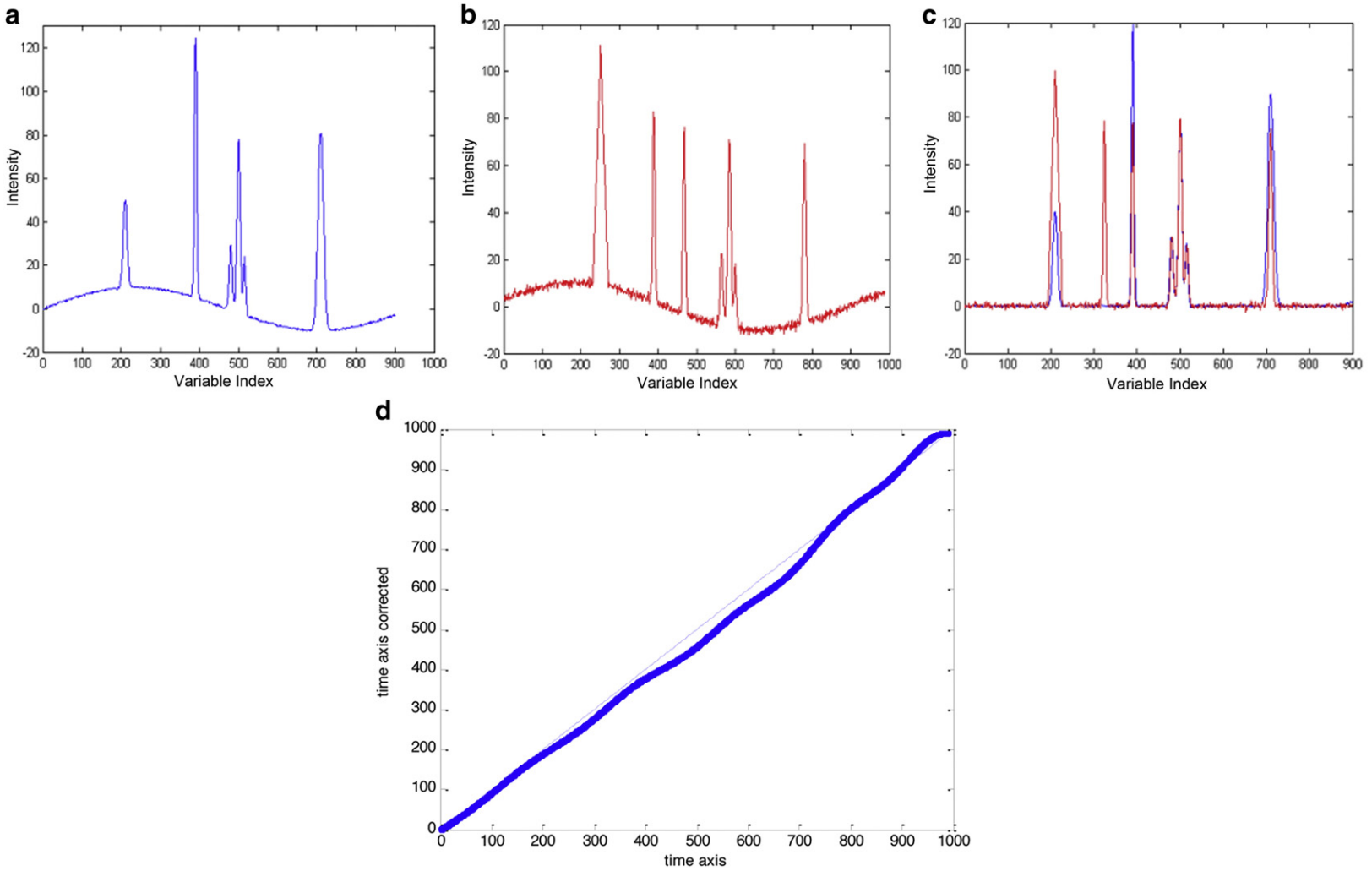

Fig. 5. Application of PSO for chromatografic signal alignment. Graphical representation of the two chromatograms which constitute data set 1: the signal in (a) is the target, while that in (b) is the one to be aligned. Panel (c) shows the two warped chromatograms, after baseline correction, while the warping function is reported in (d).

iterations set to 100, the algorithm converged to a set of parameters resulting in a correlation coefficient of 0.74, which is exactly the same result that can be obtained using the COW [49], AA [52] or FW [55] algorithm. The corresponding relation between the transformed and the original time axis of the warped chromatogram, i.e., the time warping function described in Eq. (14), is plotted in Fig. 5d.

4.3. Results on data set 2

Warping is necessary not only for an alignment of 1-dimensional chromatographic signals, but of the 2-dimensional signals as well. In the case of the LC–MS or GC–MS signals, warping can be performed for total ion chromatograms (TICs) of the studied samples. As described in Section 4.1.2, data set 2 contains LC–MS matrices for six samples, based on which, for the sake of the alignment procedure, the corresponding six TICs are calculated. Out of these six TICs, the first one is selected as a target, so that all the remaining have to be warped to it. Accordingly, altogether five pairs of signals (each one containing the target chromatogram) need to be aligned, and it is necessary to optimize the transfer function for each pair individually, i.e., for each pair of chromatograms a different set of regression coefficients (see Eq. (16)) was optimized by the PSO algorithm. In this case, 6 radial basis functions were considered and PSO was run with a population of 50 individuals and a constant inertia weight of 0.8, until a maximum number of 100 iterations. As an example of the results, regions of the target signal and of the TIC profile of sample no. 2 before and after warping are presented in Fig. 6.

Once the optimal transformation parameters are found for the studied pair of signals, the corresponding warping function can be used for the alignment of all the individual ion chromatograms, so that the entire LC–MS matrices can be used in further steps of data analysis, if necessary.

4.4. Results on data set 3

The third data set considered in our study contains GC–MS signals registered on wine samples (of dimensionality 44 samples × 2700 time points × 200 m/z ratios). As in the case of the LC–MS data, warping can be performed using TICs. In this case, all samples were warped to sample no. 19, which has the highest mean correlation coefficient with the remaining signals. The optimal settings of PSO were, as follows: $k = 7$, population size = 50, maximum number of iterations = 100, and constant inertia weight = 0.8. As shown in Fig. 7a–b, where the TIC chromatograms of the samples are reported before and after warping, a perfect alignment is achieved using PSO-based parametric time warping.

Moreover, as reported in Fig. 7c–d, where the correlation coefficients between pairs of chromatograms are reported in the form of a colormap, while before alignment the minimum value of the correlation coefficient was around 0.79 and many pairs of profile had a correlation lower than 0.92, PSO-based warping leads to a minimum correlation of 0.965 and most of the values are in the range of 0.99–1. As a further proof of the effectiveness of the proposed approach, one can inspect the amount of variance explained by the first principal component extracted on the data before and after warping: while in the first case PC1 retains 46.3% of the original variance, for aligned signals this percentage becomes 65.3%.

5. A second chemometric example: PSO-based projection pursuit for robust PCA

As a second example of the potential of PSO algorithms in chemometrics, the possibility of improving the method of Croux and Ruiz for the calculation of robust PCA models based on projection pursuit is discussed [59]. Generally speaking, projection-pursuit (PP) methods aim at finding structures in multivariate data by projecting them onto a lower-dimensional subspace [60,61], whose axis are selected among some candidate direction as those which maximize a certain criterion, which is defined as the projection index. In this context, principal component analysis can be seen as a PP in which the projection index is the variance along the particular projection direction. As a consequence of this, during the years several methods for the calculation of robust principal component models have been based on the use of a projection pursuit approach, taking as projection index a robust measure of variance [62], such as the square of the median absolute deviation (MAD) or of the Q_n estimator (first quartile of the pairwise differences between all data points [63]). In the original Croux and Ruiz algorithm (CR), the search directions of the first robust PC are chosen among those connecting each of the data points to the origin of the axes (after robust centering, by subtraction of the multivariate median or the L1-median [64]). Subsequent PCs are estimated in a similar way, but the search is done in the orthogonal complement of the previously identified PCs (i.e., along directions passing through the data points deflated for the previously extracted components). It is then evident from the definition that the original CR algorithm may not lead to optimal results for data set with low sample size or low samples to variable ratio. Moreover, since each of the projection directions is determined by one of the data points, the algorithm may implode, i.e., irrespective of the data, by defining k as half the number of samples, from the $(k + 1)$th PC included all the calculated eigenvalues will be zero. To overcome these drawbacks, different modifications have been proposed to the original algorithm like, e.g., a grid search, or the

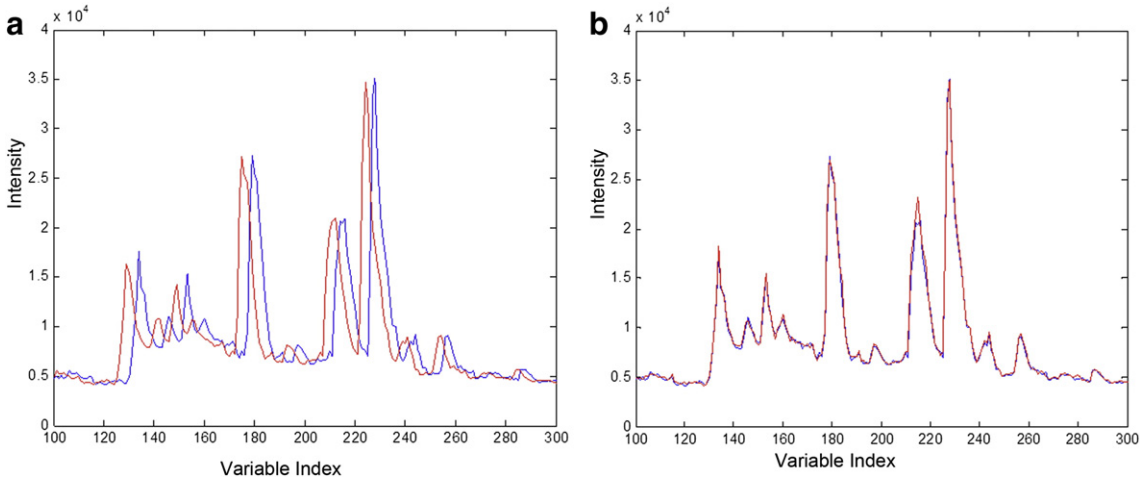

Fig. 6. Application of PSO for chromatografic signal alignment. Data set 2: fragment of target TIC (sample nr. 1) and of the TIC of sample nr. 2 before (a) and after (b) warping.

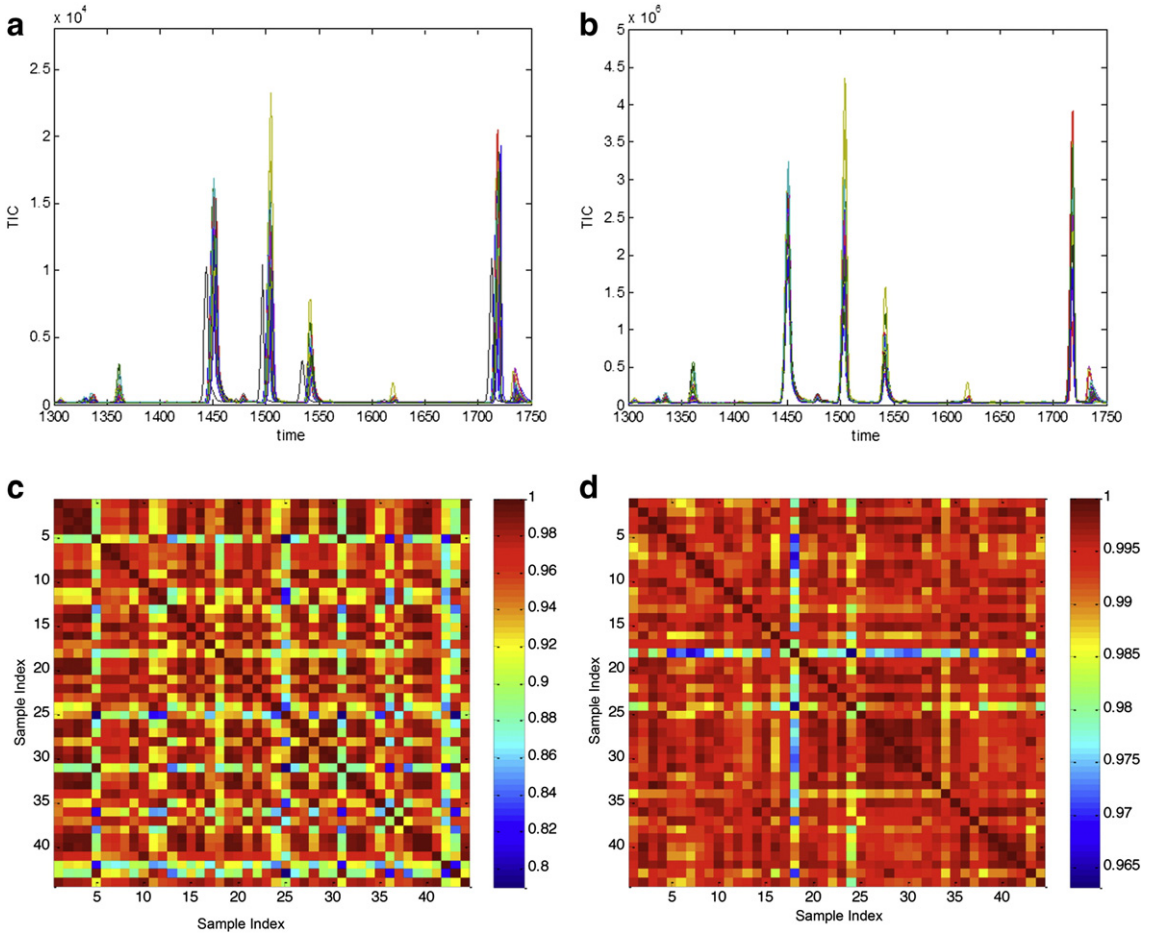

Fig. 7. Application of PSO for chromatografic signal alignment. TIC calculated from GC–MS signals constituting data set 3 before (a) and after (b) warping. The correlation between pairs of samples before and after warping is reported in panels (c) and (d), respectively, in the form of heatmaps.

integration of the set of candidate solutions with randomly generated directions in the search space [65].

The present example addresses the possibility of adopting a PSO-based approach for the calculation of robust PCA models in the framework of the CR algorithm, i.e., using projection pursuit. Considering a data set **D**, composed of n samples and m variables, the optimization problem involved in PSO-based projection pursuit for the calculation of robust PCA can be defined as the search of the components of the m-dimensional vector of robust loadings **a**, so that the robust variance along **a** is maximum (here, squared Q_n was used as the robust estimator of variance):

$$\mathbf{a}_1 = \arg\max_{\|\mathbf{a}\|=1}(Q_n(\mathbf{Da})) . \tag{17}$$

Calculation is iterative and, for all the components successive to the first, occur on the data matrix deflated from the contribution of the previously calculated robust loadings, i.e., on the orthogonal complement of the matrix $\mathbf{A}_k$, k being the number of previously calculated PCs:

$$\mathbf{D}_k = \mathbf{D}\left(1-\mathbf{A}_k\mathbf{A}_k^T\right). \tag{18}$$

In the proposed approach, the robust loadings $\mathbf{a}_i$ are obtained as the global best vectors obtained after successive applications of a PSO-based projection pursuit algorithm in which each particle represents a candidate projection direction and the fitness is defined as the squared Q_n along that same direction. For the calculation of each component a population of 50 particles was used and the maximum number of iterations was set to 200; linearly decreasing inertia, varying from 0.9 to 0.4 was also adopted. As an example, the results of the application of the proposed approach to the Hawkings–Bradu–Kass data set, which is a famous data set in robust statistics [66], are discussed. The Hawkings–Bradu–Krass data set is a small simulated data set of dimension 75×4; it contains two groups of leverage points and outliers which are labeled, respectively objects 1–10 and 11–14. Since most of the variance is captured by the first two components, that have been selected as the model complexity. Fig. 8 shows the results of the application of classical PCA and of the proposed approach on the Hawkings–Bradu–Krass data set, together with the outcomes of the original CR approach and of another commonly used robust PCA approach, ROBPCA [67].

As evident from the figure, PSO-based projection pursuit approach provides a sample mapping that is similar to the other two standard robust approaches and moreover, when considering the explained robust variance, i.e. the squared Q_n, it performs better than both the CR algorithm and ROBPCA. Indeed, PSO-projection pursuit leads to a robust variance of 1.917 and 1.665 for rPC1 and rPC2, respectively while the corresponding values for CR are 1.864 and 1.620 and those for ROBPCA are 1.344 and 1.112.

6. A discrete PSO for variable selection

As discussed in Section 2.3, although PSO was originally developed for real-valued optimization, it may also be used for discrete data. In this context, one of their possible applications is in the domain of variable selection, where the task is to find the optimal subset of descriptors/predictors, e.g. for improving the predictive ability of a regression or classification model or for biomarker discovery. Indeed, already in a previous paper [17], the authors discussed the possibility of adopting a

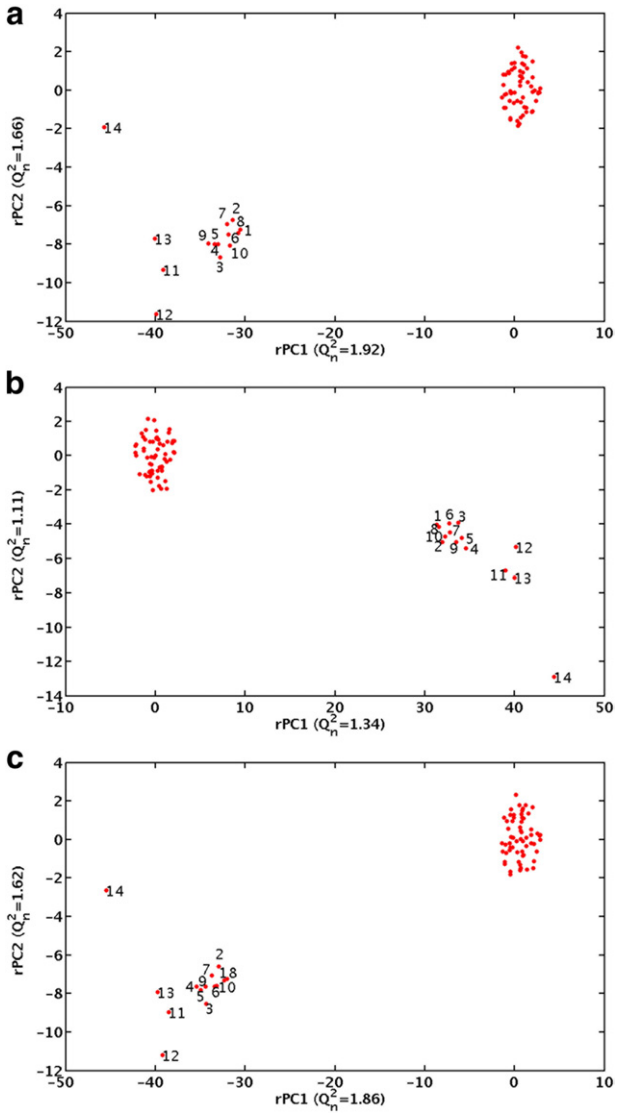

Fig. 8. Application of PSO for the extraction of robust principal components. Projection of the samples composing the Hawkings–Bradu–Krass data set onto the first two robust PCs calculated by the proposed PSO–PP approach (a), the original Croux–Ruiz PP approach (b) and ROBPCA algorithm (c). For the sake of better visualization, the sample index for the 14 outliers and high leverage points have been shown on the plots.

strategy based on discrete PSO for the unsupervised identification of subspaces with clustering tendencies within the data. In that occasion, PSO proved to be very effective in highlighting not only the variables that were responsible for the main data structure but also other subspaces corresponding to local optima, carrying interesting information especially in the case of a real data set coming from the analysis of genomic microarrays [17]. In the present paper, the possibility of using PSO for variable selection is addressed in the framework of PLS, i.e. in the context of multivariate calibration [68].

In order to operate variable selection, the parameter vector, i.e. each particle, has as many dimensions as the number of variables in the data matrix and a binary coding is adopted to account for whether the particular predictor is included or not in the PLS model. Fitness, as in other evolutionary approaches for variable selection, such as genetic algorithms, may be defined as the value of R^2 in cross-validation. To avoid the risk of overfitting, similar to what was suggested by Leardi for GA-PLS [69], we propose to repeat multiple times the PSO simulation and build the final model based on the consensus result of the different repetitions. For the present paper, this approach has been applied to a data set coming from a study on the characterization of oat samples by NIR

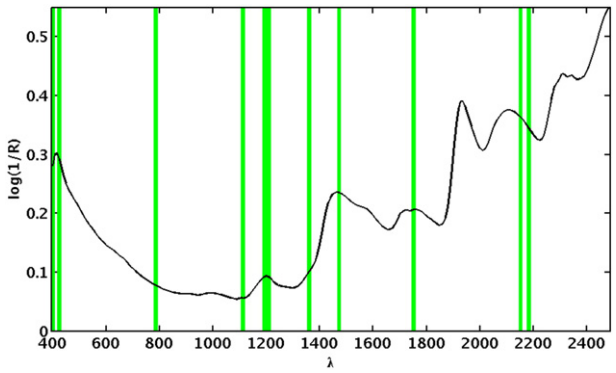

Fig. 9. Application of PSO for variable selection: Representation of the 11 variables selected by the PSO–PLS approach in the case of the oat data set, superimposed to the mean NIR spectrum of the training samples.

spectroscopy [70]. For the sake of the present example, only a part of the original data set was considered: in particular, 166 NIR spectra, divided into 126 training and 40 test sets are used to quantify the protein content of the samples and, prior to PSO analysis, the 1050 original wavelengths have been compressed to 262 by taking one variable every fourth. The variable selection stage was performed by carrying out 30 PSO runs of 200 iteration each, considering a swarm of 30 particles and a chaotic inertia weight update; R^2 in cross-validation was used as the fitness criterion. Only the variables, which were selected in the global best of at least 75% of the runs, were retained in the final PLS model.

The proposed approach, retained 11 variables only in the final PLS model and the plot reported in Fig. 9 shows that all of them seem to be chemically relevant, so that one may rule out the possibility of overfitting. Moreover, when the optimal model according to PSO was used to predict the protein content of the 40 external test set samples, the RMSEP calculated was slightly lower than that of the full model computed on all the predictors (0.342 for the former and 0.359 for the latter).

7. Selected literature applications

As anticipated in Section 1, despite the potential of the approach, there are still relatively few applications of particle swarm optimization reported in the chemometric literature. The present section aims at integrating the possible examples of application sketched in the previous paragraphs, by presenting some other possible uses of PSO, which have been so far described by scientists in the field of chemometrics. Schwaab et al. used PSO for nonlinear fitting, in order to estimate model parameters and their confidence intervals for reaction kinetics example, highlighting non-elliptical shapes of the confidence regions, which in some cases were found to be unbounded, non-convex and composed of unconnected parts [16]. On the other hand, Guo et al. adopted a strategy based on PSO to tune the parameters of support vector machines trained with different kernels [71]. Recently, three papers addressed the possibility of using PSO to calculate MCR solutions and their feasible bands. Indeed, Parastar et al. proposed to use PSO to calculate the elements of the rotation matrix leading from PCA to MCR solutions and demonstrated that the obtained results are always within the feasible regions estimated by state of the art approaches [19]. A similar approach was followed by Beyramysoltan et al. for studying the effects of the equality constraint on reducing the rotational ambiguity in three component systems [20]. On the other hand, Skvortsov introduced a modified version of the algorithm, called modified charged PSO (cPSO), together with a tailored convergence rule, causing particles not to accumulate on the global optimum but rather to reach a steady state: this approach allowed to investigate rotational ambiguity of MCR solution for selected test cases and to estimate the range of feasible bands [21].

Still in the context of mixture analysis, Fan et al. have proposed a new component analysis method based on PSO for blind source separation [22]. Indeed, they showed that, in the context of process manufacturing, blind source separation can be expressed as nonlinear constrained optimization problem, where the unknown demixing matrix can be estimated by an approach based on PSO. The use of PSO for improved process monitoring and modeling was also recently considered by Krause et al. [23], who proposed a self-organizing sensor network where swarm intelligence was used to score different models, in order to select the optimal sensor/model combination at the various time steps for the determination of the process trajectory. In particular, the approach was applied to a fermentation bioprocess. Lastly, in the framework of variable selection, where the PSO algorithm has so far found most of its applications in chemometrics [17,18,24,25,27], it is worth mentioning the possibility of adopting hybrid strategies. Li et al. have used a discrete-continuous PSO approach to achieve at the same time variable selection and to calculate the optimal weighting scheme in PLS-DA for metabolomics applications [27]. On the other hand, Cernuda and coworkers have proposed a hybrid PSO–ACO approach to integrate the advantages and compensate for the drawbacks of the individual techniques and have shown its potential for variable selection in the context of NIR spectroscopy [25].

8. Software

All the examples shown and the algorithm discussed have been implemented through in-house functions running under Matlab R2012b (The Mathworks, Natick, MA) environment. Matlab codes and selected examples may be freely downloaded from the website http://www.chem.uniroma1.it/romechemometrics/.

9. Conclusions

Due to their flexibility, simplicity of use and implementation and versatility, particle swarm optimization may be applied to many different chemometric fields, as demonstrated by the examples discussed and also by the brief literature survey presented. Indeed, it has proved to be useful for signal alignment, where the warping function may be estimated directly through the implementation of a PSO-radial basis function approach, in projection pursuit, where it can be used to select the optimal direction in space according to the projection index, or for variable selection, but for sure the possibility of application exceeds those discussed in the present paper. Moreover, with respect to genetic algorithms, which are the family of evolutionary optimization methods currently most used in chemometrics, PSO appears to be better suited to deal with real-valued optimization, as it don't need any recoding (discretization) of the variables or redefinition of the operators, while at the same time it can be very easily adapted to deal with binary- or discrete-coded problems, as a further proof of its versatility. Altogether, particle swarm optimization has a great potential in chemometrics; it is then the authors' hope that the present tutorial introduction to the subject may trigger some further interest in the topics and foster a wider use of this technique.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] R.C. Eberhart, Y. Shi, J. Kennedy, Swarm Intelligence, Morgan Kaufmann, Burlington, MA, 2001.
[2] G. Beni, J. Wang, Swarm Intelligence, RSJ Press, Tokyo, Japan 1989, pp. 425–428.
[3] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley and sons, New York, NY, 2005.
[4] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence. FROM NATURAL to Artificial Systems, Oxford university press, New York, NY, 1999.
[5] C. Blum, D. Merkle (Eds.), Swarm Intelligence: Introduction and Applications, Springer verlag, Heidelberg, Germany, 2008.
[6] R. Leardi, Genetic algorithms in chemometrics and chemistry: a review, J. Chemometr. 15 (2001) 559–569.
[7] J.H. Kalivas (Ed.), Adaption of Simulated Annealing to Chemical Optimization Problems, Elsevier, Amsterdam, The Netherlands, 1995.
[8] X.-S. Yang, Nature-inspired Optimization Algorithms, Elsevier, London, UK, 2014.
[9] M. Dorigo, Optimization, Learning and Natural Algorithms(PhD thesis) Politecnico di Milano, Italy, 1992.
[10] M. Dorigo, T. Stützle, Ant Colony Optimization, Bradford books, Cambridge, MA, 2004.
[11] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks 1995, vol.4, IEEE press 1995, pp. 1942–1948.
[12] M. Meissner, M. Schmuker, G. Schneider, Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training, BMC Bioinformatics 7 (2006) 125.
[13] N. Mohammadi, S.J. Mirabedini, Comparison of particle swarm optimization and backpropagation algorithms for training feedforward neural network, J. Math. Computer Sci. 12 (2014) 113–123.
[14] E. Hadavandi, Developing a time series model based on particle swarm optimization for gold price forecasting, Proceedings of the Third International Conference on Business Intelligence and Financial Engineering, IEEE press, New York, NY 2010, pp. 337–340.
[15] X.S. Yang, S. Deb, S. Fong, Accelerated particle swarm optimization and support vector machine for business optimization and applications, in: S. Fong (Ed.), Networked Digital Technologies. Proceedings of the Third International Conference NDT2011, Springer, Heidelberg, Germany 2011, pp. 53–66.
[16] M. Schwaab, E. Chalbaud Biscaia Jr., J.L. Monteiro, J.C. Pinto, Nonlinear parameter estimation through particle swarm optimization, Chem. Eng. Sci. 63 (2008) 1542–1552.
[17] F. Marini, B. Walczak, Finding relevant clustering directions in high-dimensional data using particle swarm optimization, J. Chemometr. 25 (2011) 366–374.
[18] N. Sorol, E.L. Arancibia, S.A. Bortolato, A.C. Olivieri, Visible/near infrared-partial least-squares analysis of Brix in sugar cane juice. A test field for variable selection methods, Chemometr. Intell. Lab. Syst. 102 (2010) 100–109.
[19] H. Parastar, H. Ebraimi-Najafabadi, M. Jalali-Heravi, Multivariate curve resolution-particle swarm optimization: a high-throughput approach to exploit pure information from multi-component hyphenated chromatographic signals, Anal. Chim. Acta 772 (2013) 16–25.
[20] S. Beyramysoltan, R. Rajkó, H. Abdollahi, Investigation of the equality constraint effect on the reduction of the rotational ambiguity in three-component system using a novel grid search method, Anal. Chim. Acta 791 (2013) 25–35.
[21] A.N. Skvortsov, Estimation of rotation ambiguity in multivariate curve resolution with charged particle swarm optimization (cPSO-MCR), J. Chemometr. 28 (2014) 727–739.
[22] S.-K.S. Fan, Y. Lin, C.-M. Fan, Y.-Y. Wang, Process identification using a new component analysis model and particle swarm optimization, Chemometr. Intell. Lab. Syst. 99 (2009) 19–29.
[23] D. Krause, M.A. Hussein, T. Becker, Online monitoring of bioprocesses via multivariate sensor prediction within swarm intelligence decision making, Chemometr. Intell. Lab. Syst. 145 (2015) 48–59.
[24] Q. Shen, J.-H. Jiang, C.-X. Jiao, G.-L. Shen, R.-Q. Yu, Modified particle swarm optimization algorithm for variable selection in MLR and PLS modeling: QSAR studies of antagonism of angiotensin II antagonists, Eur. J. Pharm. Sci. 22 (2004) 145–152.
[25] C. Cernuda, E. Lughofer, W. Märzinger, W. Summerer, Hybrid evolutionary particle swarm optimization and ant colony optimization for variable selection, Proceedings of the 3rd World Conference on Information Technology (WCIT-2012), Series 3rd World Conference on Information Technology (WCIT-2012), vol. 3, AWERProcedia Information Technology & Computer Science, Famagusta, Cyprus 2013, pp. 7–14.
[26] R.-M. Luo, Y.-Q. Li, H.-L. Guo, Y.-P. Zhou, H. Xu, H. Gong, Adaptive configuration of radial basis function network by regression tree allied with hybrid particle swarm optimization algorithm, Chemometr. Intell. Lab. Syst. 124 (2013) 50–57.
[27] Y.-Q. Li, Y.-F. Liu, D.-D. Song, Y.-P. Zhou, L. Wang, S. Xu, Y.-F. Cui, Particle swarm optimization-based protocol for partial least-squares discriminant analysis: application to 1H nuclear magnetic resonance analysis of lung cancer metabonomics, Chemometr. Intell. Lab. Syst. 135 (2014) 192–200.
[28] R. Fletcher, Practical Methods of Optimization, John Wiley and sons, New York, NY, 2000.
[29] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison Wesley, Boston, MA, 1984.
[30] S. Edelkamp, S. Schroedl, Heuristic Search: Theory and Applications, Morgan Kaufmann, Burlington, MA, 2011.
[31] Y. Shi, R.C. Eberhart, Parameter selection in particle swarm optimization, in: V.W. Porto, N. Saravanan, D. Waagen, A.E. Eiben (Eds.), Evolutionary Programming VII, Proceedings of the 7th International Conference EP98Springer, Heidelberg, Germany 1998, pp. 591–600.
[32] F. van den Bergh, An analysis of Particle Swarm Optimizers(PhD thesis) University of Pretoria, South Africa, 2001.
[33] M. Clerc, J. Kennedy, The particle swarm: explosion stability and convergence in a multi-dimensional complex space, IEEE Trans. Evolution. Comput. 6 (2002) 58–73.
[34] A. Engelbrecht, Particle swarm optimization: velocity initialization, Proceedings of the IEEE Congress on Evolutionary Computation 2012, IEEE press, New York, NY 2012, pp. 1–8.
[35] E. Ozcan, C.K. Mohan, Analysis of a simple particle swarm optimization system, in: C.H. Dagli, M. Akay, A.L. Buczak, O. Ersoy, B.R. Bernandez (Eds.), Intelligent Engineering Systems Through Artificial Neural Networks, Proceedings of the Artificial Neural

Networks in Engineering Conference (ANNIE 98), vol. 8, American Society of Mechanical Engineering (ASME), New York, NY 1998, pp. 253–258.
[36] R.C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, Proceedings of the 2000 Conference on Evolutionary Computation, vol.1, IEEE press, New York, NY 2000, pp. 84–88.
[37] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, Proceedings of IEEE 1998 International Conference on Evolutionary Computation, vol.1, IEEE press, New York, NY 1998, pp. 69–73.
[38] J.C. Bansal, P.K. Singh, M. Saraswat, A. Verma, S. Singh Jadon, A. Abraham, Inertia weight strategies in particle swarm optimization, Proceedings of the 2011 Third World Congress on Nature and Biologically Inspired Computing, IEEE press, New York, NY 2011, pp. 640–647.
[39] R.C. Eberhart, Y. Shi, Tracking and optimizing dynamic systems with particle swarms, Proceedings of the 2001 Congress on Evolutionary Computation, vol.1, IEEE press, New York, NY 2001, pp. 94–100.
[40] J. Xin, G. Chen, Y. Hai, A particle swarm optimizer with multistage linearly-decreasing inertia weight, Proceedings of the International Joint Conference on Computational Sciences and Optimization (CSO 2009), vol.1, IEEE press, New York, NY 2009, pp. 505–508.
[41] M.S. Arumugam, M.V.C. Rao, On the performance of the particle swarm optimization algorithm with various inertia weight variants for computing optimal control of a class of hybrid systems, Discrete Dyn. Nat. Soc. 2006 (2006) 79295.
[42] Y. Feng, G.F. Teng, A.X. Wang, Y.M. Yao, Chaotic Inertia Weight in Particle Swarm Optimization, IEEE press, New York, NY 2007, p. 475.
[43] A.E. Muñoz Zavala, A comparison study of PSO neighborhoods, in: O. Schütze, C.A. Coello Coello, A.-A. Tantar, E. Tantar, P. Bouvry, P. Del Moral, P. Legrand (Eds.), EVOLVE — A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II, Springer, Heidelberg 2013, pp. 251–265.
[44] J. Kennedy, A discrete binary version of the particle swarm algorithm, Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics, vol. 5, IEEE press, New York, NY 1997, pp. 4104–4108.
[45] D.H. Ackley, A Connectionist Machine for Genetic Hillclimbing, Kluwer Academic Publishers, Boston, MA, 1987.
[46] T. Bäck, Evolutionary Algorithms in Theory and Practice, Oxford university press, Oxford, UK, 1996.
[47] G. Malmquist, R. Danielsson, Alignment of chromatographic profiles for principal component analysis: a prerequisite for fingerprinting methods, J. Chromatogr. A 687 (1994) 71–88.
[48] A. Kassidas, J. MacGregor, P. Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE J. 44 (1998) 864–875.
[49] N.P. Vest Nielsen, J.M. Carstensen, J. Smedsgaard, Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping, J. Chromatogr. A 805 (1998) 17–35.
[50] P.H.C. Eilers, Parametric time warping, Anal. Chem. 76 (2004) 404–411.
[51] K.J. Johnson, B.W. Wright, K.H. Jarman, R.E. Synovec, High-speed peak matching algorithm for retention time alignment of gas chromatographic data for chemometric analysis, J. Chromatogr. A 996 (2003) 141–155.
[52] M. Daszykowski, Y. Vander Heyden, C. Boucon, B. Walczak, Automated alignment of one-dimensional chromatographic signals, J. Chromatogr. A 1217 (2010) 6127–6133.
[53] F. Savorani, G. Tomasi, S.B. Engelsen, icoshift: a versatile tool for the rapid alignment of 1D NMR spectra, J. Magn. Reson. 202 (2010) 190–202.
[54] G. Tomasi, F. Savorani, S.B. Engelsen, icoshift: an effective tool for the alignment of chromatographic data, J. Chromatogr. A 1218 (2011) 7832–7840.
[55] B. Walczak, W. Wu, Fuzzy warping of chromatograms, Chemometr. Intell. Lab. Syst. 77 (2005) 173–180.
[56] M. Daszykowski, W. Wu, A.W. Nicholls, R.J. Ball, T. Czekaj, B. Walczak, Identifying potential biomarkers in LC–MS, J. Chemometr. 21 (2007) 292–302.
[57] http://www.models.kvl.dk/Wine_GCMS_FTIR (Last accessed: August 2015).
[58] T. Skov, D. Balabio, R. Bro, Multiblock variance partitioning. a new approach for comparing variation in multiple data blocks, Anal. Chim. Acta 615 (2008) 18–29.
[59] C. Croux, A. Ruiz-Gazen, A fast algorithm for robust principal components based on projection pursuit, in: A. Prat (Ed.), COMPSAT. Proceedings in Computational Statistics, Part III, Springer, Heidelberg, Germany 1996, pp. 211–216.
[60] J.H. Friedman, J.W. Tuckey, A projection pursuit algorithm for exploratory data analysis, IEEE Trans. Comp. C-23 (1974) 881–890.
[61] P.J. Huber, Projection pursuit, Ann. Stat. 13 (1985) 435–475.
[62] R. Maronna, R.D. Martin, V. Yohai, Robust Statistics — Theory and Methods, John Wiley and sons, New York, NY, 2006.
[63] P.J. Rousseouw, C. Croux, Alternatives to the median absolute deviation, J. Am. Stat. Assoc. 88 (1993) 1273–1283.
[64] Y. Vardi, C.H. Zhang, The multivariate L-1 median and associated data depth, Proc. Nat. Acad. Sci. 97 (2000) 1423–1426.
[65] C. Croux, P. Filzmoser, M. Oliveira, Algorithms for projection-pursuit robust principal component analysis, Chemometr. Intell. Lab. Syst. 87 (2007) 218–225.
[66] D.M. Hawkins, D. Bradu, G.V. Kass, Location of several outliers in multiple regression data using elemental sets, Technometrics 26 (1984) 197–208.
[67] M. Hubert, P.J. Rousseeuw, K. Vanden Branden, ROBPCA: a new approach to robust principal components analysis, Technometrics 47 (2005) 64–79.
[68] H. Martens, T. Naes, Multivariate Calibration, John Wiley and sons, New York, NY, 1989.
[69] R. Leardi, Genetic algorithms-PLS as a tool for wavelength selection in spectral data sets, in: R. Leardi (Ed.), Nature-inspired Methods in Chemometrics, Elsevier, Amsterdam, The Netherlands 2003, pp. 169–195.
[70] S. Bellato, V. Del Frate, R. Redaelli, D. Sgrulletta, R. Bucci, A.D. Magrì, F. Marini, Use of near infrared reflectance and transmittance coupled to robust calibration for the evaluation of nutritional value in naked oats, J. Agric. Food Chem. 59 (2011) 4349–4360.
[71] X.C. Guo, J.H. Yang, C.G. Wu, C.Y. Wang, Y.C. Liang, A novel LS-SVMs hyper-parameter selection based on particle swarm optimization, Neurocomputing 71 (2008) 3211–3215.